

Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Sähköisen liiketoiminnan järjestelmät

Toukokuu 2013

Jesse Tuominen

# SILVERLIGHT-SOVELLUKSEN KEHITTÄMINEN AGILE- MENETELMÄN JA MVVM- SUUNNITTELUMALLIN AVULLA



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

Jesse Tuominen

# SILVERLIGHT-SOVELLUKSEN KEHITTÄMINEN AGILE- MENETELMIEN JA MVVM- SUUNNITTELMALLIN AVULLA

Tässä opinnäytetyössä esitetään Cargotec Oy:lle toteutettu C-Service tuotetietokatalogi-sovellus, joka on tehty Silverlight-tekniikalla. Tavoitteena oli parantaa sovelluksen tilausprosessia, käyttökokemusta sekä vähentää ylläpitokustannuksia. Työssä määritellään lisäksi sovelluskehityksessä käytettyjä Silverlight-tekniikkaa, MVVM-ohjelmistoarkkitehtuuria sekä ketteriä menetelmiä.

Silverlight on Microsoftin kehittämä ohjelmistokehitys, jonka avulla voidaan rakentaa RIA- (Rich Internet Application) eli rikkaita internetsovelluksia. Rikkailla internetsovelluksilla on työpöytäsovelluksille ominaisia piirteitä. Silverlight pyrkii parantamaan käyttökokemusta sekä tarjoamaan nykyaikaisen ja kehittyneen käyttöliittymän. Silverlight on yhteensopiva useiden eri selainten sekä käyttöjärjestelmien kanssa.

MVVM-arkkitehtuuri on Microsoftin kehittämä suunnittelumalli, joka koostuu sanoista malli, näkymä ja näkymämalli (Model, View, ViewModel). MVVM-arkkitehtuuri mahdollisti kehitystiimille tehokkaan tavan työskennellä projektin eri osioiden kanssa häiritsemättä toistensa työtä. Arkkitehtuurille ominainen tiedostorakenne antoi selkeän järjestyksen, jota kehitystiimin oli helppo seurata.

Sovelluskehitystyö toteutettiin ketterissä menetelmissä käytettävien iteraatioiden avulla. Yksi iteraatio oli noin viikon pituinen. Iteraatioiden käyttäminen selkeytti projektityöskentelyä, antaen kehittäjille tarkat tehtävät mihin keskittyä jakson aikana.

Menetelmien käyttäminen helpotti tehtävien organisointia sekä kehittäjien ja managereiden välistä vuorovaikutusta. Ketteriä menetelmiä ei seurattu sen puhtaassa muodossaan, vaan enemmänkin projektiin sopivien parhaiden käytäntöjen kautta. Tässä opinnäytetyössä saatujen havaintojen perusteella Silverlight-sovelluksen kehittäminen MVVM-arkkitehtuuria hyödyntämällä osoittautui tehokkaaksi tavaksi rakentaa sovelluksia. Projektin aikana opittuja menetelmiä ja suunnittelumalleja voidaan hyödyntää tulevissa sovelluskehitysprojekteissa.

## ASIASANAT:

sovelluskehitys, silverlight, MVVM, ketterät menetelmät

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | e-Business Systems

May 2013 | 39

Tuukka Hämeenoja

Jesse Tuominen

# DEVELOPING A SILVERLIGHT APPLICATION USING AGILE METHODS AND MVVM DESIGN PATTERN

This thesis deals with C-Service application development process made for Cargotec Oyj. C-Service is a product catalog application that was developed by using Silverlight framework, MVVM architecture and Agile methods. The main goal was to improve the order process, usability and lower the maintenance costs of the C-Service application. The focus of this thesis lies on the usage of Silverlight framework, MVVM architecture and Agile methods used in development of C-Service application.

Silverlight is an application framework developed by Microsoft. Silverlight is used in order to build rich internet applications and it has same kind of features as a desktop application. Silverlight is cross-browser and cross-platform compatible for the development of modern user interfaces and provides better user experience for internet applications.

Agile methods consist of various project management tools and methods that are used in software development. These methods have common features where the development work is always of the highest priority. C-Service development was done by using Agile methods and iterations to give clear goals and tasks to the development team. Iterations were one week long and they simplified the development work.

Developers gain good advantage in C-Service development by using agile methods. This helped in the interaction between the developers and managers to be always aware of the different situations concerning the development work. Development team used the best practices of agile methods to fit the size of development work. Based on the findings in this thesis, the usage of Silverlight and MVVM architecture in application development is a productive way of working. These methods and architectures described in this thesis can be used in the upcoming development projects.

## KEYWORDS:

application development, MVVM, agile methods, Silverlight

## SISÄLTÖ

<b>1 JOHDANTO</b>	<b>1</b>
<b>2 ECRAFT OY AB JA KEY ACCOUNT ESPOO</b>	<b>2</b>
2.1 Avainasiakas Cargotec Oyj	3
2.2 C-Office-tietojärjestelmä	4
<b>3 SUUNNITTELMALLIT</b>	<b>6</b>
3.1 MVC-arkkitehtuuri	6
3.2 MVVM -arkkitehtuuri	7
<b>4 MENETELMIEN KÄYTTÄMINEN ASIAKASPROJEKTISSA</b>	<b>10</b>
4.1 Ketterät menetelmät (Agile)	10
4.2 Lean-malli	11
4.3 SCRUM	13
4.4 Menetelmien käyttö eCraft Oy Ab:ssä	16
<b>5 TEKNOLOGIA</b>	<b>17</b>
5.1 Silverlight	17
5.2 Silverlight ja Windows Presentation Foundation	20
5.3 UX Factory lähdekoodikirjasto	20
<b>6 SILVERLIGHT SOVELLUKSEN TUOTTAMINEN AVAINASIAKASYRITYKSELLE</b>	<b>22</b>
<b>7 LOPPUPÄÄTELMÄ</b>	<b>29</b>
<b>LÄHTEET</b>	<b>30</b>

## **KUVAT**

<b>KUVA 3.1: MVC-MALLI</b>	<b>16</b>
<b>KUVA 3.2: MVVM-ARKKITEHTUURI</b>	<b>17</b>
<b>KUVA 4.2: ITERAATIO OHJELMISTOKEHITYKSESSÄ</b>	<b>20</b>
<b>KUVA 4.3: ONGELMAN TUNNISTAMINEN</b>	<b>22</b>
<b>KUVA 4.4: SCRUM-PROSESSI</b>	<b>24</b>
<b>KUVA 6.1: HAKUTOIMINTO</b>	<b>33</b>
<b>KUVA 6.2: PUUNÄKYMÄ</b>	<b>35</b>
<b>KUVA 6.3: C-SERVICE SILVERLIGHT</b>	<b>36</b>
<b>KUVA 6.4: SOVELLUSARKKITEHTUURI</b>	<b>37</b>

## **TAULUKOT**

<b>TAULUKKO 2.1: C-SERVICEN HYÖDYT JA HAITAT</b>	<b>13</b>
<b>TAULUKKO 5.1: SILVERLIGHT SELAIN YHTEENSOPIVUUS</b>	<b>22</b>

## KÄYTETYT LYHENTEET ja SANASTO

.NET	Microsoftin kehittämä ohjelmistokomponenttikirjasto, joka mahdollistaa Windows-pohjaisten sovellusten ajamisen ja kehittämisen.
Agile	Ketterät menetelmät, joukko ohjelmistotuotantomenetelmiä, jotka keskittyvät suoraan viestintään ja nopeaan reagointiin.
BI	Business Intelligence mahdollistaa organisaation kerätä ja ylläpitää tietoa. Liiketoiminnallisen arvon parantamiseksi sitä käytetään mm. tulosten mittaamiseen, analysointiin, raportointiin ja tiedon sekä osaamisen hallintaan.
C#	C sharp on Microsoftin kehittämä ohjelmointikieli .NET-ympäristöön.
CRM	Customer Relationship Management, asiakkuudenhallinta ja siihen kuuluvat tietojärjestelmät.
ERP	Enterprise Resource Planning, toiminnanohjausjärjestelmä joka sisältää erilaisia integraatioita kuten tuotantoa, jake- lua ja varastonhallintaa.
HTML5	Hyper Text Markup Language eli HTML-merkkikielen viides versio. Käytetään modernien verkkosivujen ja sovel- lusten kehittämiseen.
JavaScript	Ohjelmointikieli, jota käytetään selaimessa käytettävien skriptien tekemiseen. Mahdollistaa verkkosivujen dynaami- suuden.
LEAN	Lean-ajattelu on tapa johtaa poistamalla kaikki turha, pa- rantamaan asiakastytyväisyyttä. Lean-ajattelu kuuluu ket- teriin menetelmiin.

LINQ	Language-Integrated Query mahdollistaa tietokantakyselyt .NET-ajoympäristössä.
MVC	Malli-Näkymä-Ohjain (Model-View-Controller), yleisin Malli-Näkymä -arkkitehtuurimallien menetelmistä.
MVVM	Malli-Näkymä-NäkymäMalli (Model-View-ViewModel), uusin Malli - Näkymä -arkkitehtuurimalli menetelmistä.
OOB	Out-of-the-Browser, selaimen ulkopuolella tapahtuva käyttäminen. Sovellus pystytään asentamaan työpöydälle.
SCRUM	Ketterien menetelmien (Agile) projektinhallinnan menetelmä.
Silverlight	Microsoft Silverlight on web-ohjelmointiympäristö, joka mahdollistaa rikkaiden internetsovelluksien kehittämisen.
SQL	Structured Query Language on kyselykieli, jolla tehdään relaatiotietokantaan erilaisia dataan liittyviä toimintoja, kuten lisäyksiä, poistoja ja tiedon hakuja.
TFS	Team Foundation Server, sovelluskehitysprojektin hallintajärjestelmä, joka mahdollistaa tiimityöskentelyn yhden tai useamman projektin kanssa.
UI	User Interface eli käyttöliittymä.
UxFactory	eCraft Oy Ab:n kehittämä lähdekoodikirjasto.
Visual Studio	Microsoftin ohjelmistonkehitysympäristö, joka mahdollistaa tehokkaamman tavan rakentaa sovelluksia eri alustoille.
WPF	Windows Presentation Foundation on DirectX-rajapinnan päälle rakennettu XAML-merkintäkieltä hyödyntävä lähdekoodikirjasto. Se sisältää moderneja käyttöliittymäominaisuuksia, joita hyödynnetään työpöytäsovelluksissa.

XAML	Extensible Application Markup Language on XML-pohjainen merkintäkieli, jota käytetään sovelluksien kehittämiseen Microsoftin eri alustoille.
XML	Extensible Markup Language on merkintäkieli, jota käytetään tiedonvälitykseen ja tallentamiseen.



# 1 JOHDANTO

Opinnäytetyön tavoite oli kehittää liiketoimintasovellus asiakasyritykselle käyttäen Microsoftin Silverlight-teknologiaa. Järjestelmän tarkoituksena oli yhdistää asiakasyrityksen vanhat tietojärjestelmät uuden sovelluksen avulla. Sovelluksen toteuttajana toimi ohjelmistoyritys eCraft Oy Ab ja kehitystyö suoritettiin heidän asiakkaalle Cargotec Oyj:lle. Sovelluskehitystyö aloitettiin kesäkuussa 2012 ja projektin kesto oli viisi kuukautta. Kehitystyön suoritti eCraftin Key Account -projektitiimi, johon kuului ohjelmistoarkkitehti, tiimimanageri sekä kaksi sovelluskehittäjää. Sovellus, jota kutsutaan nimellä C-Service, on toteutettu käyttäen Microsoft Silverlight -teknologiaa sekä MVVM-arkkitehtuuria (Model-View-ViewModel) (eCraft Oy Ab 2013b).

Projektin lähtökohtana oli mahdollistaa Cargotec Oyj:lle sekä heidän asiakkailleen tehokkaampi tapa tehdä tilauksia, parantaa asiakkaiden käyttökokemusta sekä vähentää ylläpitokustannuksia. Nykyiset järjestelmät toimivat ylläpitosopimuksella, mikä tarkoittaa päivittäistä kommunikointia asiakkaiden kanssa, virheiden selvittämistä sekä lähdekoodin korjaamista. Vanha järjestelmä oli toteutettu käyttäen ActiveX-teknologiaa, jota on käytetty ja tutkittu oppimisprosessina uuden järjestelmän kehittämiseksi (eCraft Oy Ab 2013b).

Opinnäytetyössäni vertailen MVVM-arkkitehtuuria vastaavanlaisiin ohjelmistoarkkitehtuurimalleihin sekä selvitän MVVM-ohjelmistoarkkitehtuurin hyötyjä C-Service sovelluskehitystyössä. Arvioin projektinhallinnan näkökulmasta ketterissä menetelmissä käytettyä SCRUM projektinhallinnan menetelmää ja kerron miten se mahdollisti eCraft Oy Ab:n Key Account projektitiimille tehokkaan tavan työskennellä.

## 2 ECRAFT OY AB JA KEY ACCOUNT ESPOO

eCraft on suomalainen, vuonna 1999 perustettu asiantuntijayritys, jonka pääasiallinen tehtävä on tarjota asiakasyrityksille toiminnanohjauksen parantaminen. eCraftin liikevaihto oli vuonna 2012 noin 12 miljoonaa euroa ja sillä on toimipisteitä kolmella paikkakunnalla – Espoossa, Vaasassa sekä Malmössä. Toimitusjohtajana toimii Juha Koivisto (eCraft Oy Ab 2013e).

eCraft toimittaa toiminnanohjauksen ydinjärjestelmiä, kuten ERP, CRM ja BI-järjestelmiä, ja tuottaa niiden päälle käyttäjäystävällisiä käyttöliittymiä. eCraft liiketoiminnan tunnuslause on ”We create less software” (luomme vähemmän ohjelmistoa). Tunnuslause tarkoittaa, että vähennetään asiakkaiden päävaivaa kehittämällä heille juuri sitä mitä heidän liiketoimintansa tarvitsee. Vaikeat järjestelmät hankaloittavat käyttökokemusta, tuottavat väärää tietoa sekä voivat johtaa yritysten ja heidän asiakkaidensa rahojen menetykseen. Hyvin tuotetulla järjestelmällä käyttäjät saavuttavat asetetut tavoitteet tehokkaasti, tuloksia syntyy ja yrityksen kulut pienenevät. Yksi syy, miksi yritys hankkii toiminnanohjausjärjestelmän, on tuottavuuden parantaminen. Huonon käytettävyyden vuoksi, tuottavuus voi jäädä osin saavuttamatta (eCraft Oy Ab 2013d).

eCraft Oy Ab:n vahvuus on sen ammattitaitoinen ja monipuolisesti sertifioitu henkilöstö. eCraft on sertifioitunut mm. Microsoft Gold Certified Partneriksi, Lawson Service Partneriksi, Lawson Product Partneriksi, QlickView Partneriksi sekä HP Gold Preferred Partneriksi. eCraft tukee työntekijöitään, jotta he voivat ylläpitää ammattitaitoaan ajan tasalla nopeasti kehittyvällä alalla antaen heille mahdollisuuden kouluttautua ja sertifioitua. Tämä on sekä yrityksen itsensä että sen työntekijöiden etu. Työntekijät kehittyvät työssään ja saavat henkilökohtaisia sertifikaatteja, joista yritys saa samalla ammattitaitoisia sertifioituja työntekijöitä (eCraft Oy Ab 2013a).

Projektityö tehtiin eCraft Oy Ab:n Key Account Espoo (KAE) tiimin kanssa. KAE-tiimin jäsenet vastaavat avainasiakkaiden tarpeista sekä järjestelmien ylläpitä-

misestä sekä ongelmien ratkaisusta. KAE koostuu henkilöistä, jotka omaavat vahvan Microsoft teknologiaosaamisen sekä asiakaslähtöisen ajattelutavan.

## 2.1 Avainasiakas Cargotec Oyj

Avainasiakkuudella tarkoitetaan yrityksen toiminnalle tärkeitä asiakkaita. Jokainen asiakas on tärkeä yrityksen liiketoiminnalle, mutta avainasiakkaat vastaavat suuresta osasta yrityksen liikevaihdoista. Key Account Espoo tiimi on asiakas-keskeinen tiimi, jonka tärkein lähtökohta on asiakas. Yrityksen tuotoksiin tyytyväinen asiakas takaa laadukkaan yritystoiminnan. Tämän vuoksi KAE-tiimi on päivittäisessä kontaktissa useiden avainasiakkaiden kanssa varmistaen, että asiakkaat pystyvät toteuttamaan omaa liiketoimintaansa parhaalla mahdollisella tavalla. Yksi suurimmista eCraft Oy Ab:n avainasiakkaista on suomalainen lastinkäsittelyratkaisuja toimittava Cargotec Oyj.

Cargotec Oyj on suomalainen, vuonna 2005 perustettu, lastinkäsittelyratkaisuja toimittava yritys. Maailmanlaajuinen Cargotecin verkosto toimii lähellä asiakkaita, ja kattavat huoltopalvelut takaavat laitteiden jatkuvan, luotettavan ja ympäristöystävällisen toiminnan. Cargotecin liikevaihto vuonna 2012 oli 3,3 miljardia euroa, ja konserni työllistää noin 10 000 henkilöä. Cargotecin B-sarjan osake on listattu NASDAQ OMX Helsingissä kaupankäyntitunnuksella CGCBV. Yrityksen pääkonttori sijaitsee Helsingissä. Cargotec tarjoaa ratkaisuja asiakkailleen tavaravirtojen käsittelyn tehostamisessa. Yrityksen toimitusjohtajana toimii Mika Vehviläinen (Cargotec Oyj 2013a).

Cargotec Oyj tehostaa tavaravirtojen kulkua niin maalla kuin merelläkin – kaikkialla, missä tavara liikkuu. Cargotecin brändit MacGregor, Kalmar ja Hiab tunnetaan johtavista lastin- ja kuormankäsittelyalan ratkaisuistaan ympäri maailman. Näistä Hiab toimittaa ja huoltaa kuormausnostureita, ajoneuvotrukkeja, takalaitanostureita sekä muita ajoneuvojen lastinkäsittelylaitteita. Kalmar toimittaa laitteita konttien sekä perävaunujen siirtelyyn. Tällaisia laitteita ovat kontti- ja kuljetuslukit sekä terminaalitruktorit. MacGregor toimittaa laivojen lastinkäsittelylaitteita, kuten lastinkiinnityslaitteita ja laivanostureita (Cargotec Oyj 2013a).

Cargotec Oyj tilasi eCraft Oy Ab:lta päivytetyn tilausjärjestelmämoduulin nykyiseen palvelujärjestelmään, C-Officeen. Uuden moduulin tarkoitus on päästä irti ongelmista nykyisessä järjestelmässä sekä yksinkertaistaa ja nopeuttaa asiakkaiden käyttökokemusta tilausvaiheessa (eCraft Oy Ab 2013c).

## 2.2 C-Office-tietojärjestelmä

C-Office on järjestelmäkokonaisuus, joka sisältää pienempiä järjestelmämoduuleita, joista yksi on C-Service. C-Office järjestelmä on Cargotec Oy:n toiminnanohjausjärjestelmä, jolla on useita päivittäisiä käyttäjiä. C-Office on verkossa toimiva portaali, jonka kautta käyttäjät kirjautuvat niihin palveluihin, joihin heidän käyttöoikeudet sallivat.

Uusi Silverlight pohjainen C-Service-palvelu tullaan implementoimaan osaksi C-Office-palvelua. C-Officen kirjautuminen mahdollistaa sen, että saadaan käyttäjän käyttöoikeudet tarkastettua C-Service palveluun kirjautuessa. Eri käyttäjillä on eri oikeudet eri tuotetietoihin (eCraft Oy Ab 2013b).

C-Service on .NET-pohjainen katalogi, joka sisältää Cargotec Oyj:n tuotteiden mallit, varaosat sekä piirustukset. Järjestelmän käyttäjät koostuvat Cargotecin henkilöstöstä sekä yhteistyökumppaneista, joiden tehtävänä on ostaa varaosia sekä korjata Cargotecin tuotteita ja välineitä.

C-Service-sovellusta on mahdollista käyttää kolmella eri tavalla:

- **C-Service .NET Client**, on työpöydälle asennettava sovellus. Työpöytäversion edut ovat siinä, etteivät käyttäjien tarvitse ladata kuin ne tuotetiedot, jotka ovat oleellisia heidän omalle työlleen. Lokaali C-Service on nopeampi tapa käyttää sovellusta ja se on toteutettu ActiveX-tekniikalla.
- **C-Service C-Officessa**, on verkossa toimiva "online client". Palvelun käyttäminen vaatii käyttäjältä verkkoyhteyden sekä kirjautumisen C-Office-palveluun.
- **C-Service upotettuna C-Parts-palveluun**, on tarkoitettu varaosien ostamiseen keskittyviin asiakkaiden tarpeisiin. C-Parts-palveluun vaatii verkkoyhteyden sekä kirjautumisen C-Office-palveluun.

C-Service-palvelu toimii samalla tavoin kaikkia kolmea tapaa käytettäessä. Käyttäjä itse määrittää, mitä tapaa haluaa käyttää sekä mikä näistä tavoista antaa hänelle itselleen parhaan mahdollisen tehokkuuden. Taulukossa 2.1 on listattu C-Servicen eri käyttötapojen hyötyjä ja haittoja (eCraft Oy Ab 2013c).

**Taulukko 2.1 C-Service hyödyt ja haitat.**

Tyyli	Hyödyt	Haitat
<b>C-Service Web Client</b>	Nopea kirjautuminen	Ei offline tukea
<b>C-Service Embedded C-Partsissa</b>	Nopea kirjautuminen, varaosien tilaaminen	Ei offline tukea
<b>C-Service .NET sovellus</b>	Offline mahdollisuudet	Tuotetietojen lataaminen työasemalle

C-Service Kalmar on toteutettu Silverlight-teknologialla ja toiminnallisuuksiltaan samantapainen kuin C-Service Web Client. Eroina näihin kahteen on lähinnä se, että Kalmar sisältää ainoastaan Cargotecin Kalmar-tuotteita.

C-Servicen uusin versio on nimeltään C-Service Silverlight. Sovellus on nimensä mukaisesti toteutettu Silverlight 5 -teknologialla. Uusin versio haluttiin toteuttaa kahdesta pääsystä, vanhentuneen teknologian uusimiseksi sekä vanhan C-Servicessä olevien ylläpidollisten ongelmien korjaamiseksi. Uusimmassa versiossa yhdistettiin ominaisuuksia vanhoista C-Service- ja Kalmar-sovelluksista sekä liitettiin mukaan niiden tuotekatalogit. Tämä mahdollisti käyttäjille yhtenäisen käyttöliittymän, joka helpottaa käyttäjien liikkumista eri tuotekategorioiden välillä (eCraft Oy Ab 2013b).

C-Servicen ylläpitämiseen ja kehittämiseen käytetään neljää Windows Server -palvelinta ja kahta Windows SQL Server -palvelinta. Palvelimilla ylläpidetään testi- ja tuotantosovelluksia sekä tuotantotietoja. Tuotantotiedolla tarkoitetaan yritysten käyttämiä tietoja, jotka ovat heidän tuotannossa käytössä. Tuotantotiedon muokkaamisessa on aina riskinsä ja siksi yleensä käytetään testipalvelimella olevaa testikantaa muutoksien testaamiseen. Ennen muutoksien implementointia tuotantokantaan tullaan se testaamaan testiympäristössä. Näin varmistetaan siitä, etteivät sovellukset hajoa myöhemmin, kun ne implementoidaan tuotantokantaan (eCraft Oy Ab 2013b).

## 3 SUUNNITTELUMALLIT

Ohjelmistotekniikassa käytetään suunnittelumalleja kuvaamaan suunnittelutason ratkaisuja ohjelmiston toteuttamisessa. Niitä käytetään minimoimaan ohjelmistovirheet ja tavoitteena on hyödyntää hyväksi todettuja ratkaisuja kehitystyössä. Suunnittelumallien kautta kaikilla on mahdollisuus kehittää omaa suunnittelutietämystään ja saada apua järjestelmien dokumentoinnissa. Suunnittelumallit ovat kuin rakennuspalikoita, joiden käyttöä helpottaa suunnittelijoiden yhtenäinen sanasto.

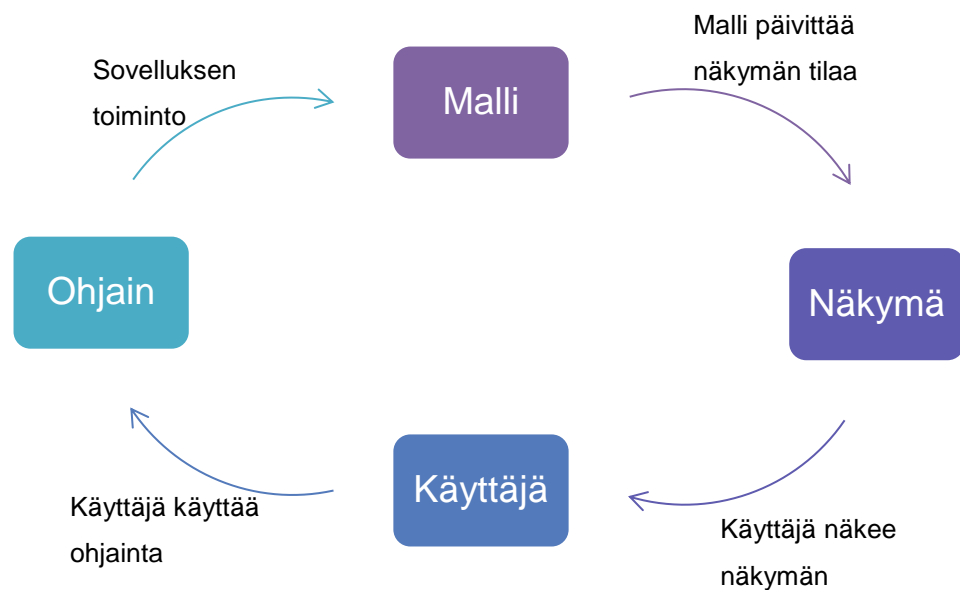
### 3.1 MVC-arkkitehtuuri

MVC-arkkitehtuuri on ohjelmistoarkkitehtuurimalli, joka muodostuu sanoista malli, näkymä ja ohjain (Model, View, Controller). Näistä kolmesta komponentista tärkein on malli, joka pitää huolen järjestelmän ydintoiminnallisuudesta. Mallikomponentti tarjoaa loogiset toiminnot, tiedot sekä se huomioi sovelluksen näkymät, jotka ovat yhteydessä sovelluksen tilaan. Tilan muuttuessa mallikomponentti ilmoittaa muille komponenteille tilan muutoksesta ja pitää huolen tiedon tallentamisesta, ylläpidosta ja käsittelystä.

Näkymä näyttää visuaalisesti mallikomponentin tiedot käyttäjälle. Tilan muuttuessa näkymäkomponentti päivittää näkymän eli sovelluksen ulkoasun ja sen miten tiedot esitetään sovelluksen käyttöliittymässä. Tilan muuttuessa käyttäjä pystyy juuri tästä näkymästä huomioimaan tilan muutokset. Ohjainkomponentti ottaa vastaan käyttäjän tekemät tilamuutokset eli toisin sanoen käyttäjän käytössä sovellusta, ottaa ohjain sovelluksen käskyt vastaan ja muuttaa niiden pohjalta mallia ja näkymää.

MVC-arkkitehtuurin etuja ovat, että malli voidaan suunnitella, ohjelmoida ja testata yksittäisenä komponenttina ilman vaikuttamista muihin komponentteihin. Mallikomponenttia voidaan käyttää useiden näkymäkomponenttien kanssa, jotka ovat sidottuja samaan tietoon; jos halutaan päivittää käyttöliittymää, ei tarvit-

se tehdä muutoksia itse malliin. Käyttöliittymän ulkoasun muokkaaminen on suhteellisen helppoa. Tämän arkkitehtuurimallin haittoihin lukeutuvat mahdolliset näkymien turhat päivityskutsut sekä tietokyselyt, jotka voivat lisätä suoritus-aikaa, koska malli ei välttämättä ole optimoitu ja tietokannasta saatetaan hakea useasti turhaa tietoa. Kuvassa 3.1 on esitetty MVC-mallin arkkitehtuuri (Siddigi & Vice, 2012).



**Kuva 3.1 MVC-malli**

### 3.2 MVVM-arkkitehtuuri

MVVM-arkkitehtuuri on Microsoftin kehittämä suunnittelumalli, joka koostuu sanoista malli, näkymä ja näkymämalli (Model, View, ViewModel). MVVM esiteltiin yleisölle vuonna 2005, kun Microsoftin Josh Gossmanin julkaisi blogissaan artikkelin Avalonista (aiempi nimitys Windows Presentation Foundationille), jossa käytettiin MVVM-arkkitehtuuria. MVVM on kehitetty MVC-arkkitehtuurin pohjalta ja keskittyy modernien käyttöliittymien kehitykseen, jotka tukevat tapahtumapohjaista ohjelmointi kuten HTML5, WPF ja Silverlight. MVVM kehitettiin tukemaan WPF- ja Silverlight-teknologiaa sekä niille ominaista Databinding-menetelmää, joka sitoo tietoa näkymäkomponentissa olevaan elementtiin.

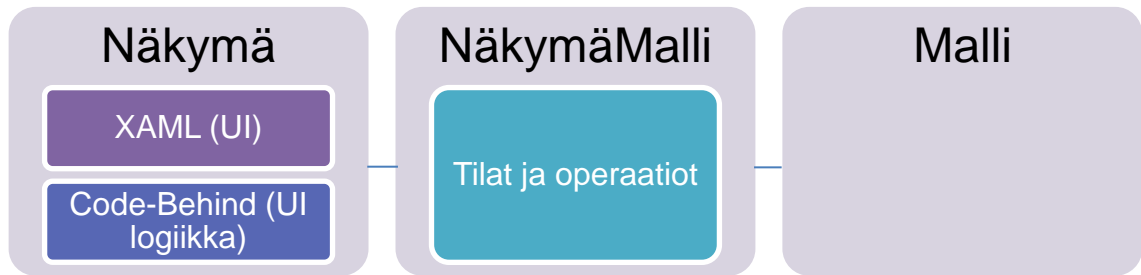
MVVM kehitettiin korvaamaan MVC- ja MVP-arkkitehtuurissa ilmenneitä rajoitteita sekä yhdistelemään niiden vahvuuksia (Siddigi & Vice, 2012).

Kun MVVM-arkkitehtuuria hyödynnetään oikein, se mahdollistaa paljon tapoja parantaa ja tehostaa sovelluskehitystä. Tällaisia hyötyjä ovat esimerkiksi laadun parantaminen, ylläpidettävyys ja lähdekoodin uudelleen käyttäminen sekä itse kehittäjän työtehokkuuden parantuminen. MVVM-arkkitehtuurin ansiosta uudet kehittäjät, jotka tuntevat MVVM-mallin, voivat liittyä kehitystyöhön nopeammin ja osaavat muutamalla silmäyksellä nähdä miten sovellus on rakennettu (Siddigi & Vice, 2012).

Tietyissä tilanteissa MVVM-arkkitehtuurin käyttäminen saattaa olla liioiteltua. Projektin luonteesta ja laajuudesta riippuen kannattaa ottaa huomioon onko kyseinen arkkitehtuurimalli oikea vai tulisiko käyttää jotain muuta mallia sovelluskehitystyöhön. Yksi tällaisista esimerkeistä on tilanne, jossa projekti on laajuudeltaan pieni eikä arkkitehtuurin käyttäminen tuolloin anna lisäarvoa sovelluksen suunnittelussa tai toteuttamisessa. MVVM-arkkitehtuurissa huonoja puolia on se, että siirryttäessä siihen perinteisimmistä arkkitehtuurimalleista (MVC), vaatii käyttö paljon oppimista sekä oppimisen tueksi yleensä suuremman luokan sovelluskehitysprojektin. Kuten kaikissa tehokkaissa arkkitehtuureissa ja työkaluissa, jos sovelluksessa käytettyä arkkitehtuuria on toteutettu huonosti, se vaikeuttaa ja hidastaa kehitystyötä. Lopputuloksena tulee hidas sovellus, joka ei taipu tarvittaessa tuleviin muutoksiin (Siddigi & Vice, 2012).

MVVM-mallissa mallikomponentti sisältää sovelluksen tietorakenteen, logiikan ja tiedon noutamisen sekä tallentamisen. Näkymä näyttää MVC-mallin tavoin visuaalisen tilan käyttäjälle. Näkymä on XAML:illa toteutettu käyttöliittymä, jonka elementit voidaan sitoa dataan. Näkymämalli taas yhdistää ja toimii välittäjänä sovelluksen malli- ja näkymäkomponenteille. Se toimii niin sanottuna tiedon välittäjänä, käyttäen mallissa olevia tietoja välittäen ne näkymälle sekä se välittää komentoja näkymästä mallikomponenttiin (Siddigi & Vice, 2012).





**Kuva 3.2 MVVM-arkkitehtuurimalli**

Näkymä ja Näkymämalli viestivät keskenään Databinding-menetelmän, metodien, ominaisuuksien sekä tapahtumien välityksellä. MVVM:n eri osat tunnistavat kuvassa 3.2 olevan mallin mukaan vain viereisen osan. Esimerkiksi mallikomponentti sisältää tietomallin käytettäväksi osaksi sovellusta, mutta se ei keskustele suoraan näkymän kanssa, vaan siirtää datan näkymämallille. Näkymämalli pitää huolen siitä, että data siirtyy oikeanlaisena, jotta näkymä pystyy sen näyttämään (MSDN, 2013a).

## 4 MENETELMIEN KÄYTTÄMINEN ASIAKASPROJEKTISSA

Menetelmien käyttäminen sovelluskehityksessä mahdollistaa helpon tavan työskennellä projektin parissa. Menetelmät voivat auttaa reagoimaan nopeammin asiakkaan muuttuviin vaatimuksiin tai esille tulleisiin ongelmakohtiin. Jokainen projekti on erilainen, mutta menetelmät mahdollistavat tietynlaisen peruskaavan, joka auttaa reagoimaan ongelmatilanteisiin. Väärien menetelmien seuraaminen hidastaa projektityöskentelyä. Varsinaisesti vääriä menetelmiä ei ole, vaan niitä tutkiessa kannattaa huomioida millainen menetelmä on sopiva juuri kyseessä olevan yrityksen tai projektin mieleinen. Tässä osiossa käsitellään muutamia menetelmiä, joita ilmeni C-Service Silverlight projektin kehityksessä (Shore & Warden 2008, 3–13).

### 4.1 Ketterät menetelmät (Agile)

Ketterät menetelmät sisältävät useita kehitystyöprojekteissa käytettäviä menetelmiä. Kaikille näille menetelmille yhtenäisiä piirteitä ovat toimivan ohjelmiston ensisijaisuus. Kehitystyön yhteydessä viestinnän tulisi olla suoranaista kehittäji- en ja asiakkaiden välillä ja tiimin tulisi pystyä nopeasti reagoimaan muutoksiin (Agile Manifesto 2001).

Ketterän menetelmän perusmääritelmät kehitettiin vuonna 2001, kun 17 ketterän kehityksen puolestapuhujaa kokoontui Utahissa keskustelemaan menetelmien yhtenäisestä perustasta. Kokouksessa julistettiin Agile Manifesto, joka on ketterän menetelmän perusmääritelmä. Manifesto koostuu neljästä arvosta sekä 12 periaatteesta (Agile Manifesto 2001).

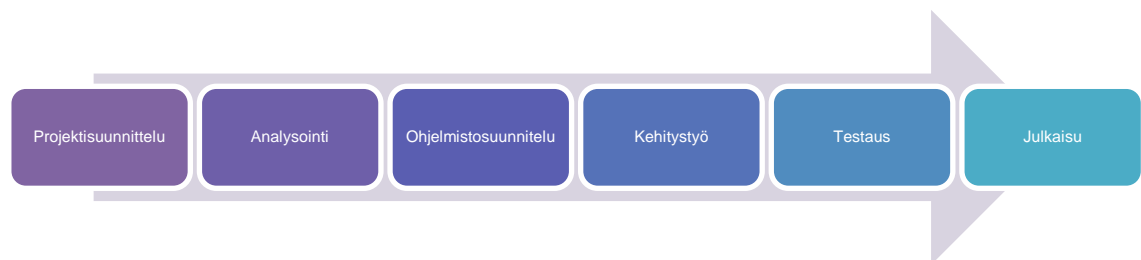
Agile Manifeston sisältö on seuraava:

*”Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme:*

- *Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja*
- *Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota*
- *Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja*
- *Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa*

*Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.” (Agile Manifesto 2001).*

Ketterissä menetelmissä tavoitteena on pienentää ohjelmistokehityksen riskejä jakamalla kehitystyö lyhyisiin iteraatioihin eli vaiheisiin. Iteraatioita toistetaan kunnes haluttu tulos saavutetaan ja jokainen iteraatio sisältää saman rakenteen vaikka sisältö muuttuisikin. Yhden iteraation kesto on minimissään viikon ja korkeintaan kuukauden mittainen. Syklin lopussa arvioidaan uudelleen projektin prioriteetit ja suunnitellaan seuraavaan iteraation sisältöä. Iteraation sykli on esitetty kuvassa 4.2. Ketterissä menetelmissä korostetaan, että projektin eteneminen on tärkein edistyksen mittari eikä dokumentaatiolla ole samanlaista painoarvoa kuin useissa muissa perinteisimmässä malleissa (Shore & Warden 2008, 3–15).



**Kuva 4.2: Iteraatio ohjelmistokehityksessä**

#### 4.2 Lean-malli

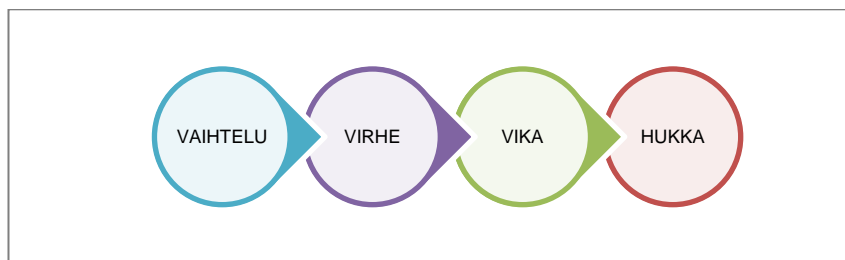
Yritystasolla eCraft käyttää Lean-mallia. Lean-malli on johtamisfilosofia, jonka kehitti Taiichi Ohno. Tämä johtamisfilosofia tarkoittaa, että yrityksen tulee huomioida asiakkaan tarpeet erityisen hyvin tarjotakseen tälle parhaan mahdollisen lisäarvon. Asiakkaalle tärkeintä ovat asiat, jotka tuottavat heille parhaat rahan-

arvoiset hyödyt. Lean ajattelussa pyritään tunnistamaan liiketoiminnan arvoketju ja poistamaan turhat toiminnalliset asiat ja näin parantamaan laatua sekä asiakastytyväisyyttä. Yksi parhaimmista esimerkeistä, jossa yritys on hyödyntänyt Lean-mallia ja kasvanut pienestä yrityksestä maailman suurimmaksi autovalmistajaksi on Toyota. Toyota huomioi tuotannossa ilmenneitä ongelmia ja tuotteiden korjaamisen sijaan he keskittyivät tuotantoprosessien seurantaan (Six Sigma 2013).

Lean-malli on vuosien mittaan kehittynyt tuoden uusia työkaluja prosessien tunnistamiseen, joiden toiminnoissa ilmenee tuotannon menetystä. Suurin osa Lean-toiminnoista on sisäisten käytäntöjen muutosta, joka tarkoittaa henkistä vakautta sekä toimintatavan muutosta, jonka tarkoituksena on luoda uusia käytäytymis- ja johtamistapoja. Lean-työkalut voidaan jakaa näkymättömiin ja näkyviin. Näkyviä työkaluja ovat käytännöt, tekniikat ja periaatteet, jotka pohjautuvat näkymättömiin työkaluihin kuten ajattelu- ja toimintarutiineihin (Six Sigma 2013).

Yrityksen toimintaa ja siihen kuuluvia toimitusketjuja tulisikin tarkastella kokonaisuutena siten, että johto ja esimiehet ymmärtävät mitä ollaan tekemässä. Prosessien parantaminen ei ole projekti- tai kampanjaluonteista, kuten perinteisessä johtamisessa on tapana ajatella. Muutoshakuisuutta pidetään erillisenä tapahtumana, joka lisätään johtamiseen. Lean-johtamisessa tämä on asia, jossa syvennyttään systeemiin kokonaisvaltaisesti ja pitkäjänteisesti. Muutokset liitetään osaksi yrityskulttuurin ja otetaan käyttöön koko organisaatiossa. Lean-menetelmät perustuvat oppimiseen, joten henkilöstöä tulee kouluttaa ja uusia järjestelyjä ja työkalujen käyttöä tulee tukea yrityskulttuurissa. Ilman kunnollista koulutusta ihmiset palaavat takaisin vanhoihin toimintatapoihin eikä muutos tule olemaan pysyvää. Lean-johtaminen sisältää yhdessä sovitut tavat, käytännöt ja työkalut. Ihmisiä tarvitsee motivoida muuttamaan toimintatapojaan ja hankki-  
maan tietoa siitä, miten muutetaan. Lean-työkalut ovat apuvälineitä, mutta ne eivät määrittele muutoksen nopeutta, vaan auttavat tavoitteen määrittelemisessä (Six Sigma 2013).

Tärkeintä Lean-menetelmien käytössä on tunnistaa ja poistaa tuotannon menetyksiä nopeasti ja tehokkaasti pienentäen yrityksen kustannuksia sekä parantamaan laatua. Menetyksiä tulee yrityksen toiminnoissa olevista vioista ja virheistä, joita vaihtelu aiheuttaa. Jos yrityksessä yritetään poistaa ainoastaan tuotannon menetykset, eikä syitä niiden syntyyn selvitetä, tapahtuu tuotannon menetyksiä todennäköisesti myöhemmin uudelleen. Jos keskitytään vaihtelun minimointiin, vähenee myös tuotannon menetyksien syntyminen. Kuvassa 4.3 tarkastellaan miten ongelma tunnistetaan (Six Sigma, 2013).



**Kuva 4.3: Ongelman tunnistaminen**

Tavoitteena on tehostaa yrityksen toimintaa sekä muokata siitä sopeutuva. Nopeuden kasvattaminen lisää myös yrityksen ketteryyttä ja yritys pystyy ratkomaan oikeat ongelmat saavuttaakseen ketterämmän tilan (Six Sigma 2013).

### 4.3 SCRUM

Scrum on tuotteiden kehittämiseen ja ylläpitoon tarkoitettu projektinhallinnan menetelmä, joka on käytössä Agile- eli ketterässä ohjelmistokehityksessä. Scrum kehitysprosessin idean kehitti Hirotaka Takeuchi ja Ikujiro Nonaka vuonna 1986. Kehitysprosessissa toimii ryhmä, joka koostuu laajan ammattitaidoin omaavista henkilöistä. Ryhmä suorittaa kehitysprosessin vaiheittain alusta loppuun. Varsinaisen Scrum-menetelmän kehittäjinä pidetään Jeff Sutherlandia, Ken Schwaberia, John Scumniotalesia ja Jeff McKennaa (Sutherland & Schwaber 2011).

Scrum-menetelmän avulla ratkaistaan monimutkaisia ongelmia tuotekehityksessä. Scrum ei ole yksittäinen kehitysprosessi, vaan menetelmä joka sisältää

monia prosesseja ja tekniikoita. Se helpottaa menetelmien vaikutuksien seuraamista, mikä taas helpottaa niiden parantamista. Scrum käyttää toistavaa ja lisäävää lähestymistapaa saadakseen tarkemman ennustettavuuden sekä hallitakseen riskejä (Sutherland & Schwaber 2011).

Scrum koostuu Scrum-tiimistä ja sen eri rooleista, tapahtumista, tuotoksista ja säännöistä. Scrum-tiimissä on tuoteomistaja, kehitystiimi ja Scrum master. Tiimissä on monitaitoisia henkilöitä, jotka itse päättävät kuinka työt tulisi tehdä, ilman ulkoista ohjausta. Scrum-tiimi noudattaa mallia, jossa optimoidaan joustavuus, luovuus ja tuottavuus (Sutherland & Schwaber 2011).

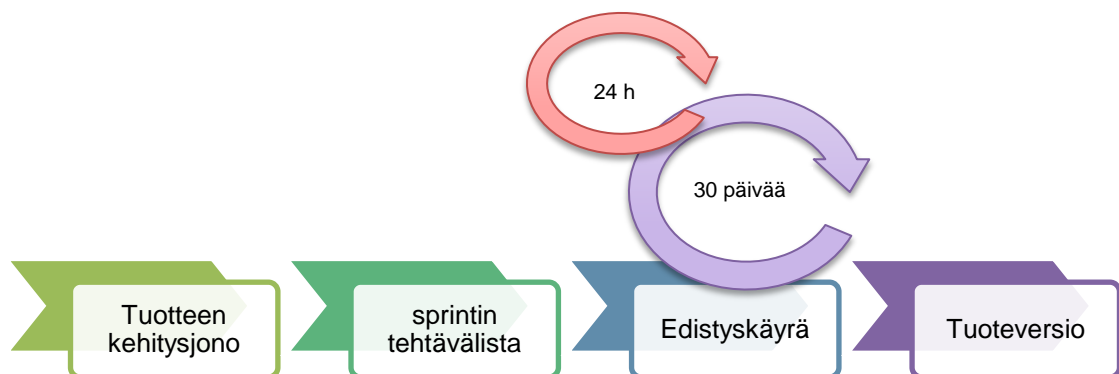
Tuoteomistaja vastaa tuotteen arvon ja kehitystiimin työn arvon maksimoimisesta sekä on vastuussa tuotteen kehitysjonon hallinnasta. Kehitysjonon hallinta tarkoittaa, että tuotteen kehitysjonon kohdat ovat merkittyjä ja ne on järjestetty saavuttamaan asetetut tavoitteet. Kehitystiimin työ on maksimoitu siten, että se pystyy kehitysjonosta ymmärtämään, mitä tehdään seuraavaksi. Tuoteomistaja on vastuussa siitä, että työt tulevat tehdyksi. Kehitystiimin ei myöskään tule reagoida muiden kuin tuoteomistajan asettamiin vaatimuksiin (Sutherland & Schwaber 2011).

Kehitystiimi koostuu ammattilaisista, jotka kehittävät valmiin julkaisukelpoisen tuotteen jokaisen sprintin aikana. Tiimin tulisi olla tarpeeksi pieni, jotta se on ketterä, mutta tarpeeksi suuri, jotta se saa suuren määrän työtä tehtyä. Kehitystiimi on ainoa, joka osallistuu tuoteversion kehitykseen. Tiimillä ei ole pienempiä alatiimejä, jotka hoitaisivat testaamisen tai liiketoiminta-analyysin, vaan hoitavat itse tämän kaiken (Sutherland & Schwaber 2011).

Scrum master pitää huolen siitä, että kaikki ymmärtävät ja käyttävät Scrumia, niin kehitystiimit ja tuoteomistaja kuin myös ulkopuolinen organisaatio. Scrum master auttaa ulkopuolista organisaatiota ymmärtämään, mitkä tavoista toimia ovat hyödyllisiä ja mitkä eivät. Tiimien tulee osata Scrumin teoria, käytännöt ja säännöt. Scrum master palvelee tuoteomistajaa ehdottamalla tekniikoita kehitysjonon hallintaan ja kommunikoi tuoteomistajan ja kehitystiimin välillä. Scrum master helpottaa kehitystiimin työtä ja auttaa valmentamalla oma-

alaitteisuuteen ja moniosaamiseen. Organisaatiota Scrum master valmentaa Scrumin käyttöönotossa ja suunnittelee toteutustavan organisaation sisällä. Scrum master aiheuttaa muutoksia, joilla kasvatetaan tuottavuutta. Scrum masterit työskentelevät toistensa kanssa parantaakseen Scrum-tehokkuutta organisaatiossa (Sutherland & Schwaber 2011).

Scrumia käytetään kaikkien muiden kuin Scrum-tapahtumien vähentämiseen. Tapahtumat ovat suunniteltua siten, että niillä on aikaraja antamaan tarpeeksi aikaa suunnittelulle ilman, että pääsee syntymään tehottomuutta. Scrum-menetelmien keskiössä ovat ns. sprintit. Sprintit koostuvat suunnittelupalaverista, päiväpalaverista, kehitystyöstä, sprinttikatselmuksesta sekä sprintin katselmuksesta. Jokaisen sprintin päätyttyä julkaistaan uusi tuoteversio, jonka jälkeen aloitetaan uusi sprintti. Sprintin aikana ei tehdä muutoksia, jotka vaikuttavat sprintin tavoitteeseen eikä laatutavoitteita lasketa. Sprinttien tulisi olla maksimissaan kuukauden pituisia sekä sisältää määritelmän siitä, mitä tullaan toteuttamaan. Kuvassa 4.4 on nähtävissä Scrum-prosessi (Sutherland & Schwaber 2011).



**Kuva 4.4: Scrum-prosessi**

#### 4.4 Menetelmien käyttö eCraft Oy Ab:ssä

eCraftilla ei käytetä mitään menetelmää sen kokonaisessa tarkoituksessaan. Yrityksessä on sisäisesti keskusteltu, mitä menetelmiä tulisi käyttää ja tulisiko valita jokin tietty kaava, jota noudattaa laadun ja tehokkuuden parantamiseksi. Yritystasolla eCraft käyttää enemmän Lean-menetelmiä kuin Scrum-menetelmiä. Kehitystyössä saatetaan käyttää toisia menetelmiä riippuen projektin luonteesta tai siihen osallistuvista henkilöistä. C-Service Silverlight-projektin etenemisessä yritetään noudattaa sprinttejä, joiden avulla seurataan kehitystyön etenemistä. Näin myös tiimin muut jäsenet, jotka eivät ole projektityössä mukana, pääsevät tietoiseksi siitä mitä muut henkilöt parhaillaan tekevät.

Tiimin sisäisissä keskusteluissa pohdittiin, tulisiko valita jokin tietty tapa työskennellä vai onko parempi olla sitomatta itseään mihinkään tiettyyn tapaan työskennellä. Projektien ja asiakkaiden erilaisuuden vuoksi menetelmiä ei voida kokonaisuutena hyödyntää, vaan yrityksessä noudatetaan ns. ”best practises” -ajattelua ja valitaan parhaat tavat työskennellä tiettyjen asiakkaiden kanssa.



## 5 TEKNOLOGIA

Teknologia osiossa käsittelen C-Service Silverlight projektissa käytettyjä kehitysalustoja, apusovelluksia sekä muita projektissa käytettyjä lähdekirjastoja ja ohjelmointikieliä. Kerron sovelluskehityksessä käytetyistä työkaluista, joilla oli projektin kehittämisessä suurin vaikutus. Osiossa käsitellään Silverlight-teknologiaa ja sen aikaisempia versioita. Vertailen myös Silverlightia sen ns. isosiskoon, Windows Presentation Foundationiin.

### 5.1 Silverlight

Silverlight on Microsoftin kehittämä ohjelmistokehys, jonka avulla voidaan rakentaa RIA (Rich Internet Application) eli ns. rikkaita internetsovelluksia. Rikkaissa internetsovelluksissa on työpöytä sovelluksille ominaisia piirteitä. Tämä mahdollistaa paremman ja kehittyneemmän käyttökokemuksen sovelluksen käyttäjälle. Silverlight-ajoympäristöä käytetään verkkoselaimen liitännäisenä, mutta kaikki selaimet eivät tue Silverlight-liitännäisiä. Tällaisia selaimia ovat esimerkiksi tablettilaitteiden selaimet kuten Windows RT:n Internet Explorer tai iPadin Safari. Taulukko 5.1 havainnollistaa käyttöjärjestelmän ja selainten yhteensopivuutta Silverlight-liitännäisen kanssa (Bugnion 2010, 7–11).

Useat yritykset hyödyntävät Silverlight-teknologiaa palvelutarjonnassaan. Tällainen on esimerkiksi MTV3, jonka Katsomo-palvelu tarjoaa monia kanavan ohjelmia katsottavaksi jälkepäin Silverlight-pohjaisen sovelluksen kautta. Toinen tunnettu yritys on Suomessa vähän aikaan sitten julkaistu Netflix, joka tarjoaa laajan valikoiman elokuvia ja sarjoja verkkosovelluksen kautta.

Taulukko 5.1: Silverlight yhteensopivuus eri selainten ja käyttöjärjestelmän.

Lähde: Microsoft

OS	IE 10	IE 9	IE 8	IE 7	IE 6	Firefox 3.6+	Safari 4+	Chrome 12+
Windows 8 työpöytä	OK	-	-	-	-	OK	-	OK
Windows Server 2012	OK	-	-	-	-	OK	-	OK
Windows 7	-	OK	OK	-	-	OK	-	OK
Windows 7 SP1	-	OK	OK	-	-	OK	-	OK
Windows Server 2008 SP2	-	-	-	OK	-	OK	-	OK
Windows Server 2008 R2 SP1	-	OK	OK	-	-	OK	-	OK
Windows Vista	-	OK	OK	OK	-	OK	-	OK
Windows Server 2003, Windows XP SP2, SP3	-	-	OK	OK	-	OK	-	OK
Macintosh OS 10.5.7+	-	-	-	-	-	OK	OK	-

Ensimmäinen Silverlight versio julkaistiin 2007. Se oli vielä todellinen raakile verrattaessa millaiseksi se on nykypäivänä kehittynyt. Silverlight 1 ei tukenut vielä ollenkaan .NET-koodia, vaan kaikki ohjelmointi toteutettiin käyttäen JavaScriptiä. Ensimmäisessä versiossa pystyttiin tosin jo käyttämään XAML-merkkaukieltä määrittämään käyttöliittymän ulkonäköä. Silverlight 1 ei sisällynyt valmiita käytettäviä kontroleita, joita voitaisiin käyttää apuna sovellusten tekemisessä. Muodot, kuten nelikulmio ja ympyrä, sisältyivät Silverlightin ensimmäiseen versioon, ja niitä voitiin käyttää kontrollien tekemiseen. Ensimmäi-

sen version tarkoitus oli luoda pohja Silverlight-liitännäiselle ja olla ensiaskel kohti rikkaampia internetsovelluksia (Bugnion 2010, 15–18).

Silverlightin toinen versio julkistettiin vuonna 2008. Uusi versio toi mukanaan useita suuria muutoksia aiempaan ensimmäiseen versioon nähden. Muutoksista suurin oli .NET-pohjaisen ohjelmoinnin lisääminen Silverlight-teknologiaan. Tämä mahdollisti .NET-sovelluksien tuomisen myös muille alustoille Windowsin lisäksi. Se myös sisälsi valmiita kontroleita, joita voitiin käyttää sovelluskehitystyössä. Monia ominaisuuksia joita käytetään Silverlightin uusimmassa viidennessä versiossa, käytettiin jo Silverlight 2 aikana. Se siis sisältää useita hyödyllisiä ominaisuuksia (Bugnion 2010, 15–18).

Kolmas versio julkaistiin vuonna 2009. Se sisälsi kattavamman valikoiman kontroleja ja ominaisuuksia, joita pystyttiin hyödyntämään sovelluskehitystyössä. Kehityksen seurauksena Silverlight-teknologiaa oli mahdollista hyödyntää paremmin liiketoimintasovelluksissa. Kolmannessa versiossa mahdollistettiin myös 2D-elementtien muokkaaminen 3D-elementin näköiseksi. Ominaisuutta nimitettiin usein myös 2.5D-ulottuvuudeksi. Tässä versiossa esiteltiin ensimmäisen kerran ominaisuus, joka mahdollisti selaimen ulkopuolella tapahtuvan käyttämisen eli ns. ”Out-of-the-browser” -ominaisuuden (OOB). Verkkoselaimessa käytettävä sovellus oli mahdollista asentaa ja käyttää myös työpöydältä (Moroney 2009, 3–10).

Neljäs versio julkaistiin kesällä 2010. Silverlight 4 mahdollisti uusien ominaisuuksien käyttämisen sovelluskehitystyössä. Näitä ominaisuuksia oli mm. Google Chrome-selain tuki, webkameran ja mikrofonin käyttäminen sovelluksessa sekä tulostusominaisuus. Neljäs versio tuki useita eri ohjelmointikieliä, joita aikaisemmat versiot eivät tukeneet. Näitä ohjelmointikieliä olivat mm. Iron Python, Iron Ruby sekä F#. Käyttöliittymäelementtejä ja -työkaluja parannettiin tukemaan kaksisuuntaista tekstiä, rikkaita tekstilaatikoita sekä mahdollistettiin HTML:n upottamista Silverlight sovellukseen. Silverlight 4:n OOB parannukset mahdollistivat tiedostojärjestelmän paremman ja turvallisemman hyödyntämisen sovelluksessa (Bugnion 2010, 15–18).

Nykyinen Silverlight versio julkaistiin joulukuussa 2011. Uusimpiin ominaisuuksiin kuuluvat konekiihdytetyt 3D-grafiikat sekä 64-bittinen selaintuki. Muita uusia ominaisuuksia olivat mm. laajemmat kontrollit, suoritustehon parantuminen, vektoripohjainen tulostus sekä etäkäytön mahdollisuudet. Työskentelyä on myös helpotettu siten, että viidennessä versiossa on mahdollista asettaa keskeytyskohta Visual Studiassa suoraan datasidottuun elementtiin. Joten virheiden etsiminen lähdekoodista sidottujen elementtien suhteen on helpottunut. Median käyttämistä Silverlight-sovelluksessa on parannettu. Suojaamatonta h.264-mediaa voidaan kiihdyttää käyttämällä tietokoneen laitteistoa nopeuttamaan mediaa siten, ettei äänenlaatu huonone (Bugnion 2010, 15–18).

## 5.2 Silverlight ja Windows Presentation Foundation

Windows Presentation Foundationia (WPF) kutsutaan Silverlightin isosiskoksi. Tämä johtuu siitä, että Silverlight on WPF:stä kevennetty ja alustariippumaton versio. WPF:n yksi rajoittavista tekijöistä onkin se, että se toimii ainoastaan Windows-käyttöjärjestelmän kanssa, kun taas Silverlight on alustariippumaton. Silverlightin aikaisempi nimi olikin Windows Presentation Everywhere. Ja koska Silverlightia kutsutaan WPF:n pikkusiskoksi, on sillä mahdollista kehittää helposti rikkaita internetsovelluksia. WPF ja Silverlight kuitenkin eroavat toisistaan muutamilla eri tavoilla. Silverlight ei pysty tarjoamaan kaikkia haluttuja ominaisuuksia, joita on WPF:ssä tai jotkut ominaisuudet joudutaan rakentamaan eri tavalla. Nämä teknologiat ovat kuitenkin kehittymässä kokoajan lähemmäksi toisiaan. Koska Silverlight on kevennetty versio WPF:stä, on se myös helpompi teknologia omaksua (Bugnion 2010, 14).

## 5.3 UX Factory lähdekoodikirjasto

UxFactory on eCraftin kehittämä ohjelmointikoodikirjasto, joka sisältää valmiitamoduuleja. UxFactory on suunniteltu tukemaan MVVM-suunnittelumallia projektien kehittämissä. UxFactory mahdollistaa nopean sovelluslogiikan implementoinnin MVVM-malliin sekä käyttöliittymän rakentamisen yhdistelemällä eri

moduuleja. Projekti kustomoidaan asiakaskohtaisesti liittämällä moduuleita, jotka sisältävät erilaisia navigointi-, haku- ja listausmoduuleja. Käyttöliittymän ulkoasua voidaan muokata vapaasti riippumatta siitä millaisia moduuleja projektiin implementoidaan.

UxFactory mahdollistaa paremman keskittymisen useampaan asiakasprojektiin samaan aikaan, koska järjestelmien perusominaisuudet on jo valmiiksi rakennettu. ECraftilla on oma sisäinen wiki-sivusto, joka sisältää dokumentaatiota siitä miten UxFactory pitää sisällään ja miten se tulisi implementoida projekteihin.

## 6 SILVERLIGHT SOVELLUKSEN TUOTTAMINEN AVAINASIAKASYRITYKSELLE

Sovelluksen tuottaminen avainasiakasyritys Cargotec Oyj:lle aloitettiin toukokuussa 2012. C-Service -sovelluksen päivittäminen oli ollut jo suunnitteilla aikaisemmin, mutta kehitystyön aloittamisesta päätettiin toukokuussa. Päivityksen tarkoituksena oli päivittää vanhasta ActiveX-sovelluksesta uudempaan Silverlight 5 -versioon. Johtuen ActiveX-sovelluksen suurista ylläpitokustannuksista päätettiin se korvata uudella Silverlight-sovelluksella, joka poistaisi suurimmat käyttöongelmat C-Servicen verkkosovelluksessa. Tässä osiossa käsittelen projektin etenemisen kulmakiviä ja sitä miten suunnittelu aloitettiin ja miten projekti-työ rakennettiin.

eCraft sai tilauksen Cargotec Oyj:ltä varaosanhankintatyökalulle. Tilauksessa määriteltiin olemassa olevien järjestelmien uudistaminen uudella työkalulla. Tarve uudelle työkalulle kartoitettiin suurien ylläpitokulujen vuoksi sekä sovelluksen vanhentuneen teknologian uudistamiseksi. Projektin tavoitteena oli yhdistää C-Servicen Hiab ja Kalmar verkkosovellukset. Sovelluksien yhdistäminen mahdollisti tehokkaan tavan korjata niissä ilmenneitä ongelmia. Tällaisia ongelmia olivat mm. sovelluksen käynnistämisen vaikeus, joka johtui sen vanhentuneesta teknologiasta.

Ensimmäiset keskustelut Silverlight-projektin toteutumisesta käytiin kehitystiimin managerin kanssa työsuhteen alussa. eCraft oli saanut pyynnön selvittää projektin laajuus ja siihen kuluva työmäärä. Samalla valmisteltiin C-Servicen kehitykselle kaksivuotissuunnitelma. Suunnitelmassa esitettiin ehdotus sovelluksen ylläpidolle sekä C-Service-työpöytäsovelluksen kehittämistä tulevaisuudessa. C-Service-työpöytäsovellus on käyttäjän tietokoneelle tallennettu versio ja sillä on kahdenlaisia käyttäjiä: normaalikäyttäjä, joka on yleensä Cargotecin työntekijä ja Cargotecin asiakas tai editoija, joka on Cargotecin työntekijä. Editoijan rooliin kuuluu tuotetietojen tekeminen, muokkaaminen ja poistaminen.

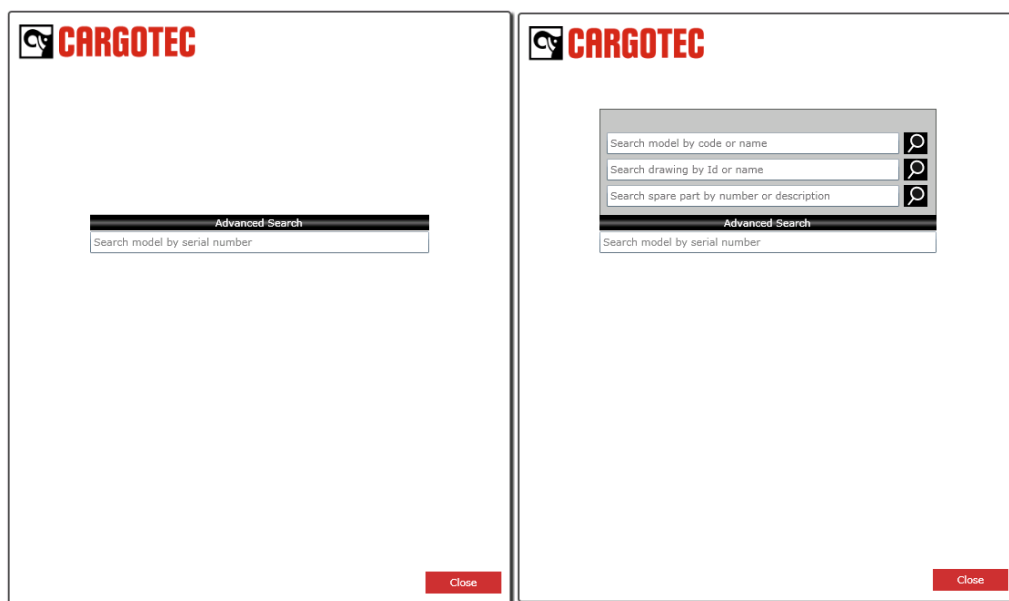
Projektin aloitusvaiheessa pidettiin ns. Workshop-päivän, jonka tavoitteena oli kehittää luonnos C-Servicen uudesta käyttöliittymästä. Päivän aikana toteutettiin selvitys sovelluksen pakollisista ominaisuuksista. Selvitystyö sisälsi C-Servicen vanhan lähdekoodin tarkastelun ja selvityksen voidaanko lähdekoodia käyttää uudelleen. eCraft antoi ratkaisuehdotuksen, jossa esitettiin näkemys sovelluksen kehittämiseen käytettävästä työmäärästä sekä suunnitelma varsinaisesta kehitystyöstä. Workshop-päivän palaveriin osallistui eCraftin projektitiimi sekä Cargotec Oyj:n Service Manageri sekä Business Partneri. Workshop-päivän aikana valmistui alustava projektisuunnitelma sekä luonnos käyttöliittymästä ja projektissa käytettävästä teknologiasta.

Workshop-päivän jälkeen projektitiimi kokoontui aloituspalaveriin, jossa tarkoituksena oli selventää kaikki projektin kehitystyöhön sisältyvät asiat. Tällaisia asioita olivat mm. testi- ja tuotantopalvelimet, sovellusten lähdekoodien säilytyspaikat. Lisäksi varmistettiin kehittäjien tarvittavat käyttöoikeudet. Projektissa käytettiin eCraftin omaa lähdekoodikirjastoa, jota kutsutaan UXFactory -nimellä. Tämä lähdekoodikirjasto mahdollistaa valmiiden komponenttien ja käyttöliittymäelementtien lisäämiseen projekteihin helposti. UXFactoryä käyttämällä saimme kaiken hyödyn yrityksen koodikirjastosta sekä pystyimme pystyttämään projektin MVVM-arkkitehtuurimallin mukaan.

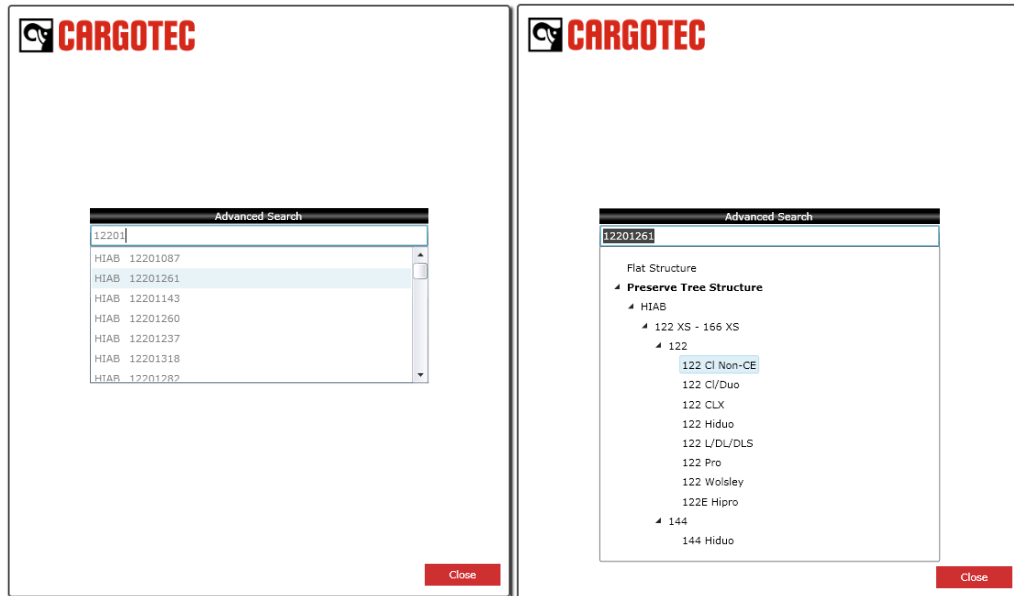
Projektitiimi tutki C-Servicen vanhaa lähdekoodia saadakseen paremman yleiskuvan projektin laajuudesta sekä siihen kuuluvista ominaisuuksista. Selvityksen lopputuloksena pystyttiin implementoimaan valmiita komponentteja nykyiseen Silverlight-sovellukseen. Tällaisia uudelleen käytettäviä komponentteja olivat mm. käyttäjän tunnistus kirjautumisvaiheessa, ostoskori, tietotaulukko sekä puuhierarkia tietojen listausta varten. Suurin osa aikaisempien C-Service-sovellusten lähdekoodeista ei ollut yhteensopivia uuden projektin kanssa.

Projektin edetessä asiakas antoi palautetta uusista toiminnoista, joita haluttiin osaksi uusinta versiota. Isoimmat muutokset olivat uusien tuotesarjanumeroiden implementointi osaksi C-Service-palvelua sekä hakutuloksien esittäminen uudella tavalla. Sarjanumeroiden implementoinnin vaikeus johtui siitä, että ne sijaitsivat eri tietokannoissa eikä niissä ollut samoja tunnistetietoja kuin C-

Serviceen jo implementoiduilla tuotetiedoilla oli. Asian korjaamiseksi jouduttiin luomaan ns. kartoitustaulu, jonka avulla pystyttiin yhdistämään sarjanumerot malleihin ja varaosiin. Samalla kehitettiin uusi hakutoiminto, joka antaa käyttäjän hakea varaosia sarjanumeron, mallin, osien tai kuvien mukaan. Tulokset esitetään joko puunäkymänä tai taulukkona riippuen siitä, mitä tietoa haetaan. Käyttäjä voi myös halutessaan hakea malleja ns. tasaisen rakenteen avulla, jolloin kaikki mahdollinen tieto mikä sarjanumeroon kuuluu, näytetään listana. Kuvassa 6.1 esitetään hakutoiminnan ominaisuuksia ja miten hakutulokset listataan puunäkymässä käyttäjän tarkasteltavaksi.

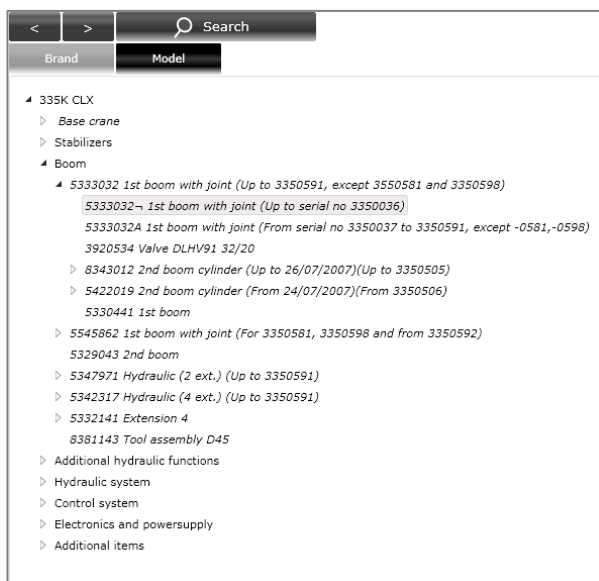






**Kuva 6.1: Hakutoiminto**

Käyttäjä näkee vain ne tuotetiedot, joihin hänen käyttäjäoikeutensa riittävät. Tuotetiedot listataan omien brändien mukaan puunäkymään. Puunäkymässä voidaan liikkua alemmalle tasolle katsomaan tietyn brändin mallia tai tietyn mallin yksittäistä varaosaa. Puunäkymä koostuu rekursiivisesti listatuista tuotteista, joiden hierarkian ylimpänä on brändi ja alimpana ovat varaosat. Puurakenteen kehittäminen Silverlight-teknologialla oli helppoa XAML:ssa käytettävien Treeview-elementtien kanssa. Treeview-elementti sidotaan (Databinding) haluttuun dataan, jolloin saadaan kuvan 6.2 mukainen tulos. Puunäkymä-elementit, kuten monet muutkin eivät kuulu Silverlightin peruskontrolleihin, vaan ne saavat käyttöönsä asentamalla Microsoftin kehittämän Silverlight Toolkit -elementtikirjaston (Bugnion 2010).



## Kuva 6.2: Puunäkymä

Käyttöliittymän suunnittelussa ja kehittämisessä otettiin viitteitä C-Service Kalmar -sovelluksesta. Kalmar-tuoteperhettä varten tehty sovellus toteutettiin aikoinaan Silverlightia käyttäen, joten sitä pystyttiin hyödyntämään uuden C-Servicen suunnittelussa. Aluperäinen ajatus oli yhdistää vahvasti Cargotec-värimaailma sovelluksen käyttöliittymän tylissä, mutta lopuksi käytettiin puna-, valko- ja mustayhdistelmää hillitysti. Eniten väriyhdistelmää on nähtävissä sovelluksen hakutoimintovalikossa, joka on muutenkin yksi suurimmista muutoksista uudessa sovelluksessa. Tummasävyinen käyttöliittymä valittiin, jotta se antaisi käyttäjälle kokonaan erilaisen käyttökokemuksen. Sovelluksen tuli näyttää uudelta ja välttää kaikkea mikä näyttäisi vanhalta C-Serviceltä tai Windows Forms -sovellukselta. Suurin osa sovelluksen tyyleistä on tallennettuina sovelluksen tyylikirjastoihin tai resurssikirjastoihin, joista niitä on tulevaisuudessa helppo muokata, jos halutaan muuttaa sovelluksen ulkomuotoa.

C-Service Silverlightin ensimmäinen julkaisu oli joulukuussa 2012, jolloin sovellus otettiin testikäyttöön. Kehittäjätiimi testasi sovelluksen omassa kehitysympäristössä, mutta varsinaisen testauksen suorittivat Cargotecin tuotepäälliköt. Testauksessa ilmeni muutamia virheitä sekä muutama käyttökokemusta parantava tekijä. Tällaisia olivat mm. puunäkymässä olevien mallien, varaosien sekä brändien näkyminen erilaisella tyyllillä. Käyttäjät halusivat, että mallit jotka sisäl-

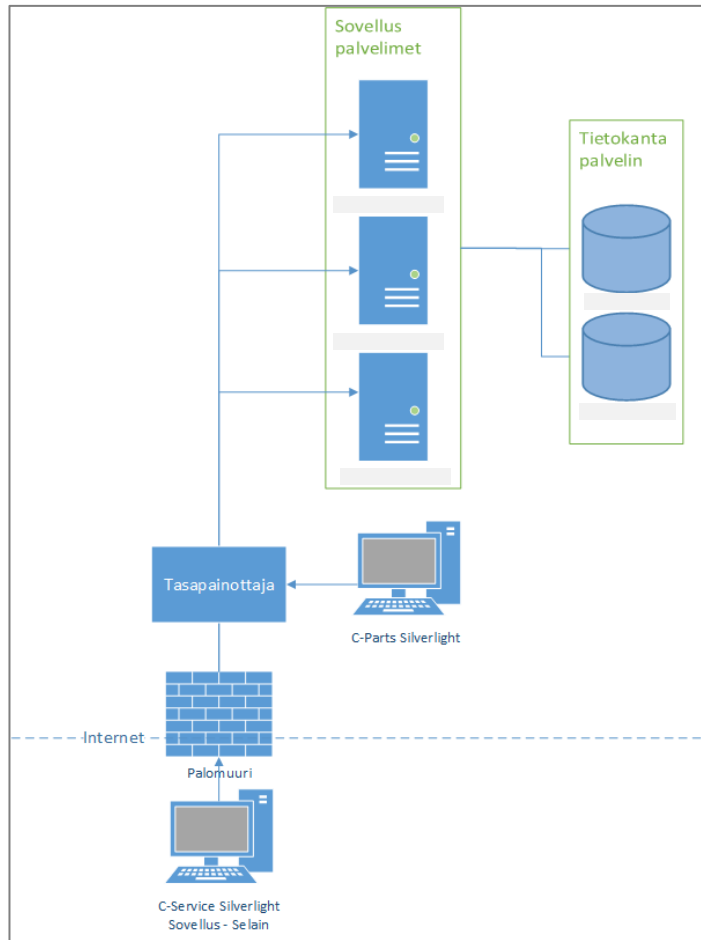
täisivät kuvia, tulisi näkyä selvemmin jo puunäkymästä. Jokaisen iteraation jälkeen uusi versio julkaistiin testattavaksi ja sitä toistettiin kunnes sovellus oli julkaisukelpoinen.

The screenshot displays a CAD software interface. On the left, a tree view shows the assembly structure under '33SK CLX'. The main area shows a 3D model of a crane boom assembly with various components labeled with letters and numbers. Below the model is a parts list table.

#	Part	Description	Quantity	Buy	Link
1	5330441	FIRST BOOM	1		5330441 1st boom
2	8343012	HYDRAULIC CYLINDER	1		8343012 2nd boom cylinder (Up to 26/07/2007)(Up to 3350505)
3	5330629	SHAFT	1		
4	5330645	SHAFT	1		
5	5328926	SHAFT	1		
6	5331218	HYDRAULIC PIPE	6		
7	5331226	PIPE	3		
8	5331161	HYDRAULIC PIPE	1		

Kuva 6.3: C-Service Silverlight

Kuvassa 6.3 oleva uusi C-Service korvaa vanhan verkossa toimivan Web Clientin ja C-Partissa olevan C-Servicen. Uusi versio poistaa monta isoa ongelmaa, joita vanhoissa sovelluksissa ilmeni. Sovellusten teknologinen päivitys on arvioitu riittävän noin viideksi vuodeksi eteenpäin, jonka jälkeen tulee arvioida ajankohtaisen teknologian sekä asiakasvaatimusten perusteella jälleen uutta päivitystä tai kokonaan uutta sovellusta. Suunnitelmissa on myös päivittää C-Service-työpöytäsovellus vuoden 2013 loppupuolella tai 2014 alussa. Projektin aikataulu riippuu siitä, näkeekö Cargotec työpöytäsovelluksen uusimisen tärkeänä tai riittääkö heillä budjettia siihen. Toistaiseksi jatkokehitystoimet ovat vielä karkeita suunnitelmia.



**Kuva 6.4: Sovellus arkkitehtuuri**

Kuvassa 6.4 on esitetty C-Service-arkkitehtuurirakenne. C-Service-sovellus on yhteydessä yhteen kolmesta mahdollisesta Windows Server -palvelimesta. Palvelimet toimivat sovelluspalvelimina, jotka tarjoavat ajankohtaisen version C-Service-sovelluksesta. C-Service-tuotantotietokanta sijaitsee erillisellä Windows SQL Server -palvelimella. C-Servicen tietorakenne on jaettu kahteen erilliseen tietokantaan. Toisessa tietokannassa sijaitsee tietoa mm. brändeistä, malleista, varaosista ja kuvista. Toisen tietokannan kautta haetaan C-Serviceen tuotetietoihin liitetyt sarjanumerot.

## 7 LOPPUPÄÄTELMÄ

Silverlightin oppiminen oli kohtuullisen helppoa ja teknologiaan oppiminen on nopeaa. Oppimista helpotti laaja kirjallisuus sekä tukea oppimiseen ja ongelmiin sai Silverlight-yhteisöltä. Silverlightin jälkeen on helppo siirtyä Microsoftin muihin teknologioihin, kuten Windows Phone tai Windows 8 -kehitykseen. (Bugnion 2010)

C-Service projektissa Silverlight pystyi tarjoamaan käyttäjilleen samanlaisen kokemuksen riippumatta siitä millä selaimella sitä käytetään. Aiempaa versiota oli mahdollista käyttää ainoastaan Internet Explorer -selaimella. Sovelluksen mahdollistaminen useammalle selaimelle antoi enemmän vapautta käyttäjille. Kehitystyö Silverlight-teknologialla oli vaivatonta useasta syystä. MVVM-arkkitehtuuria noudattamalla pystyimme työskentelemään tiiminä, nopeuttamaan kehitystyötämme sekä selviämään nopeammin ongelmatilanteista.

Ketterien menetelmien hyödyntäminen C-Service-projektissa auttoi koko Key Account -tiimiä olemaan tietoisia siitä, miten projekti etenee sekä mitä ongelmia on ilmennyt. Ketterien menetelmien Scrum-menetelmää ei käytetty kehitystyössä kokonaisuutena. Suurin syy tälle oli se, että projekti ei ollut tarpeeksi suuri eikä siihen osallistunut tarpeeksi monta tekijää. Parhaiden käytäntöjen käyttäminen kuitenkin piti työtahdin tehokkaana sekä kehittäjät ja managerit tietoisina tarvittavasta työstä. Näin pystyttiin myös jakamaan työpanos useiden eri projektien välillä, joita toteutettiin samaan aikaan C-Service Silverlightin kanssa.

C-Service Silverlight -projekti ei valmistunut suunnitellussa aikataulussa, liian kevyiden suunnitelmien vuoksi. Ongelmaksi muodostui epäselvyys siitä, mikä kuuluu projektiin ja mikä ei. Asiakas oletti tiettyjen ominaisuuksien kuuluvan, kun taas kehitystiimi oli ajatellut sen olevan muutostyö. Kokonaisuutena projekti onnistui hyvin. Seuraava askel C-Servicen kehitystyössä on toteuttaa sovelluksesta työpöytäversio.

# LÄHTEET

Agile Alliance (2001): Manifesto for Agile Software Development .

Viitattu 1.1.2013. <http://agilemanifesto.org/> .

Bugnion, Laurent (2010): Silverlight 4 Unleashed. Sams Publishing US.

Cargotec: Cargotec (2013). Viitattu 1.1.2013. <http://www.cargotec.com/fi-fi/cargotec/Sivut/default.aspx>

eCraft Oy Ab 2012a. eCraft yrityspresentaatio 2012 FI. Viitattu 15.12.2012.

(Yrityksen sisäinen dokumentti).

eCraft Oy Ab 2012b. C-Service Tech Upgrade. Viitattu 15.8.2012. (Yrityksen sisäinen dokumentti).

eCraft Oy Ab 2012c. C-Service User Manual. Viitattu 15.8.2012. (Yrityksen sisäinen dokumentti)

eCraft Oy Ab 2012d. Viitattu 12.3.2013. <http://eCraft.com/mantramme/>.

eCraft Oy Ab 2012e. Viitattu 12.3.2013. <http://eCraft.com/yritysfaktat/> .

Microsoft Developer Network (2013): Data Binding. Viitattu 15.3.2013. [http://msdn.microsoft.com/en-us/library/cc278072\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc278072(v=vs.95).aspx).

Moroney, Laurence (2009): Introduction Microsoft Silverlight 3. Microsoft Press.

Siddigi, Muhammed ja Vice, Ryan (2012): *MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF*. Packt Publishing Ltd. Birmingham.

Six Sigma (2013): Lean ja johtaminen. Viitattu 1.1.2013. <http://www.sixsigma.fi/fi/lean/lean-ja-johtaminen/>.

Shore, James & Warden, Shane (2008): The Art of Agile Development. O'Reilly Media.

Sutherland, Jeff & Schwaber, Ken. (2011): The Scrum Guide - Scrumin määritelmä ja pelisäännöt. Scrum.org.