

Lappeenrannan teknillinen yliopisto  
School of Business and Management  
Tietotekniikan koulutusohjelma

Diplomityö

**Mikko Turunen**

## **OHJELMISTOASENNUSTEN AUTOMATISOINTI TIETOKONEEN KÄYTTÖÖNOTTOVAIHEESSA**

Työn tarkastajat: Tekniikan tohtori Uolevi Nikula  
Diplomi-insinööri Pasi Hiltunen

Työn ohjaajat: Tekniikan tohtori Uolevi Nikula  
Diplomi-insinööri Pasi Hiltunen

# TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto  
School of Business and Management  
Tietotekniikan koulutusohjelma

Mikko Turunen

## **Ohjelmistoasennusten automatisointi tietokoneen käyttöönottovaiheessa**

Diplomityö

2016

67 sivua, 8 kuvaa, 6 taulukkoa

Työn tarkastajat: Tekniikan tohtori Uolevi Nikula  
Diplomi-insinööri Pasi Hiltunen

Hakusanat: ohjelmistoasennus, ohjelmistojen käyttöönotto, ohjelmistointegraatio, tietojärjestelmäintegraatio, COTS-integraatio

Keywords: software installation, software deployment, software integration, COTS integration, information system integration

Kaakkois-Suomen Tieto Oy:n suorittamien uusien työasemien käyttöjärjestelmäasennusten yhteydessä joudutaan usein asentamaan kolmannen osapuolen ohjelmistoja manuaalisesti ja vuosittain ohjelmistoasennuksiin on arvioitu käytettävän satoja henkilötyötunteja. Tämän diplomityön tavoitteena oli kehittää konstruktivisen tutkimuksen keinoin ratkaisu, jonka avulla uuden työaseman käyttöönottoon vaadittava henkilötyömäärä vähenisi.

Työssä kuvataan usean Kaakkois-Suomen Tieto Oy:n käyttämän kaupallisen ohjelmiston välisen integraation kehitysprojekti. Kehitysprojektin tuloksena syntyneen prototyypijärjestelmän avulla onnistuttiin automatisoimaan käyttöjärjestelmäasennusten yhteydessä suoritettavat ohjelmistoasennukset työssä kuvattavien rajoitteiden puitteissa.

Työssä kehitetyn integraatiojärjestelmän vaikutuksia työaseman käyttöönottoon käytettäviin henkilötyötunteihin tarkastellaan ja analysoidaan työn tulosten yhteydessä. Työn johtopäätöksissä tehdään havaintoja suunnittelun vaikutuksista ohjelmistointegraation kehitysprosessin suhteen ja lisäksi luvussa esitetään tarkistuslista, jonka tarkoituksena on helpottaa ohjelmistointegraatioiden suunnittelemista.

## **ABSTRACT**

Lappeenranta University of Technology  
School of Business and Management  
Degree Program in Computer Science

Mikko Turunen

### **Automating software installations during workstation deployment**

Master's Thesis

67 pages, 8 figures, 6 tables

Examiners : D.Sc. Uolevi Nikula  
M.Sc. Pasi Hiltunen

Keywords: software installation, software deployment, software integration, COTS integration, information system integration

After deploying operating systems to new computers at Kaakkois-Suomen Tieto Oy instances of third party software often need to be installed manually by system administrators. It has been estimated that the manual software installations require hundreds of working hours on a yearly basis. The goal of thesis was to create a solution which would reduce the amount of manual labor required for deploying a new workstation with the methods of constructive research.

This thesis describes the integration project of multiple pieces of Commercial-Off-The-Shelf software used at Kaakkois-Suomen Tieto Oy. The integration project resulted in a prototype system with which the software installations were automatized within restrictions described in the thesis.

The effects of the integrated software system on the new workstation deployments are examined and analyzed in the results chapter of the thesis. Remarks about the effects of planning on a software integration project are made in the conclusions chapter and also a checklist for aiding the planning of software integration projects is presented in the same chapter.

## **ALKUSANAT**

Tahdon kiittää Kaakkois-Suomen Tieto Oy:tä diplomityöpaikasta, KS-Tiedon Tommi Voutilaista ja Mikko Parkkista työpaikan järjestämisestä sekä käytännön työn ohjaamisesta ja Pasi Hiltusta, joka toimi kirjallisen työn ohjaajana sekä tarkastajana.

Suuri kiitos myös LTY:n Uolevi Nikulalle tämän diplomityön ja aiemmin tekemäni kandidaatintyön ohjauksesta ja tarkastuksesta.

Viimeiseksi tahdon kiittää kaikkia läheisiäni tukemisestani näiden lukuisien opiskeluvuosien aikana. Ilman teidän tukeanne aurinkokin olisi varmaan ehtinyt sammua ennen valmistumistani.

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO</b> .....	<b>4</b>
1.1	TAUSTA .....	4
1.2	TAVOITTEET JA RAJAUKSET .....	4
1.3	TYÖN RAKENNE .....	5
<b>2</b>	<b>KIRJALLISUUSKATSAUS</b> .....	<b>6</b>
2.1	OHJELMISTOJEN KÄYTTÖÖNOTTO .....	7
2.2	OHJELMISTOINTEGRAATIO .....	8
2.3	TEKNISET DOKUMENTAATIOT .....	9
<b>3</b>	<b>TUTKIMUSMENETELMÄ</b> .....	<b>11</b>
3.1	KONSTRUKTIIVINEN TUTKIMUSOTE .....	11
3.2	TUTKIMUSONGELMA.....	11
3.3	ONGELMAN LÄHESTYMINEN .....	12
3.4	KONSTRUKTIO .....	12
3.5	KONSTRUKTION VALIDOINTI.....	13
<b>4</b>	<b>TEHTY TAPAUSTUTKIMUS</b> .....	<b>14</b>
4.1	OHJELMISTOINTEGRAATIO.....	14
4.2	TILAAJAYRITYKSEN TOIMINTAYMPÄRISTÖ .....	16
4.3	KONSTRUKTION KESKEISIMMÄT OHJELMISTOT .....	17
4.3.1	<i>ServiceNow</i> .....	17
4.3.2	<i>LANDESK Management Suite</i> .....	19
4.3.3	<i>Microsoft SQL Server</i> .....	20
4.3.4	<i>Microsoft Deployment Toolkit</i> .....	20
4.3.5	<i>Active Directory</i> .....	23
4.3.6	<i>PowerShell</i> .....	23
4.4	KONSTRUKTION TOTEUTUS .....	23
4.4.1	<i>Konstruktion kehityssuunnitelma</i> .....	24
4.4.2	<i>LANDESK Management Suite – ServiceNow -integraatio</i> .....	26
4.4.3	<i>ServiceNow – Microsoft Deployment Tool -integraatio</i> .....	30
4.4.4	<i>Muutos suunniteltuun järjestelmään</i> .....	35
4.4.5	<i>Microsoft Deployment Tool – Active Directory -integraatio</i> .....	38
4.4.6	<i>Active Directory – LANDESK Management Suite -integraatio</i> .....	40
4.4.7	<i>Active Directory – ServiceNow -integraatio</i> .....	41

4.4.8	<i>Toteutunut integraatiojärjestelmä .....</i>	<i>41</i>
<b>5</b>	<b>KONSTRUKTION ARVIOINTI .....</b>	<b>44</b>
<b>6</b>	<b>POHDINTA JA JOHTOPÄÄTÖKSET .....</b>	<b>49</b>
6.1	HAVAINTOJA COTS-OHJELMISTOINTEGRAATIOIDEN SUUNNITTELUSTA .....	50
6.1.1	<i>Integroitavien ohjelmistojen määrä .....</i>	<i>52</i>
6.1.2	<i>Integraatioiden suunnat .....</i>	<i>52</i>
6.1.3	<i>Ohjelmistojen rajapinnat ja rajapintojen käyttäminen .....</i>	<i>52</i>
6.1.4	<i>Tietojen käsittely .....</i>	<i>54</i>
6.1.5	<i>Tietoturva .....</i>	<i>54</i>
6.1.6	<i>Ylläpito .....</i>	<i>54</i>
6.1.7	<i>Ohjelmistojen korvaaminen .....</i>	<i>55</i>
6.1.8	<i>Integraation kustannusten ja kannattavuuden arvioiminen .....</i>	<i>56</i>
6.2	JATKOTUTKIMUS .....	57
<b>7</b>	<b>YHTEENVETO .....</b>	<b>58</b>
	<b>LÄHTEET .....</b>	<b>59</b>

## **SYMBOLI- JA LYHENNELUETTELO**

ACM DL	Association for Computing Machinery Digital Library
ADSI	Active Directory Service Interfaces
CMDB	Content Management Database
COTS	Commercial-Off-The-Shelf
ERP	Enterprise Resource Planning
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineering
JS	JavaScript
LDAP	Lightweight Directory Access Protocol
LTI	Lite Touch Installation
MDT	Microsoft Deployment Toolkit
MID Server	Management, Instrumentation, and Discovery Server
PS	PowerShell
SCCM	System Center Configuration Manager
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
REST	Representational State Transfer
T-SQL	Transact-SQL
UDI	User Driven Installation
WSDL	Web Service Description Protocol
XML	Extensible Markup Language
ZTI	Zero Touch Installation

# 1 JOHDANTO

ICT-alan (Information and Communications Technology) ominaispiirteisiin kuuluu erilaisten toistuvien toimintojen automatisoiminen tehokkuuden parantamiseksi. Tämän diplomityön tavoitteena oli kehittää ratkaisu, jonka avulla ohjelmistojen asennukset voitaisiin suorittaa automaattisesti käyttöjärjestelmän asennuksen yhteydessä työn tilanteen yrityksen toimintaympäristössä.

## 1.1 Tausta

Kaakkois-Suomen Tieto Oy, johon jatkossa viitataan KS-Tietona tai (diplomityön) tilaajayrityksenä, asentaa vuosittain yli 500:n työasemaan käyttöjärjestelmän ja sen yhteydessä myös useita kolmannen osapuolen ohjelmistoja. Osa asennettavista ohjelmistoista on paketoitu käyttöjärjestelmän asennuskuvakkeeseen, mutta käytännössä kaikkien käyttöjärjestelmäasennusten yhteydessä asennetaan myös muita ohjelmistoja. Nämä muut ohjelmistot on asennettu KS-Tiedon työntekijöiden toimesta manuaalisesti seuraamalla työaseman tiedoista löytyvää ohjelmistolistaa ja KS-Tiedon asennuspalveluista vastaavan työntekijän arvion mukaan tämä on vaatinut vuosittain satoja henkilötyötunteja.

## 1.2 Tavoitteet ja rajaukset

Työn tavoitteena on kehittää KS-Tiedolle ohjelmistojärjestelmä jonka avulla käyttöjärjestelmän asennuksen yhteydessä asennettavat ohjelmistot pystytään asentamaan täysin automaattisesti. Kehitettävän ohjelmistojärjestelmän tulisi parantaa käyttöjärjestelmän asennusprosessin tehokkuutta vähentämällä siihen käytettäviä henkilötyötunteja. Käytännössä ohjelmistojärjestelmän kehittäminen tarkoittaa erinäisten KS-Tiedolla käytössä olevien ohjelmistojen ja järjestelmien keskinäistä integraatiota automaattisten ohjelmistoasennusten suorittamiseksi. Rajoitteeksi järjestelmän toteutukselle oli asetettu, että siinä tulee käyttää vain KS-Tiedolla käytössä olevia ohjelmistoja sekä muita KS-Tiedolla olemassa olevia resursseja, eikä laitteisto- tai ohjelmistohankintoja oltu budjetoitu työtä varten. Kehitettävän järjestelmän vaikutusta käyttöjärjestelmäasennuksiin käytettäviin henkilötyötunteihin tulee myös testata



järjestelmän hyödyllisyyden arvioimisen vuoksi. Työssä kehitettävän järjestelmän tavoiteltava toiminnallisuus rajataan uusien työasemien yhteydessä suoritettaviin käyttöjärjestelmäasennuksiin, jotta kehitystyö olisi mahdollisimman suoraviivaista ja järjestelmän vaikutusten testaaminen on selkeästi määriteltävissä.

### **1.3 Työn rakenne**

Työn toisessa luvussa käydään läpi työhön liittyviä kirjallisuuslähteitä. Kolmannessa luvussa esitellään työssä käytetty tutkimusmenetelmä. Neljännessä sisältää työssä tehdyn tapaustutkimuksen, jossa käydään läpi ohjelmistointegraatioita yleisellä tasolla, esitellään työn tilanteen yrityksen toimintaympäristöä sekä kuvataan työssä tuotetun konstruktion kehitysprojekti. Viidennessä luvussa esitetään konstruktion mittaustulokset sekä analysoidaan kyseisiä tuloksia. Kuudes luku sisältää pohdintaa työstä sekä havaintoja ohjelmistointegraatioiden suunnittelemisesta. Seitsemäs luku puolestaan sisältää työn yhteenvedon.

## 2 KIRJALLISUUSKATSAUS

Tämä diplomityö keskittyy ohjelmistoasennusten automatisointiin, joka on osa-alue ohjelmistojen käyttöönotto -nimisestä tutkimusalueesta. Varsinaista ohjelmistoasennusten automatisointia voidaan pitää käytännön ongelmana, johon etsitään ratkaisumalleja ohjelmistojen käyttöönottoon liittyvien tutkimusten kautta. Toisaalta ohjelmistoasennusten automatisointi tämän työn puitteissa vaati useiden COTS-ohjelmistojen (Commercial-Off-The-Shelf) ja -tietojärjestelmien integroimista keskenään ja ohjelmistointegraatioon keskittyvät tutkimukset ovat siten olennaisia työn kannalta. Varsinaisten COTS-ohjelmistointegraatioiden toteutukset puolestaan vaativat ohjelmistotuottajien teknisiä tietolähteitä.

Ohjelmistojen käyttöönottoon sekä ohjelmistointegraatioon liittyviä tutkimuksia etsittiin Google Scholar -hakukoneen avulla, joka osoittautui tehokkaammaksi tavaksi etsiä tutkimuksia kuin julkaisutietokantojen omien hakutyökalujen käyttäminen [1]. Google Scholar hakee tutkimuksia, artikkeleita ja kirjoja ainakin koulutuslaitosten omista julkaisualustoista sekä tieteellisiin julkaisuihin keskittyvistä tietokannoista kuten ACM DL:stä (Association for Computing Machinery Digital Library) ja IEEE Xploresta (Institute of Electrical and Electronics Engineering) [2][3]. Konstruktion käytännön toteutukseen liittyviä teknisiä lähteitä etsittiin normaalin Google-hakukoneen avulla sekä ohjelmisto- ja järjestelmätuottajien omien verkkosivujen kautta [4].

Ohjelmistojen käyttöönottoon ja asentamiseen liittyviä tutkimuksia etsittiin muun muassa seuraavilla hakusanayhdistelmillä: application deployment, software deployment, package management ja software installation. Toimivin hakusanayhdistelmä oli software deployment, joka tuotti laajimman kattauksen hyödyllisiksi koettavia lähteitä. Ohjelmistointegraatiota käsitteleviä teoksia puolestaan haettiin hakusanoilla: software integration, system integration, systems integration, information system integration, information systems integration ja COTS integration. Ohjelmistointegraatioon liittyviä tutkimuksia haettaessa oli ongelmallista että software integration, system integration sekä systems integration ovat englannin kielessä hieman epämääräisiä termejä, jotka voivat suoraan liittyä ohjelmistojen väliseen integraatioon. Suuri osa edellämainittujen hakujen

tuottamista artikkeleista käsittelevät usein tähän työhön liittymättömiä asioita, kuten esimerkiksi software integrationin tapauksessa teokset usein käsittelevät ohjelmistointegraatiota osana ohjelmistojen kehitysprosessia. Ohjelmistointegraatiota käsitteleviä artikkeleita läpikäytäessä löytyi tarkemmin työn sisältöön liittyvä ohjelmistointegraation alue, joka on nimeltään COTS-ohjelmistointegraatio. Parhaaksi hakusanayhdistelmäksi osoittautuikin COTS integration, jonka tuottamat hakutulokset olivat pääosin relevantteja tämän työn aiheen kannalta.

## 2.1 Ohjelmistojen käyttöönotto

Dearlen *Software Deployment, Past, Present and Future* -artikkeli vuodelta 2007 toimii hyvänä lähtökohtana ohjelmistojen käyttöönoton tutkimuksessa [5]. Dearlen artikkelissa selitetään mitä ohjelmistojen käyttöönotto tarkoittaa ja mitä haasteita ohjelmistojen käyttöönottoon liittyy. Lisäksi Dearlen artikkeli sisältää myös useita tapaustutkimuksia aiheesta, joiden avulla on mahdollista muodostaa käsitys tutkimusaiheen käytännön sovellutuksista.

Ohjelmistojen käyttöönottoa on alettu laajemmin tutkimaan 1990-luvun puolesta välistä lähtien. Eräs varhaisimmista aiheita käsittelevistä artikkeleista on Tallmanin kirjoittama *Project Gabriel: Automated Software Deployment in a Large Commercial Network*, jossa käydään läpi suuren ranskalaispankin ohjelmistojen käyttöönoton -automatisointiprojektin läpiviemistä [6]. Artikkelissa käsitellään käytännön ongelmia, jotka toistuvat myöhemmin suoritettujen tutkimusten tutkimuskysymyksissä ja artikkelista ilmeneekin ohjelmistojen käyttöönoton automatisoinnin monimutkaisuus käytännön ja teorian tasolla.

Carzaniga et. al. tutkimusartikkelissa *A Characterization Framework for Software Deployment Technologies* vuodelta 1998 määrittellään kehysympäristö ohjelmistojen käyttöönottoa varten, josta selviää ohjelmistojen käyttöönotossa suoritettavat päätoiminnot, mukaanlukien ohjelmistojen asentaminen [7]. Suuressa osassa myöhemmin julkaistuista tutkimuksista käytetään kyseistä Carzaniga et. al. luomaa kehysympäristön mallia tai variaatiota kyseisestä mallista selittämään ohjelmistojen käyttöönottoon liittyviä vaiheita, jonka vuoksi *A Characterization Framework for Software Deployment Technologies* -

artikkeliä voidaan pitää keskeisenä teoksena ohjelmistojen käyttöönottoa tutkittaessa.

Yleinen aihe ohjelmistojen käyttöönottoa käsittelevissä tutkimuksissa on erilaisten kehysympäristöjen kehittäminen, joiden avulla ohjelmistojen käyttöönottoprosessi voidaan toteuttaa mahdollisimman kokonaisvaltaisesti. Esimerkki tämäntyyppisestä tutkimuksesta on Dolstran *The Purely Functional Software Deployment Model* -väitöskirja, jossa tutkitaan ohjelmistojen käyttöönottoa hallinnoivan järjestelmän toimintaa ja työssä myös tuotetaan tällainen järjestelmä [8]. Dolstran väitöskirjan sisällöstä löytyy myös vapaasti saatavilla oleva verkkoartikkeli, jossa käsitellään väitöskirjan sisältöä ytimekkäästi [9].

2010-luvulla ohjelmistojen käyttöönottoon liittyvien tutkimusten painopiste vaikuttaa siirtyneen kohti pilvipalveluissa ja virtuaaliympäristöissä tapahtuvia ohjelmistojen käyttöönottoja. Eräs tällainen tutkimus on Juve & Deelmanin *Automating Application Deployment in Infrastructure Clouds*, joka keskittyy pilvi- ja virtuaalialustojen resurssienhallinnan parantamiseen ohjelmistojen käyttöönoton yhteydessä [10].

## **2.2 Ohjelmistointegraatio**

Hyvä aloituspointe ohjelmistointegraatioon perehdyttäessä on Hasselbringin kirjoittama *Information System Integration* -lehtiartikkeli [11]. Kyseisessä artikkelissa käydään läpi ohjelmistointegraation perusteita selkeiden esimerkkien avulla. Artikkelissa myöskin tarkastellaan mihin muihin tietoteknisen tutkimuksen osa-alueisiin ohjelmistointegrointi on sidoksissa, joka voi edesauttaa kokonaiskuvan muodostamista tutkimusalueesta.

Eräänlaisena jatkumona Information System Integrationille Crnkovic & Landin *Existing Approaches to Software Integration - and a Challenge for the Future* myöskin käsittelee ohjelmistointegraatiota melko yleisellä tasolla, mutta tieteellisen tutkimuksen muodossa [12]. Crnkovic & Landin artikkelissa jaotellaan ohjelmistointegraatiot kolmeen eri päätyyppiin ja näiden päätyyppien tunnistaminen helpotti ohjelmistointegraatioon liittyvän kirjallisuuden etsimistä huomattavasti. Artikkelin myöskin sisältää suuren määrän lähdeviittauksia

Boehm & Abtsin *COTS Integration: Plug and Pray?* -lehtiartikkelista selviää COTS-integraatioiden pääpiirteet helposti omaksuttavassa muodossa [13]. Artikkelissa käydään läpi COTS-integraatioiden hyötyjä ja haittoja verrattuna räätälöityihin järjestelmiin ja artikkelin avulla pystyy muodostamaan hyvän käsityksen perinteisten ohjelmistointegraatioiden ja COTS-integraatioiden yhtäläisyyksistä sekä eroavaisuuksista.

COTS-integraatiota on alettu laajemmin tutkimaan ohjelmistojen käyttöönoton tavoin 1990-luvun puolivälissä. Eräs viitatuimmista aihealueen artikkeleista on Vidger et. al. tutkimus *COTS Software Integration: State of the Art* vuodelta 1996 [14]. Vidger et. al. tutkimuksessa käsitellään hyvin pitkälti samoja asioita kuin *COTS Integration: Plug and Pray?* lehtiartikkelissa, mutta laajemmin ja tieteellisessä muodossa.

Suuri osa COTS-integraatiota käsittelevistä artikkeleista keskittyy COTS-integraatioiden kustannusten arvioimiseen tai määrittämiseen. Abts et. al. tutkimusartikkeli *COCOTS: A COTS Software Integration Cost Model - Model Overview and Preliminary Data Collection Findings* on malliesimerkki tällaisesta tutkimuksesta, jossa esitetään kehysmalli COTS-integraatioiden hinnan arvioimista varten [15].

Viitatuimmat COTS-integraatioon liittyvät tutkimukset on julkaistu 2000-luvun taitteessa, mutta aihetta tutkitaan edelleen aktiivisesti. Valtaosa myöhemmistä tutkimuksista kuitenkin käsittelee COTS-integraatioiden kustannuksia, mutta tutkimuksen kohteena olevia integraatioita on vain tarkennettu johonkin sovellusalueeseen, kuten esimerkiksi ERP-järjestelmiin (Enterprise Resource Planning) Rosa et. al. *COTS integration and estimation for ERP* -tutkimuksessa [16].

### **2.3 Tekniset dokumentaatiot**

Ohjelmistotuottajien tekniset dokumentaatiot työssä hyödynnetyistä ohjelmistoista olivat elintärkeitä työn konstruktion tuottamisen kannalta. Ylivoimaisesti käytetyin tällaisista lähteistä oli Microsoftin TechNet-kirjasto, josta etsittiin tietoa SQL Server, Active Directory, Microsoft Deployment Tool sekä Microsoft Powershell -ohjelmistojen hyödyntämisen yhteydessä [17]. Konstruktion kannalta olennainen tietolähde oli myöskin

ServiceNow'n wikisivusto, jota ilman konstruktion toteutus olisi ollut lähes mahdotonta [18]. Ohjelmistotuottajien dokumentaatioiden lisäksi erinäiset keskustelufoorumit ja blogit olivat hyödyllisiä joidenkin yksittäisten ongelmien käytännön ratkaisujen kannalta, mutta tällaisiin lähteisiin viittaamista pyrittiin välttämään työssä niiden epämuodollisuuden vuoksi.

### **3 TUTKIMUSMENETELMÄ**

Diplomityötä suunniteltaessa oli tiedossa, että työssä tulitisiin kehittämään ratkaisu, jonka avulla KS-Tiedon työasematilausten yhteydessä tilattavat työasemalle asennettavat ohjelmistot asentuisivat automaattisesti työaseman käyttöjärjestelmän asentamisen yhteydessä. Tämän esitiedon pohjalta oli perusteltua, että tutkimus suoritettaisiin konstruktiiivisen tutkimusotteen mukaisesti.

#### **3.1 Konstruktiiivinen tutkimusote**

Lukan mukaan tapaustutkimuksia voidaan suorittaa konstruktiiivisen tutkimusotteen menetelmin. Konstruktiiivisen tutkimusotteen olennaisimpiin ominaisuuksiin kuuluu, että tutkimuksessa pyritään ratkaisemaan jokin konkreettinen ongelma uuden konstruktion, eli tutkimuksessa tuotettavan ratkaisun avulla, ja että tutkimuksessa tuotetun konstruktion toimivuutta testataan jollakin tavalla.. Konstruktio itsessään on abstrakti käsite joka voi ilmentyä rajoittamattomilla tavoilla kuten esimerkiksi suunnitelmina, malleina ja fyysisinä tuotteina. Konstruktiiivinen tutkimusprosessi voidaan jakaa Lukan mukaan seuraaviin osiin [19]:

1. Etsi käytännönläheinen ongelma, joka voi myös olla teorian kannalta hyödyllinen
2. Arvioi tutkimuskohteen potentiaali pitkän aikavälin tutkimustyöhön
3. Kerää tietoa ongelmasta ja sen aihealueesta
4. Kehitä ratkaisu ongelmaan ja kehitä ongelman ratkaiseva konstruktio
5. Kehitä konstruktion implementaatio ja testaa sen toimivuus
6. Arvioi kuinka laajalti ratkaisua voi hyödyntää
7. Tunnista ja analysoi tutkimuksen teoreettinen hyödyllisyys

#### **3.2 Tutkimusongelma**

KS-Tiedolta tilataan vuosittain yli 500 uutta työasemaa, joihin usein asennetaan myös käyttäjän tilaamia ohjelmistoja ennen toimitusta. Tilatut ohjelmistot on täytynyt asentaa työasemille manuaalisesti tilauslistaa seuraamalla, joka on vaatinut vuositasolla suuren

määrän henkilötyötunteja. Lisäksi olemassa oleville työasemille asennettujen ohjelmistojen seuraaminen on ollut vaivalloista, koska vuosien varrella ohjelmistoasennuksia on suoritettu kirjaamatta niitä mihinkään inventaariojärjestelmään. Tutkimusongelma on täten seuraava: *Miten työasematilausten käsittelyyn käytettävää työtuntimäärää saadaan vähennettyä?*

### **3.3 Ongelman lähestyminen**

Tilaaajayrityksessä oli alustavasti suunniteltu ohjelmistoasennusten automatisointia työasematilausten käsittelyprosessissa, joka vähentäisi työaseman toimitukseen käytettäviä työtunteja. Ohjelmistoasennusten automatisointi tilaaajayrityksen toimintaympäristössä vaati usean tietojärjestelmän välistä integraatiota ja tällaisen integraatioprojektin toteuttamiseen ei riittänyt resursseja tilaaajayrityksen sisältä ennen tämän diplomityön aloittamista. Tutkimusongelma siis pyrittiin ratkaisemaan seuraavan työhypoteesin mukaisesti: *Automatisoimalla uusien työasemien ohjelmistoasennukset, työasematilausten käsittelyyn käytettävien työtuntien määrä vähenee.*

### **3.4 Konstruktio**

Työssä integroidaan tilaaajayrityksessä käytössä olevia tietojärjestelmiä ja ohjelmistoja suuremmaksi järjestelmäkokonaisuudeksi, joka on työssä implentoitava konstruktio. Toteutettava järjestelmäkokonaisuus tulee mahdollistamaan uusille työasemille tilattujen ohjelmistojen automaattisen asentamisen käyttöjärjestelmän asennuksen yhteydessä. Tuotettavan konstruktion avulla on myös mahdollista kerätä tietoja käytössä oleville työasemille asennetuista ohjelmistoista ja siirtää näitä tietoja tilaaajayrityksen inventointijärjestelmään, joka ei ole aiemmin ollut mahdollista. Konstruktio myös luo pohjan yksittäisten ohjelmistotilausten automatisoinnille sekä työaseman tilaajan vanhan työaseman ohjelmistojen automaattiselle asentamiselle tilattuun, uuteen työasemaan.



### **3.5 Konstruktion validointi**

Konstruktio validoitiin asentamalla työasemalle käyttöjärjestelmä sekä ennalta valittuja ohjelmistoja käyttöjärjestelmän asennuksen yhteydessä ja tutkimalla asennusprosessin eri vaiheisiin kuluva aikaa. Asennuksia suoritettiin toteutetun järjestelmän avulla sekä manuaalisesti ilman järjestelmää kuten asennusprosessit on aiemmin suoritettu tilaajayrityksessä. Yksittäistä asennusprosessia tarkkailtiin prosessin kokonaiskeston, prosessiin kuluneen manuaalisen työn kokonaiskeston sekä ohjelmistojen asentamiseen kuluneen manuaalisen työn keston osalta.

## 4 TEHTY TAPAUSTUTKIMUS

KS-Tiedon uusien työasemien tilaukset tehdään ServiceNow-pilvipalvelussa olevan asiakasportaalin kautta. Asiakasportaalin tilauslomakkeessa valitaan työaseman malli, lisälaitteet kuten näyttö, näppäimistö ja hiiri sekä työasemalle asennettavat ohjelmistot. Kun tilaus on lähetetty ServiceNow'hun tilaajayrityksen käsiteltäväksi, työasema ja lisälaitteet tilataan laitteistotoimittajalta. Kun tilatut laitteistot saapuvat tilaajayrityksen toimipisteelle, työasemaan asennetaan käyttöjärjestelmä ja asiakkaan tilaamat lisäohjelmistot asennetaan manuaalisesti. Näiden vaiheiden jälkeen työasema on valmis käyttöönotettavaksi ja työasema sekä tilatut lisälaitteet voidaan toimittaa asiakkaalle.

Työn tilanteen yrityksen työasemien käyttöönottoprosessissa on hyödynnetty asiakasryhmäkohtaisia käyttöjärjestelmän asennuslevy kuvia. Räättälöidyistä käyttöjärjestelmän asennuslevy kuvista huolimatta lähestulkoon jokaiseen uuteen työasemaan tilataan ohjelmistoja ja ne täytyy asentaa manuaalisesti tilattujen ohjelmistojen listaa seuraamalla. Työssä tuotetun konstruktion tarkoituksena oli automatisoida työaseman loppukäyttäjän tilaamien ohjelmistojen asennus ennen uuden tietokoneen käyttöönottoa niin, ettei käyttöönotossa tarvitsisi tehdä muuta manuaalista työtä kuin käynnistää käyttöjärjestelmän asennus työasemien asennuspisteellä. Konstruktioto toteutettiin integroimalla useita tilaajayrityksen käyttämiä ohjelmistoja ohjelmistojärjestelmäksi, jonka avulla tietokoneen käyttöönoton yhteydessä suoritettavat ohjelmistojen asennukset oli tarkoitus automatisoida.

### 4.1 Ohjelmistointegraatio

IEEE määrittelee ohjelmistotekniikan sanastossaan integraation vapaasti käännettynä seuraavanlaisesti: "Prosessi jonka avulla yhdistetään ohjelmistokomponentteja, laitteistokomponentteja, tai molempia järjestelmäkokonaisuudeksi" [20]. Keskusteltaessa ohjelmistojen välisestä integraatiosta, integraatiolla tarkoitetaan usein prosessia, jonka tuloksena prosessin kohteena olevat ohjelmistot pystyvät jakamaan tietoa toistensa kanssa. Puhuttaessa integraatiosta tämän määritelmän mukaisesti, syntyy integraation tuloksena useasta ohjelmistosta koostuva suurempi järjestelmäkokonaisuus ja siten määritelmä on

yhteneväinen IEEE:n määritelmän kanssa. Integraatiolla voidaan myös tarkoittaa kahden tai useamman ohjelmiston välistä tiedon jakamista itsessään, jolloin termistä tulee hieman abstraktimpi kuin prosessista puhuttaessa. Näiden kahden määritelmän väliset erot ovat kuitenkin melko vähäisiä ja termin varsinainen sisältö onkin usein riippuvainen kontekstista missä sitä käytetään. [21]

Ohjelmistointegraatioita on selkeintä käsitellä yksisuuntaisina kahden ohjelmiston välisinä operaatioina, jolloin integroitavat ohjelmistot voidaan nimetä aktiiviseksi ja passiiviseksi ohjelmistoksi. Aktiivinen ohjelmisto vaikuttaa passiiviseen ohjelmistoon jollakin tavalla, kuten pyytämällä siltä tietoja tai välittämällä sille tietoja omasta aloitteestaan. Ohjelmistointegraatio on käytännössä mahdollista toteuttaa kahdella tavalla: hyödyntämällä passiivisen ohjelmiston rajapintaa, jonka kautta tietoja välitetään tai manipuloimalla passiivisen ohjelmiston käyttämää tiedostoa joka sisältää tietoja. Kolmantena mahdollisuutena on passiivisen ohjelmiston manipulointi järjestelmäkutsutasolla, mutta tällaista ratkaisua voidaan kutsua hakkeroinniksi ja ratkaisun toteutettavuus sekä ylläpidettävyys on erittäin kyseenalaista. Kaikissa tapauksissa aktiivisessa ohjelmistossa eli rajapintaa hyödyntävässä, tiedostoja manipuloivassa tai passiivista ohjelmistoa manipuloivassa ohjelmistossa on myös oltava jonkinlainen mekanismi näiden operaatioiden suorittamiseen. [12]

Rajapinnat joiden avulla integraatioita toteutetaan ovat joko ohjelmistotuottajan määrittelemiä tai jonkin standardointiorganisaation määritelmiä. Esimerkiksi Microsoftin SQL Serverissä (Structured Query Language) käytettävä Transact-SQL on täysin Microsoftin hallinnoima rajapinta, ja toisena esimerkkinä Lightweight Directory Access Protocol (LDAP) on määritelty Internet Engineering Taskforcen toimesta Request for Comments: 4511 dokumentissa [22][23]. Tiedostomanipulaation avulla toteutettavissa integraatioissa manipuloitavan tiedoston formaatti voi rajapintojen tapaan olla joko ohjelmistotuottajan määrittelemä tai formaatti voi noudattaa yleistä standardia kuten Extensible Markup Languagea (XML) [24].

Passiivisen ohjelmiston käyttämän rajapinnan tai tiedostoformaatin yleisyydestä riippuen integraatio voi olla toteutettavissa aktiivisen ohjelmiston tarjoamien työkalujen avulla, joko

jonkinlaisen käyttöliittymän kautta tai asetustiedostojen muokkaamisen avulla. Mikäli integraatio ei onnistu valmiiden työkalujen avulla, integraatio vaatii tällöin, että aktiivisessa ohjelmistossa on tuki ohjelmakoodin, joka usein tarkoittaa jonkin skriptikielen tukemista, suorittamiseen ja integraatio on toteutettava ohjelmoimalla. [25]

Ohjelmistointegraation mahdollistava tiedonjakaminen on jaettavissa kahteen eri tyyppiin: tiedon noutamiseen passiivisesta ohjelmistosta ja tiedon välittämiseen passiiviselle ohjelmistolle. Tiedon noutaminen passiivisesta ohjelmistosta ei vaikuta sen tietoihin tai toimintaan ja operaatio ei ole intrusiivinen. Tiedon vieminen passiiviseen ohjelmistoon puolestaan on intrusiivinen operaatio, koska se vaikuttaa passiivisen ohjelmiston tietoihin tai toimintaan. Tiedon vieminen passiiviseen ohjelmistoon on edelleen jaettavissa alatyyppeihin, joita ovat tiedon vieminen, tiedon muokkaaminen ja tiedon poistaminen. Kaikkien edelläkuvattujen operaatioiden yhteydessä välitettävää tietoa on usein myös muotoiltava ohjelmistoille yhteensopivaan muotoon. Tiedon muotoilu on lähtökohtaisesti suoritettava aktiivisen ohjelmiston toimesta, koska passiivisen ohjelmiston tietojen käsittelyrutiineihin ei välttämättä pystytä suoranaisesti vaikuttamaan [26].

## **4.2 Tilaaajayrityksen toimintaympäristö**

KS-Tiedon logistiikkakeskus käsittelee vuosittain yli 4000 laitetta yli 10000:n loppukäyttäjän tarpeisiin, joista uusia työasemia on noin 500 kappaletta [27]. Työasemiin kuuluu sekä kannettavia että perinteisiä pöytäkoneita ja pääsääntöisesti näihin työasemiin asennetaan Windows 7 -käyttöjärjestelmä, tilaajan käyttöympäristön vaatimia yleisiä ohjelmistoja sekä erikseen tilattavia ohjelmistoja erityistarpeisiin. Osaan uusista työasemista asennetaan myös muita käyttöjärjestelmiä kuten Windows XP:tä erikoisinstrumenttien tai -ohjelmistojen toimivuuden takaamisen vuoksi.

KS-Tieto tarjoaa palveluita useiden erityyppisten asiakkaiden, kuten terveydenhuollon, julkishallinnon, opetuslaitosten sekä yritysten tarpeisiin. Asiakkaiden toimintaympäristöjen palveluvaatimukset voivat tällöin poiketa toisistaan merkittävästi, joka aiheuttaa haasteita asiakkaiden palvelemiseen. Esimerkiksi terveydenhuollon palveluiden saatavuuden tulee olla mahdollisimman suuri ja ongelmien ilmetessä ne täytyy ratkaista mahdollisimman

nopeasti, koska ongelmat järjestelmissä ja palveluissa voivat vaikuttaa ihmisten terveydentilaan negatiivisesti. Opetuslaitoksissa puolestaan on paljon erilaisia loppukäyttäjiä, joista osa saattaa jopa tahallisesti vioittaa ICT-järjestelmiä ja –laitteistoja, josta seuraa, että ympäristössä voi ilmetä hyvin monimuotoisia ongelmatilanteita.

KS-Tiedon asiakkailla on suuri määrä erilaisia ohjelmistoja käytössään monimuotoisten käyttöympäristöjen vuoksi. KS-Tieto tarjoaa tukea käytännössä kaikkiin asiakkaiden käyttämiin ohjelmistoihin, mutta virallisesti tuettuja ohjelmistoja on noin 140 kappaletta. KS-Tiedon tarjoama ohjelmistotuki koostuu pääsääntöisesti ohjelmistopäivitysten suorittamisesta sekä vikatilanteiden selvittämisestä. Perinteisten työasemalla suoritettavien ohjelmistojen tuen lisäksi KS-Tieto ylläpitää myös useita asiakkaiden käyttämiä ohjelmistojärjestelmiä sekä tarjoaa näitä järjestelmiä varten asiakastukea. Tuettavien ohjelmistojen suuri määrä aiheuttaa myös haasteita uusien työasemien käyttöönottoprosessille, koska osalla ohjelmistoista on erinäisiä esivaatimuksia, kuten riippuvuuksia muista ohjelmistoista tai lisenssirajoituksia, joista käyttöjärjestelmän ja ohjelmistojen asentamisen suorittajan tulee olla tietoinen.

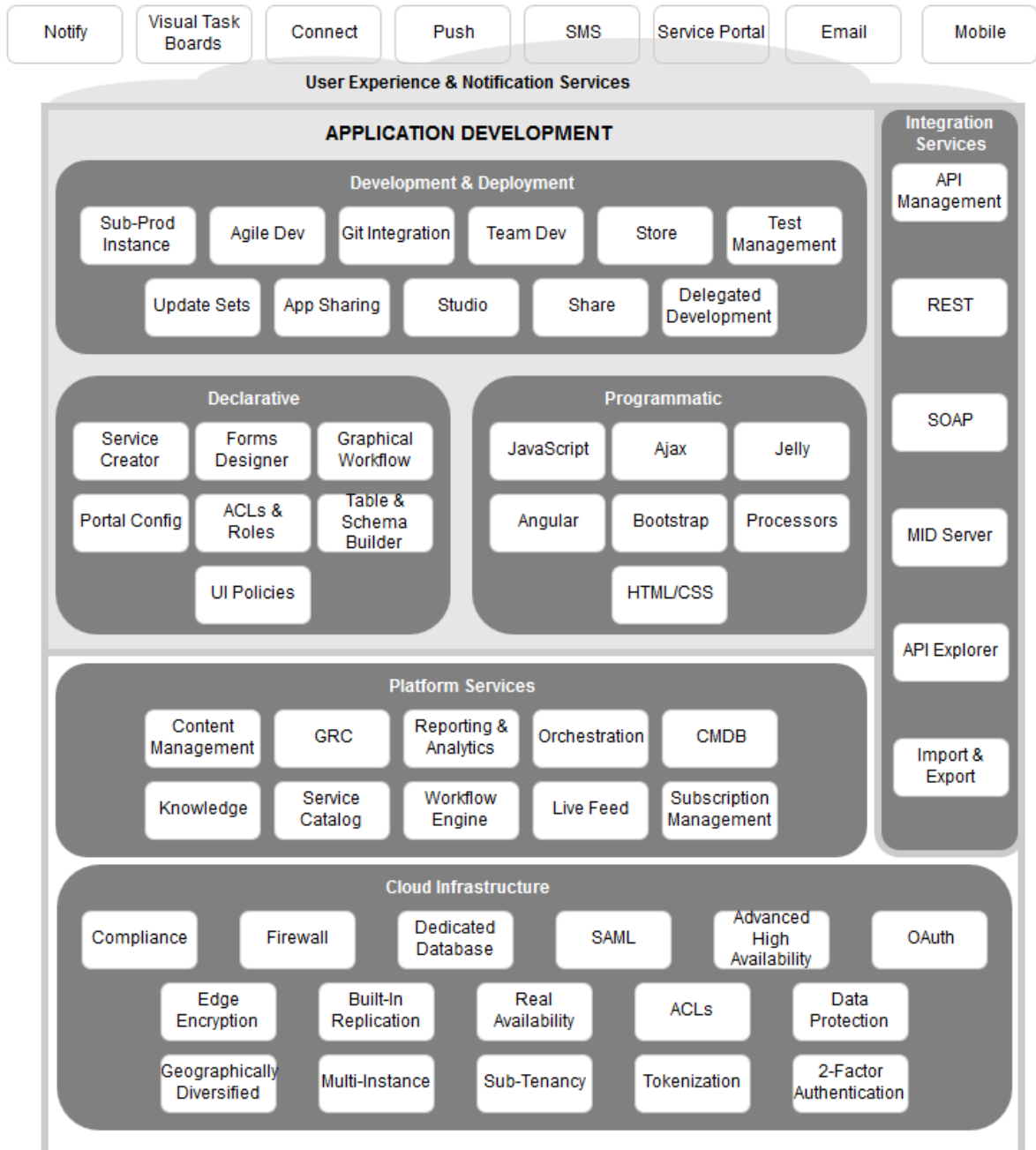
### **4.3 Konstruktion keskeisimmät ohjelmistot**

Tässä diplomityössä tuotettu konstruktio vaati valmistuakseen sekä toimiakseen useita erilaisia ohjelmistoja, tietojärjestelmiä ja palveluita joihin tullaan viittaamaan konstruktion kehityksen kuvauksessa useasti. Seuraavissa alaluvuissa kuvataan projektin kannalta olennaisimpien ohjelmistojen, tietojärjestelmien sekä palveluiden toimintaa, jotta konstruktion tuottamista kuvaavien lukujen seuraaminen ja ymmärtäminen olisi mahdollisimman selkeää.

#### **4.3.1 ServiceNow**

ServiceNow on pilvipalvelualusta, jonka avulla voidaan hallinnoida ja automatisoida monimuotoisia yritystoimintoja. ServiceNow-alustasta löytyy työkaluja yritysten palveluiden hallintaan, ICT-palveluiden hallintaan, ICT-toimintojen hallintaan, ICT-yritystoimintojen hallintaan sekä ohjelmistokehitykseen. Kuva 1 havainnollistaa ServiceNow'n rakennetta korkealla tasolla sekä alustan tarjoamien työkalujen

kattavuutta. ServiceNow on rakenteeltaan modulaarinen ja sen toiminnallisia ominaisuuksia pystytään helposti muokkaamaan ottamalla käyttöön liitännäisiä joko alustan graafisen käyttöliittymän kautta tai tilaamalla liitännäisen aktivointi pilvipalvelun toimittajalta. [28][29][30][31]



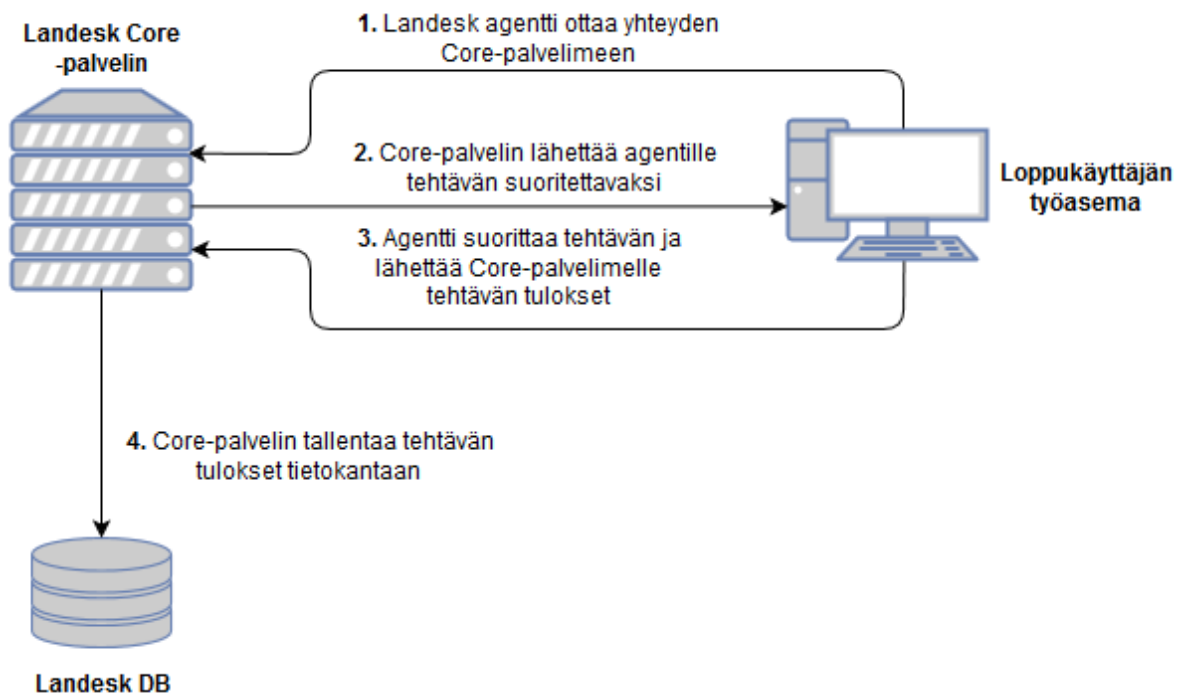
**Kuva 1.** ServiceNow-pilvipalvelualustan ominaisuuksia.

Tilaaajayrityksessä ServiceNow'ta käytetään pääasiallisesti ICT-palveluiden hallintaan. ServiceNow'ta käytetään ICT-palveluiden hallinnan suhteen muun muassa loppukäyttäjien ongelmatapausten kirjaamiseen, käsittelyyn ja seurantaan; loppukäyttäjien laitteisto- ja ohjelmistotilausten kirjaamiseen sekä käsittelyyn; ja tilaaajayrityksen ylläpidettävien laitteistojen ja ohjelmistojen seurantaan.

#### **4.3.2 LANDESK Management Suite**

LANDESK Management Suite on ICT-laitteistojen hallinnointiin tarkoitettu ohjelmisto. LANDESK Management Suiten ominaisuuksiin lukeutuu keskitetty työasemien hallinnointi, työasemien laitteistojen sekä ohjelmistojen tarkkailu, kolmannen osapuolen ohjelmistojen jakelu työasemille, käyttöjärjestelmien käyttöönotto, sisäverkkoon kytkettyjen laitteistojen tunnistaminen ja työasemien etäkäyttö. [32]

LANDESK Management Suite koostuu Core-keskuspalvelinohjelmistosta, tietokantapalvelimesta ja työasemille asennettavasta agenttiohjelmistosta ja Kuva 2 havainnoillistaa tämän järjestelmän rakennetta ja toimintaa. ICT-laitteistoja hallinnoidaan Core-palvelinohjelmiston kautta, joka kokoaa laitteistotietoja tietokantapalvelimelle työasemille asennettujen agenttiohjelmistojen avulla. Core-ohjelmisto jakaa agenttiohjelmistoille erilaisia tehtäviä suoritettavaksi, joita ovat esimerkiksi työaseman laitteisto- ja ohjelmistotietojen kerääminen sekä kolmannen osapuolen ohjelmistojen asentaminen. Core-ohjelmiston avulla on myös mahdollista suorittaa agenttiohjelmistoilla erittäin monimutkaisia konditionaalaisia tehtäviä, jotka mahdollistavat lähes minkä tahansa työaseman hallinnointiin liittyvän operaation automatisoimisen.



**Kuva 2.** LANDESK Management Suiten toimintaperiaate.

#### 4.3.3 Microsoft SQL Server

Microsoft SQL Server (Structured Query Language) on relaatiotietokantajärjestelmä Windows-käyttöjärjestelmille. SQL Serverin päätoiminnot ovat syötettyjen tietojen tallentaminen tietokantaan sekä tietojen noutaminen tietokannasta ulkopuolisille ohjelmistoille ja palveluille. SQL Serveristä löytyy kattava kokoelma työkaluja tietokantajärjestelmän hallinnoimiseen, joiden avulla pystytään esimerkiksi profiloimaan järjestelmän toimintaa optimointia tai ongelmatilanteiden ratkaisemista varten. [33][34]

#### 4.3.4 Microsoft Deployment Toolkit

Microsoft Deployment Toolkit (MDT) on kokoelma työkaluja Windows työpöytä- ja palvelinkäyttöjärjestelmien käyttöönottoprosessien automatisointia varten [35]. MDT:llä voidaan suorittaa käyttöjärjestelmän käyttöönotto kolmella erilaisella tavalla, jotka ovat Lite Touch Installation (LTI), Zero Touch Installation (ZTI) ja User Driven Installation (UDI) [36]. Taulukosta 1 selviää edellämainittujen käyttöönottoprosessien olennaisimpia ominaisuuksia. Tilaajayrityksessä käyttöönotetaan käyttöjärjestelmät LTI-prosessin avulla,



koska osa ylläpidettävistä työasemista sijaitsee tilaajayrityksen sisäverkon ulkopuolella. Taulukosta 1 voidaan havaita, että sisäverkon ulkopuolisiin, palomuurin takana oleviin työasemiin voidaan asentaa käyttöjärjestelmiä vain LTI-prosessin avulla ja tämä vaatimus rajaa muut käyttöönottoprosessit pois. Projektin toteutushetkellä sisäverkon ulkopuolella sijaitseviin työasemiin ei vielä suoritettu käyttöjärjestelmien asennuksia, mutta tällaisia asennuksia oli tarkoitus suorittaa lähitulevaisuudessa.

**Taulukko 1.** LTI, ZTI ja UDI-käyttöjärjestelmän käyttöönottopojen ominaisuuksia.

<b>LTI</b>	<b>ZTI</b>	<b>UDI</b>
Automaation taso määriteltävissä tarpeen mukaan	Vain täysin automatisoidut käyttöjärjestelmäasennukset	Automaation taso määriteltävissä tarpeen mukaan
Minimaaliset ohjelmistovaatimukset	Vaatii Microsoft System Center Configuration Manager 2007 R3 (SCCM) tai sitä uudemman version	Vaatii Microsoft System Center Configuration Manager 2007 R3 (SCCM) tai sitä uudemman version
Tukee verkon yli asentamista verkkokansiosta tai paikallisesti asennusmedialta	Tukee verkon yli asentamista SCCM 2007 R3 tai sitä uudemman version avulla. Tukee asentamista paikallisesti asennusmedialta	Tukee verkon yli asentamista SCCM 2007 R3 tai sitä uudemman version avulla. Tukee asentamista paikallisesti asennusmedialta
Asennukset voidaan suorittaa hitaiden ja epävakaiden verkkoyhteyksien yli	Asennukset vaativat nopean ja vakaan verkkoyhteyden	Asennukset vaativat nopean ja vakaan verkkoyhteyden
Mahdollista asentaa käyttöjärjestelmä palomuurin takana oleville työasemille	Vaatii Remote Procedure Call -yhteyden asennuskohteeseen	Vaatii Remote Procedure Call -yhteyden asennuskohteeseen
Saattaa olla nopeampi konfiguroida kuin ZTI- ja UDI-asennukset	Vaatii paljon konfigurointia	Vaatii paljon konfigurointia
Kasvattaa konfiguraatiovirheiden riskiä	Vähentää konfiguraatiovirheiden riskiä	Kasvattaa konfiguraatiovirheiden riskiä

### **4.3.5 Active Directory**

Active Directory on Microsoftin kehittämä hakemistopalvelu Windows-toimialueita varten. Active Directoryn avulla määritetään toimialueen verkossa sijaitseville resursseille, joita ovat esimerkiksi työasemat, hakemistot ja tiedostot, yksilölliset tunnisteet. Näiden yksilöllisten tunnisteiden avulla resurssien hallinnointi ja seuranta on helpompaa verrattuna fyysisten tunnisteiden avulla tapahtuvaan hallinnointiin. [37]

Active Directoryyn kuuluvan Active Directory Domain Services -palvelun avulla myös autentikoidaan toimialueen jäsenet, kuten käyttäjät ja tietokoneet, sekä määritetään jäsenten käsittelyoikeudet hakemistopalvelun resurssien suhteen. Resurssien käsittelyoikeuksia usein hallinnoidaan ryhmien avulla, joihin voidaan liittää muun muassa käyttäjiä, tietokoneita ja muita ryhmiä. [38]

Active Directoryä on mahdollista hallinnoida graafisella työkalulla, Windows PowerShellillä, Active Directory Service Interfaces -sovellusohjelmointirajapinnalla (ADSI), avoimen standardin Lightweight Directory Access Protocol -sovellusohjelmointirajapinnalla tai .NET ohjelmistokehyksen System.DirectoryServices nimiavaruuden avulla. [37][39][40]

### **4.3.6 PowerShell**

Windows PowerShell on komentotulkki ja komentosarjakieli Windows-käyttöjärjestelmille. PowerShell-komentotulkilla voidaan muun muassa hallita interaktiivisesti useita Microsoftin kehittämiä Windows-käyttöjärjestelmissä toimivia ohjelmistotuotteita, kuten Active Directoryä, Cmdlet-konsoliohjelmien avulla sekä automatisoida Windows-käyttöjärjestelmien hallintatoimenpiteitä käyttämällä komentosarjaohjelmia, joita kutsutaan myös skripteiksi. [41][42][43]

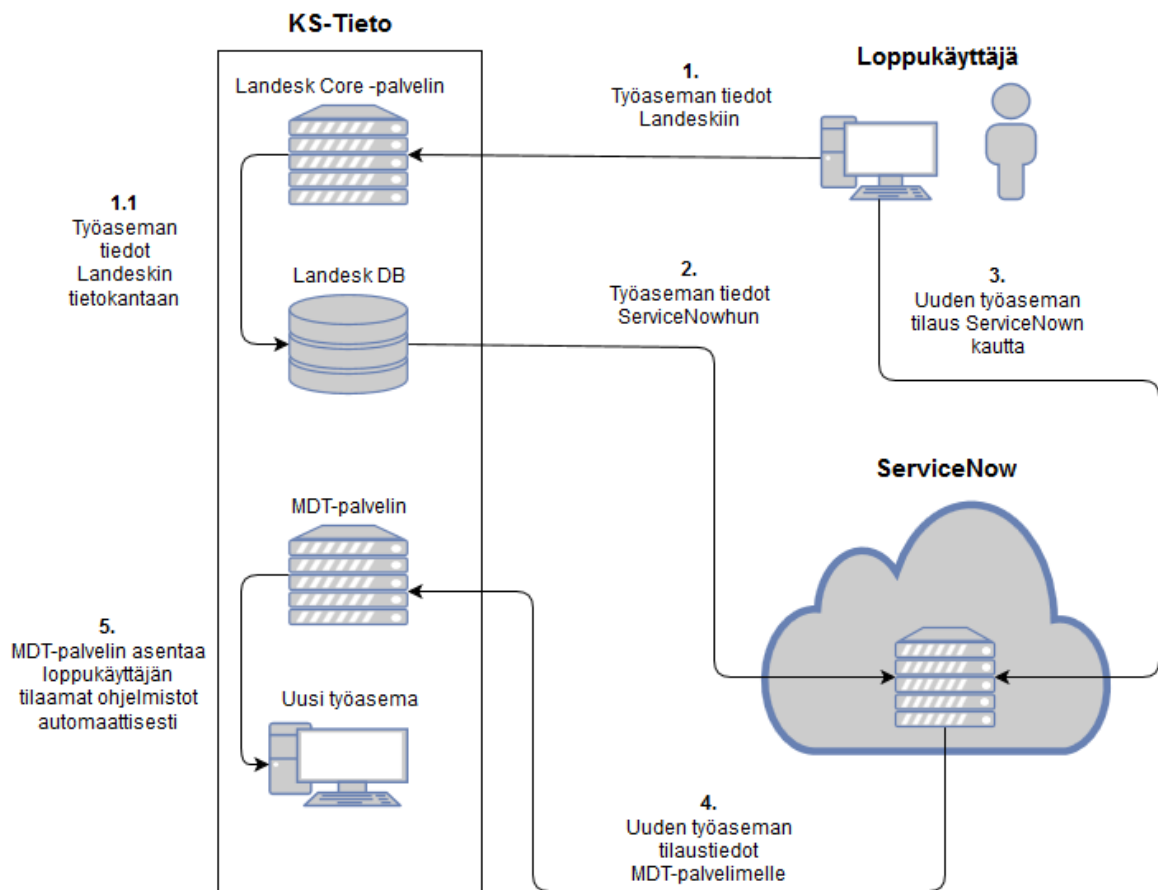
## **4.4 Konstruktion toteutus**

Työssä toteutetun konstruktion tarkoituksena oli automatisoida uudelle työasemalle tilattujen ohjelmistojen asentaminen käyttöjärjestelmäasennuksen yhteydessä. Tilaajayrityksen tekemän alustavan selvityksen mukaan ohjelmistoasennusten

automatisointi oli saavutettavissa integroimalla kolme tilaajayrityksen käytössä olevaa ohjelmistoa keskenään. Seuraavissa alaluvuissa kuvataan integraatioprojektin vaiheita sekä esitellään kehitystyön tuloksena syntynyt konstruktio.

#### **4.4.1 Konstruktiokehityssuunnitelma**

Tilaajayrityksen selvityksen mukaan integraatioprojekti olisi ollut kolmivaiheinen. Alustavan selvityksen mukaan projekti oli myös toteutettavissa ilman ylimääräisiä investointeja ja projekti tulikin toteuttaa tilaajayrityksessä käytössä olevien ohjelmistojen ja resurssien avulla. Ensimmäisessä vaiheessa tuli hakea LANDESK Management Suitesta, johon viitataan jatkossa pelkällä Landesk nimellä, käytössä olevien työasemien asennettujen ohjelmistojen tiedot ServiceNow-pilvipalveluun ja yhdistää ne ServiceNow'sta löytyviin työasemien laitetietueisiin. Toisessa vaiheessa tuli noutaa uuden työaseman tilaajan tilaushetkellä käytössä olevalle työasemalle asennettujen ohjelmiston tiedot sekä tilattujen uusien ohjelmistojen tiedot ServiceNow'sta MDT:hen. Viimeisessä vaiheessa oli tarkoitus konfiguroida MDT asentamaan tilatut ohjelmistot työaseman käyttöjärjestelmän asennuksen aikana. Alustavan suunnitelman mukaisen integraatiojärjestelmän rakennetta selventää Kuva 3.



**Kuva 3.** Integraatiosuunnitelma.

Projekti oli tarkoitus toteuttaa yhdessätoista viikossa, joka oli selvityksen tehneiden työnohjaajien mukaan realistinen tavoite ajallisesti. Projektia varten suunniteltu aikataulu selviää Taulukosta 2.

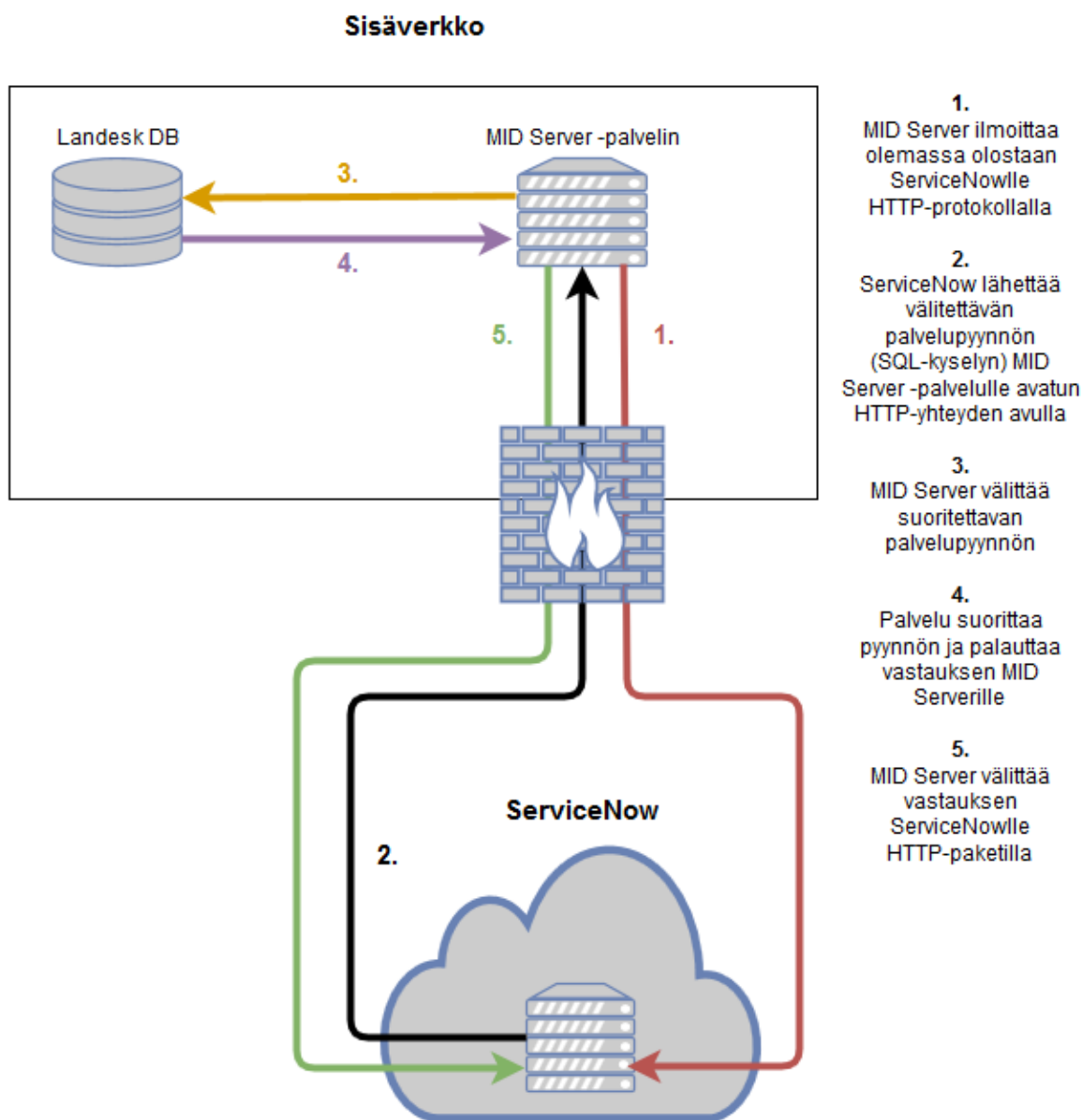
**Taulukko 2.** Projektin aikataulu

<b>Viikko</b>	<b>Tavoitteet</b>
1	Projektiympäristöön perehtyminen
2	Testausympäristön luominen Landesk–ServiceNow-integraatiota varten, Landesk-tietokannan duplikaatio testausympäristöön, Yhteyden muodostaminen testausympäristön tietokantaan ServiceNow’sta.
3	Landeskin tietokantaan tehtävien kyselyjen kehittäminen, kyselyjen tulosten käsittely ServiceNow’ssa.
4	Landeskin tietokantaan tehtävien kyselyjen kehittäminen, kyselyjen tulosten käsittely ServiceNow’ssa.
5	Landesk–ServiceNow-integraation testaaminen.
6	ServiceNow’n ja MDT:n välisen yhteyden muodostaminen.
7	ServiceNow’sta haettavien tietojen käsittelyn kehittäminen.
8	Ohjelmistojen asentaminen MDT:n avulla.
9	ServiceNow–MDT-integraation testaaminen.
10	Kehitetyn järjestelmän testaamista kokonaisuudessaan sekä järjestelmällä saavutettavien hyötyjen mittaamista.
11	Kehitetyn järjestelmän testaamista kokonaisuudessaan sekä järjestelmällä saavutettavien hyötyjen mittaamista.

#### **4.4.2 LANDESK Management Suite – ServiceNow -integraatio**

Landesk–ServiceNow-integraatiossa oli tarkoituksena automatisoida työasematietojen hakemista Landeskin tietokannasta ServiceNow’n tietokantaan. ServiceNow tarjoaa osana alustan ydinkomponentteja MID Server -nimisen sovelluksen, jonka avulla voidaan hallita asiakkaan verkossa toimivia palveluja monipuolisesti ServiceNow-pilvipalvelun kautta ja tämän sovelluksen avulla pystytään muun muassa suorittamaan SQL-kyselyjä pilvipalvelun puolelta [44][45]. Kuva 4 havainnoillistaa miten MID Server toimii asiakkaan sisäverkossa ottaen yhteyden ServiceNow-pilvipalveluun, josta seuraa, ettei MID Serveriä käyttävän asiakkaan tarvitse erikseen avata reitityksiä palomuriin

saadakseen yhteyden ServiceNow'hun. Tilaaajayrityksen Landesk on konfiguroitu käyttämään Microsoft SQL Server 2008 R2 -tietokantaa ja ServiceNow'n MID Serverin avulla voidaan suorittaa SQL-kyselyjä Landeskin tietokantaan suoraan pilvipalvelusta yksinkertaisen konfiguroinnin jälkeen [45].



**Kuva 4.** MID Serverin toimintaperiaate.

Landesk–ServiceNow-integraatio aloitettiin asentamalla ServiceNow'n MID Server tilaaajayrityksen sisäverkossa sijaitsevalle palvelimelle, jonka jälkeen MID Server konfiguroitiin ottamaan yhteys Landeskin tietokantaan sekä ServiceNow-pilvipalveluun. MID Serverin konfiguroinnin jälkeen pilvipalveluun asetettiin tietokantayhteyden vaatimat

tiedot ja tietokantayhteys testattiin yksinkertaisella SQL-kyselyllä, jolla varmistettiin että tietokannasta pystytään ylipäättään noutamaan dataa ja että SQL-kysely tuottaa oikeanlaisia tuloksia.

Tietokantaintegraation kehitys aloitettiin tekemällä SQL-kysely, joka haki Landeskin tietokannasta yksinkertaisia tietoja työasemista, joita olivat esimerkiksi työaseman toimialuenimi ja työasemalle asennettu käyttöjärjestelmä. Kun halutut tietueet saatiin noudettua ServiceNow'hun, täytyi ne yhdistää ServiceNow'n oman tietokannan tietueisiin. Noudettujen työasematietueiden yhdistäminen ServiceNow'n tietokannan tietueisiin oli melko yksinkertainen operaatio, joka onnistui ServiceNow'n graafisen käyttöliittymän kautta vaivattomasti.

Työasemille asennettujen ohjelmistojen hakeminen Landeskin tietokannasta ServiceNow'hun osoittautui huomattavasti monimutkaisemmaksi operaatioksi kuin mitä alustavasti oli arvioitu. ServiceNow'n puolella ongelmat johtuivat pääsääntöisesti siitä, ettei ohjelmistotietueiden nimeämiseen oltu kehitetty tilaajayrityksen sisäistä standardia vaan ohjelmistotietueet oli nimetty melko mielivaltaisesti. Useat ServiceNow'hun luoduista ohjelmistotietueista oli nimetty tilaajayrityksessä käytettyjen puhekielisten ohjelmistonimien mukaan, joka vaikeutti oikeiden ohjelmistonimien hakemista Landeskistä. Osa ohjelmistotietueista oli useamman ohjelmiston paketteja, jotka asennettiin erillisinä paketteina ja osa näistä pakettiohjelmistoista löytyi myöskin erillisinä ohjelmistotietueina ServiceNow'ssa. Ohjelmistotietueiden nimiin liittyvän sekavuuden lisäksi jokaisesta ohjelmistotietueesta oli useita kopioita, joiden ainoana erona oli asiakasyrityskentän sisältö. Asiakasyrityskenttä mahdollistaa yrityskohtaisen ohjelmistojen hakusuodatuksen, mutta vastapainona se hankaloittaa tietueiden käsittelyä yleisellä tasolla. Tilaajalle ehdotettiin asiakasyrityskentän muuttamista niin että se olisi listaviittaus yrityksiin, joille ohjelma voidaan asentaa, jolloin yksittäistä ohjelmistoa kohtaan olisi vain yksi ohjelmistotietue ServiceNow'n tietokannassa. Asiakasyrityskenttään ehdotettua muutosta ei kuitenkaan toteutettu projektin aikana, koska se olisi vaatinut kattavan selvitystyön siitä, miten se olisi vaikuttanut muihin tietueisiin ja selvitystyön lisäksi manuaalisesti tehtävä muutostyö olisi myöskin vaatinut lukuisia työtunteja.



Landeskiin liittyvät ongelmat olivat samankaltaisia ServiceNow'n ongelmien kanssa. Landeskin tietokannassa näkyvien ohjelmistojen nimet saattoivat olla hyvin samankaltaisia toistensa kanssa, vaikka niiden käyttötarkoitukset saattoivat poiketa toisistaan huomattavasti. On myös mahdollista, että ohjelmiston nimi muuttuu eri versioiden välillä merkittävästi, josta seuraa, että aina kun jokin ylläpidettävä ohjelmisto päivitetään uudempaan versioon, on tarkistettava Landeskin tietokannasta muuttuuko ohjelmiston Landeskille näkyvä nimi. Samankaltaiset nimet eivät itsessään olleet varsinainen ongelma, mutta kun niitä pyrittiin yhdistämään jossain määrin mielivaltaisesti nimettyihin ServiceNow-ohjelmistotietueisiin niin tietueiden yhdistämistyö muodostui vaivalloiseksi ja aikaa vieväksi toimenpiteeksi.

Työasemille asennettujen ohjelmistojen seuranta vaatii tässä tapauksessa manuaalisesti ylläpidettävää Landeskin tietokantaan tehtävää SQL-kyselyä, joka täytyy tarkistaa ja mahdollisesti myös päivittää aina, kun jokin ylläpidettäväksi listatuista ohjelmistoista päivitetään uuteen versioon, koska Landeskin tietueisiin ei käytännössä pysty tekemään muutoksia. Toisekseen ServiceNow'n puolella täytyy ylläpitää manuaalisesti useita JS-skriptejä (JavaScript) joiden avulla noudetut tietueet yhdistetään oikeisiin ohjelmisto-yritystietueisiin. Kolmas manuaalisesti ylläpidettävä artefakti tulisi olemaan ServiceNow-ohjelmistotietueiden muuntaminen MDT:lle sopivaan muotoon ohjelmistojen varsinaisessa asennusvaiheessa.

Kun tietokantaintegraatiota varten kehitetty ratkaisu oli onnistuneesti testattu tulimme työn käytännön ohjaajien kanssa siihen tulokseen, että Landeskin tietokantaan tehtävät ohjelmistotietueiden SQL-kyselyt ajettaisiin vain kerran, jotta ServiceNow'n ohjelmistotietueet saataisiin ajan tasalle. Kun ServiceNow'n tietokanta päivitetäisiin kerran ajan tasalle tulisi se ylläpitämään itsensä automaattisesti ServiceNow'n kautta tehtävien työasema- sekä ohjelmistoasennustilausten myötä ja manuaalisesti ylläpidettäville Landesk tietokannan SQL-kyselyille ei tällöin olisi tarvetta. Tästä seuraisi, että ainoa manuaalisesti ylläpidettävä artefakti olisi ServiceNow'n ohjelmistotietueiden muuntaminen MDT:lle sopivaan muotoon.

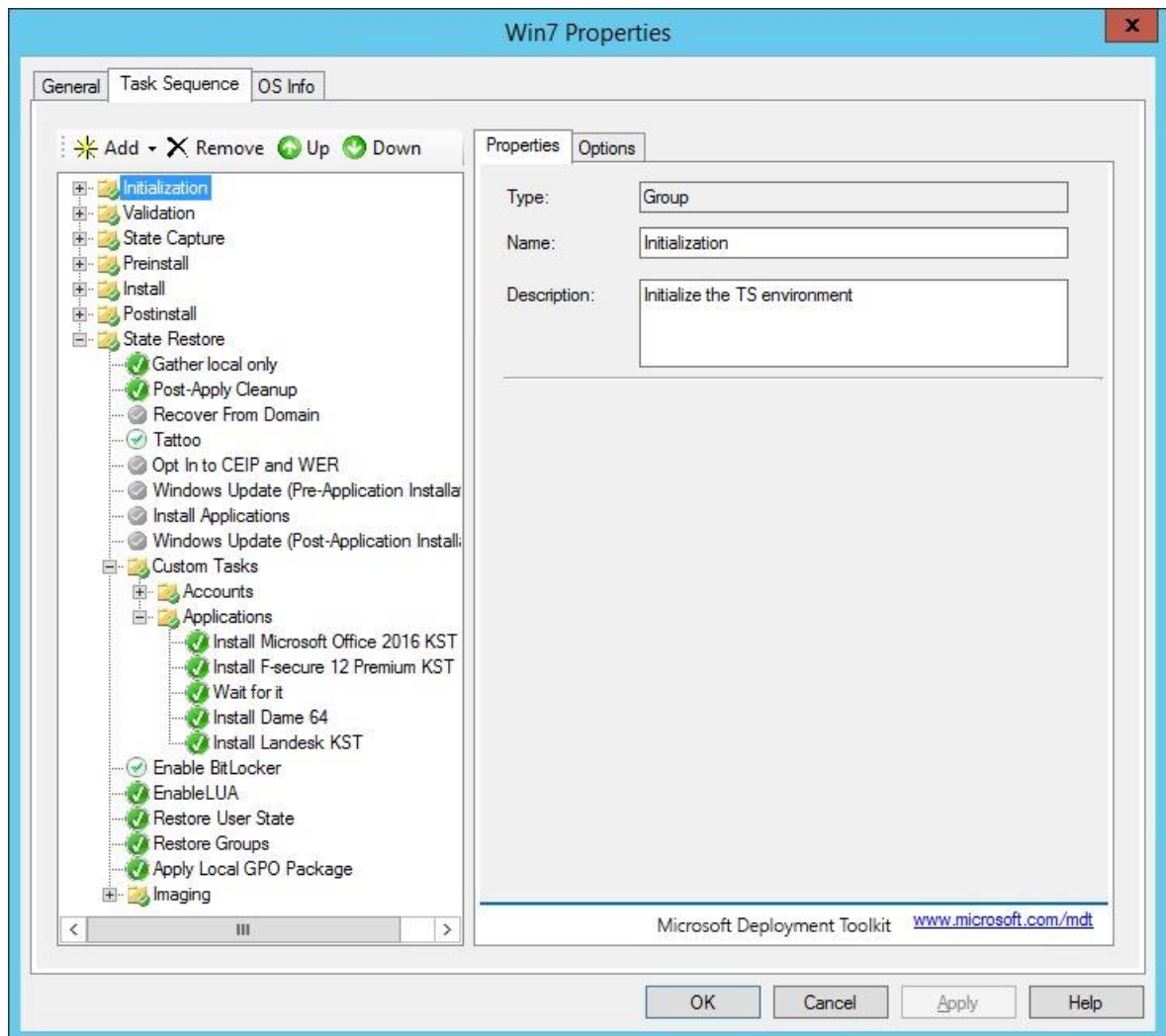
#### 4.4.3 ServiceNow – Microsoft Deployment Tool -integraatio

ServiceNow–MDT-integraation ydinkysymys oli kuinka ServiceNow’sta saadaan noudettua uudelle työasemalle asennettavat ohjelmistot. ServiceNow’hun saa haettua vaivattomasti dataa useissa eri formaateissa graafisen käyttöliittymän kautta, mutta datan noutamiseen ServiceNow’sta kolmannen osapuolen järjestelmään ei ole olemassa valmiita ratkaisuja. ServiceNow’n tietokannan dataa pystyy kuitenkin käsittelemään yleisimpien verkkopalveluprotokollien, kuten REST (Representational State Transfer), Simple Object Access Protocol (SOAP) tai Web Service Description Protocol (WSDL), avulla [46]. Ohjelmistotietojen noutaminen jonkin verkkopalveluprotokollan avulla vaati, että sitä varten joko kehitetään sovellus tai PS-skripti (PowerShell), jonka avulla tiedot noudetaan ja muunnetaan MDT:lle sopivaan muotoon. Täysiverisen sovelluksen kehittäminen tietojen noutamista varten todettiin liian raskaaksi prosessiksi projektiin varatun rajoitetun ajan vuoksi ja päädyimme ratkaisemaan ongelman PS-skriptin avulla. Käytettäväksi verkkopalveluprotokollaksi valikoitui suorat HTTP-palvelupyynnöt (Hypertext Transfer Protocol), jotka palauttivat vastaukset XML-muodossa, koska kyselyiden muodostaminen oli suoraviivaista ja XML-tiedostojen koneellinen lukeminen on helppo toteuttaa useilla työkaluilla. HTTP-palvelupyyntöjen valikoitumiseen tietojen haussa käytettäväksi protokollaksi vaikutti myöskin se, että PowerShell tarjoaa versiosta 3.0 eteenpäin `Invoke-WebRequest` komentorivityökalun, jonka avulla pystytään suorittamaan HTTP-palvelupyyntöjä erittäin pienellä vaivannäöllä [47].

Tilaaajayrityksessä käyttöjärjestelmän asentaminen MDT:n avulla toimii yksinkertaistettuna seuraavanlaisesti:

0. Luodaan MDT-palvelimelle asennussekvenssi, jonka avulla käyttöjärjestelmä asennetaan.
1. Käynnistetään Windows Preinstallation Environment (WinPE) niminen alkulatausohjelma työasemalle joko fyysiseltä medialta tai verkon yli.
2. Asetetaan WinPE:ssä manuaalisesti uuden työaseman toimialuenimi ja valitaan asennettavan käyttöjärjestelmän asennussekvenssi MDT-palvelimelta
3. WinPE lataa valitun asennussekvenssin MDT-palvelimelta ja asentaa sekvenssin mukaisen käyttöjärjestelmän
4. Asennetaan tilatut ylimääräiset ohjelmistot manuaalisesti

Asennussekvenssin luominen on esivaatimus, joka täytyy tehdä vain kerran. Yksittäinen asennussekvenssi koostuu vaiheista, jotka jakavat käyttöjärjestelmän asennusprosessin loogisiksi osiksi, sekä askeleista, joiden avulla määritetään käyttöjärjestelmän asennusprosessissa suoritettavat toiminnot yksityiskohtaisesti. Konstruktion toiminnan kannalta olennaisin vaihe oli Kuvassa 5 näkyvä State Restore, jonka sisältä löytyvässä Custom Tasks alavaiheessa on mahdollista suorittaa komentorivi- ja PS-skriptejä. Kuvassa 5 näkyy myös Install Applications – askel, mutta kyseiseen askeleeseen määritettävät ohjelmistot ovat staattisia, eli asennettavia ohjelmistoja pystyy muokkaamaan vain kääntämällä asennussekvenssi uudelleen. Tilaaajayrityksessä ohjelmistoja oli jo aiemmin asennettu Custom Tasks -vaiheen Applications -aliaskeleessa yksinkertaisten komentoriviskriptien avulla, mutta käytännössä nämäkin ohjelmistoasennukset olivat staattisia, koska nämä asennusskriptit vain määrittävät asennettavan ohjelmiston asennusparametrit ilman ehtolausekkeita. ServiceNow-MDT-integraation yhtenä tavoitteena olikin toteuttaa työaseman tilaajan tilaamien ohjelmistojen dynaaminen asentaminen olemassa olevia asennusskriptejä monimutkaisemmilla PowerShell-skripteillä Custom Tasks -vaiheen aikana. [36]



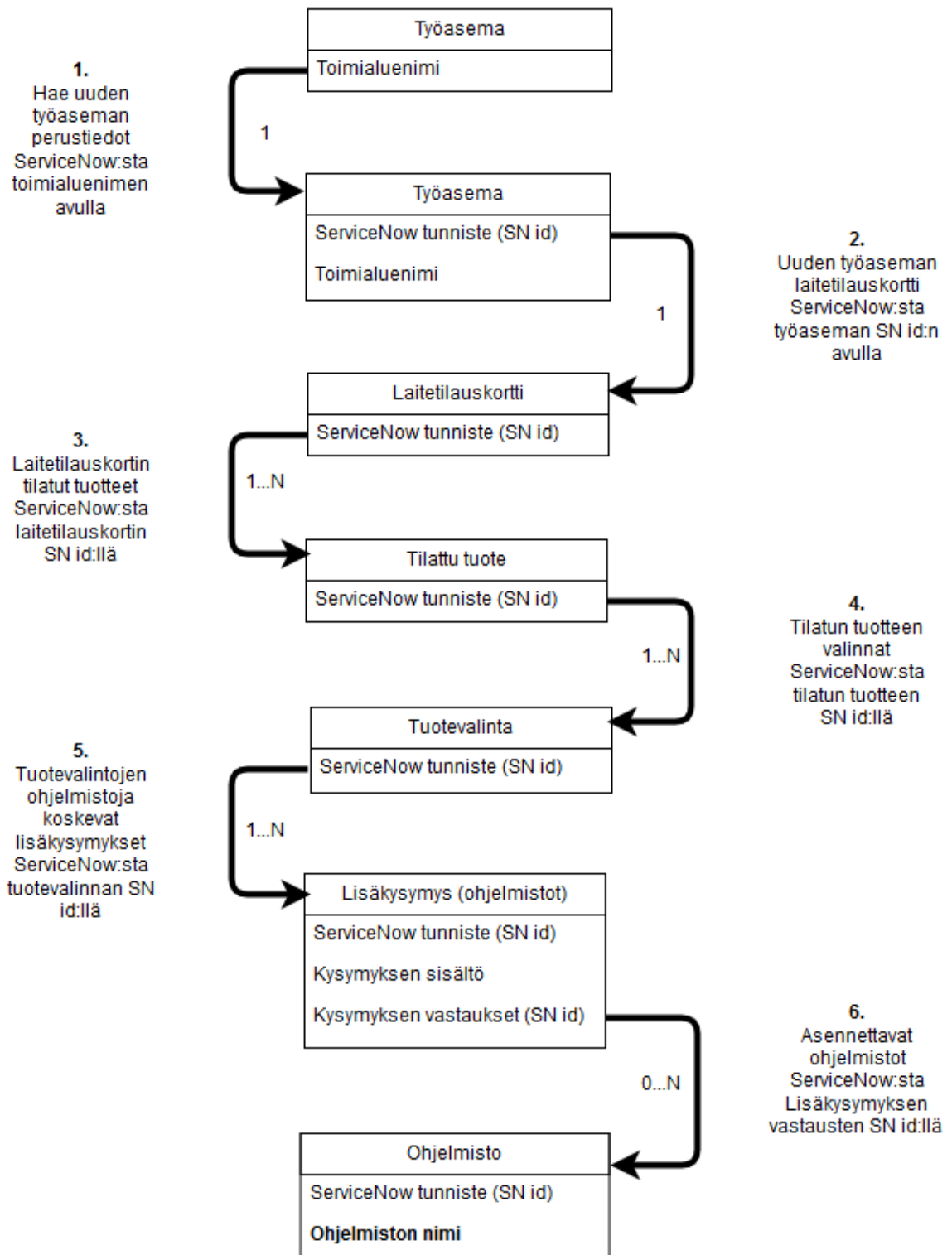
**Kuva 5.** Microsoft Deployment Toolin asennussekvenssi.

Ensimmäinen listattu kohta on välttämätön käyttöjärjestelmän asennuksen kannalta ja siihen ei liity mitään erikoisia toimenpiteitä. Toisen vaiheen manuaaliset toimenpiteet ovat automatisoitavissa MDT:n ominaisuuksien avulla, mutta toimenpiteiden automatisointi rajattiin konstruktion vaatimusten ulkopuolelle, koska ne eivät suoranaisesti liity ohjelmistoasennusten automatisointiin. Asennussekvenssin lataus ja suoritus on konstruktion toiminnan kannalta keskeisin vaihe, koska asennussekvenssin suorituksen aikana oli tarkoitus suorittaa työaseman tilaajan tilaamien ohjelmistojen asentaminen PowerShell-skriptien avulla, jonka myötä neljäs listattu vaihe olisi poistunut kokonaan.

Käyttäjärjestelmän asennuksen aikana työasemalla suoritettavan PS-skriptin ensimmäinen algoritmiversio oli seuraavanlainen:

1. Lue asennuksessa olevan työaseman toimialuenimi
2. Nouda HTTP-palvelupyynnöllä asennettavan työaseman laitetilaustietue työaseman toimialuenameä käyttäen ServiceNow'sta
3. Lue noudetusta laitetilaustietueesta työasemalle asennettavat ohjelmistot
4. Asenna ohjelmistot verkkoasemalta

Työaseman toimialuenumien selvittäminen PS-skriptillä on triviaali toimenpide ja tilatun työaseman laitetilaustietueen noutaminen ServiceNow'sta onnistui toimialuenumien avulla helposti. Asennettavien ohjelmistojen selvittäminen laitetilaustietueelta oli puolestaan työläs ja lukuisia työtunteja vaatinut vaihe, koska ohjelmistot löytyvät laitetilaustietueelta usean yksilöllisen järjestelmätunnisteviittauksen takaa ja nämä viittaukset täytyi selvittää manuaalisesti tarkastelemalla järjestelmätunnisteiden merkityksiä ja suhteita. Kuva 6 havainnoillistaa asennettavien ohjelmistojen selvittämiseen vaadittuja vaiheita. Tämän jälkeen asennettavat ohjelmistot täytyi vielä yhdistää oikeisiin asennusohjelmien verkkopolkuihin, joka olisi ollut työläs, vaikkakin yksinkertainen, toimenpide mikäli kaikkien ohjelmistojen olisi täytynyt olla saatavilla. Päädyimme kuitenkin yhdistämään aluksi vain viisi ohjelmistoa niiden asennuspolkuihin helpottaaksemme skriptin testaamista sekä nopeuttaaksemme projektin etenemistä.



**Kuva 6.** Asennettavien ohjelmistojen noutaminen ServiceNow’sta.

Tässä vaiheessa ilmeni, että työasemille aiotaan asentaa käyttöjärjestelmiä jatkossa myös etänä, kun aiemmin käyttöjärjestelmät oli aina asennettu tilaajayrityksen toimipisteellä. Käyttöjärjestelmän etäasennus tarkoitti, että tilattujen ohjelmistojen asennuksesta vastaava skripti olisi myös viety asennuksen aikana etänä uudelleen asennettavalle työasemalle. MDT:n asennussekvenssin suorittamat skriptit viedään salaamattomina asennuskohteeseen ja koneen äärellä olevä käyttäjä pystyy tarkastelemaan skriptejä käyttöjärjestelmän asennuksen aikana. Skriptin viemisessä etätyöasemalle ei itsessään ollut mitään ongelmaa, mutta skripti vaati ServiceNow'n sekä toimialueen hallinnointitason tunnuksia toimiakseen ja näiden tunnusten vieminen etätyöasemalle ymmärrettiin mahdolliseksi tietoturvariskiksi. Powershellistä löytyy SecureString-tietotyyppi, jonka avulla voidaan salata tekstiä sisältävän muuttujan kuten esimerkiksi tunnuksen salasanan sisältö. Valitettavasti Powershellin SecureString-tietotyypit ovat sidottuja siihen työasema-käyttäjä yhdistelmään, jolla kyseinen tietotyyppi on luotu, joka rajasi ulos SecureString-muodossa tallennettujen tunnusten käyttämisen [45]. Käyttäjätunnusten turvallinen käsittely oli vaikea ongelma ratkaistavaksi, vaikka tunnukset olisi viety muulla tavoin salattuna olisi salauksen purkava avain pitänyt saada etätyöasemalle turvallisesti. Yksi mahdollinen vaihtoehto vaikutti olevan tunnuspalvelun kehitys, jolta skripti olisi pyytänyt tunnukset salatun yhteyden yli. Tunnuspalvelun olisi kuitenkin täytynyt tunnistaa asennuksessa oleva työasema sekä skripti jollakin tavalla, joka johti taas alkuperäiseen ongelmaan. Eräs vaihtoehto oli koko ohjelmistoja asentavan skriptin suorittaminen salatun yhteyden yli luotetulta työasemalta, mutta tämä vaihtoehto jouduttiin hylkäämään koska tilaajayrityksen tietoturvapoliittikan puitteissa PS-skriptien suorittaminen etänä ei ollut mahdollista.

#### **4.4.4 Muutos suunniteltuun järjestelmään**

Etsittäessä ratkaisua tunnusten turvalliseen käsittelyyn käyttöjärjestelmän MDT-asennuksen yhteydessä mietimme myös vaihtoehtoisia tapoja asentaa tilatut ohjelmat. Landeskin avulla pystytään asentamaan ohjelmistoja turvallisesti ilman valvontaa, mutta ongelmana oli, että miten Landesille välitettäisiin tieto uuden ohjelmiston asennustarpeesta [49]. Kävi ilmi, että tilaajayrityksessä oli ollut suunnitelmana luoda kaikille sovelluksille Active Directory -ryhmät, joiden avulla työasemille asennettuja ohjelmistoja pystyttäisiin osittain hallinnoimaan. Landesk pystyy lukemaan työaseman Active Directory -tietoja ja täten ohjelmistoasennus pystytään käynnistämään, kun Landesk

havaitsee että työasema on liitetty uuteen ohjelmistoryhmään [50]. Seuraavaksi täytyi ratkaista, miten työasema liitetään oikeisiin Active Directory -ohjelmistoryhmiin turvallisesti. Käyttöjärjestelmän MDT-asennusta pystyy seuraamaan MDT-palvelimella suoritettavan PS-skriptin avulla ja PS-skripteillä pystyy myös hallinnoimaan Active Directorya, joka mahdollistaa työasemien liittämisen oikeisiin ohjelmistoryhmiin käyttöjärjestelmän asennuksen aikana [51][52][53].

Ennen kuin asennusprosessin muutos lyötiin lukkoon oli kuitenkin punnittava muutoksesta mahdollisesti koituvia hyötyjä ja haittoja. Suurimpana hyötynä oli, että hallinnointitason tunnuksia ei tarvitsisi lainkaan viedä asennettavalle työasemalle, koska logiikasta vastaavaa PS-skriptiä suoritettaisiin tilaajayrityksen sisäverkossa sijaitsevalta palvelimelta. Toisaalta PS-skriptiä jouduttaisiin edelleen kehittämään, jotta skripti pystyisi tarkkailemaan MDT-asennuksien tilaa sekä lisäämään työasemia Active Directoryn ohjelmistoryhmiin.

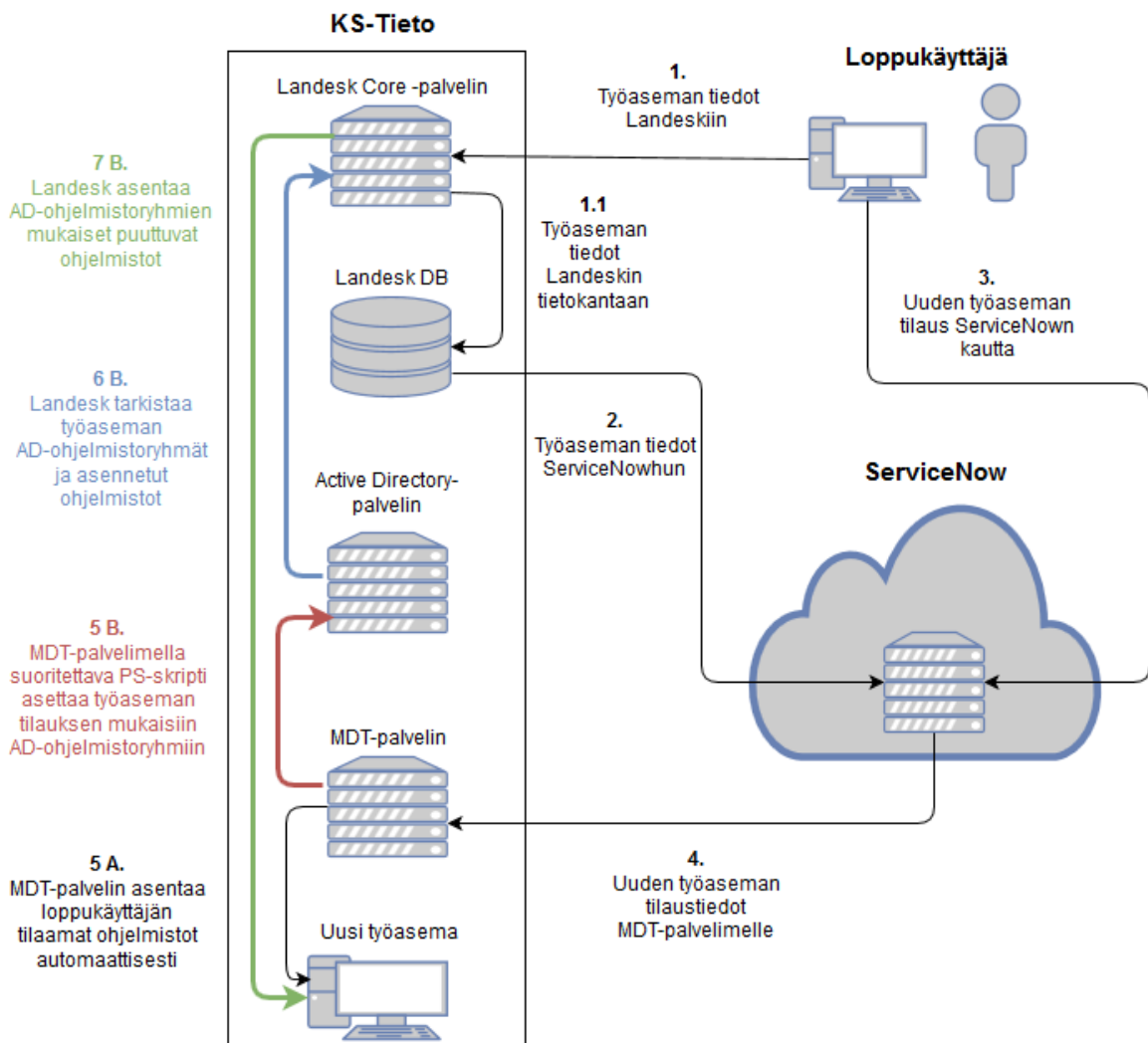
Active Directoryn hyödyntämistä puolsi kolmannen osapuolen ohjelmistojen, mukaan lukien ServiceNow'n ja Landeskin, tarjoama yhteensopivuus, joka helpottaisi myös tulevaisuudessa mahdollisesti tehtäviä muita ohjelmistoja käsitteleviä integraatioita. Toisena puoltavana tekijänä oli, että Powershellin Active Directory -moduulin avulla Active Directorya voidaan hallinnoida tehokkaasti ja tarvittaessa Active Directoryä voidaan myös hallinnoida avoimella LDAP-protokollalla [37]. Ongelmallista Active Directoryn ja ohjelmistolistojen hyväksikäytössä oli että ohjelmistoryhmien käyttämisestä oli vasta suunniteltu tilaajayrityksessä. Koska ohjelmistoryhmien hyödyntäminen oli vasta suunnitteluasteella, projekti tulisi täten jäämään "Proof-of-Concept" prototyypiksi.

Tilaajayrityksessä asennettiin ja ylläpidettiin ohjelmistoja Landeskillä ennen projektin aloitusta ja Landeski koettiin jo pelkästään tämän vuoksi hyväksi ratkaisuksi projektin kannalta. Landeskin avulla pystytään siis asentamaan ohjelmistoja ilman valvontaa turvallisesti, mutta ennen Active Directoryn avulla tapahtuvaa ohjelmistotilauksen välitystä, ominaisuutta ei pystytty hyödyntämään projektissa helposti. Landeski-Active Directory -yhdistelmä mahdollistaa myös käytössä olevien työasemien ohjelmistotilausten automatisoinnin, joka ei olisi onnistunut alunperin suunnitellulla ratkaisulla. Landeskillä



suoritettavissa ohjelmistoasennuksissa ei vaikuttanut olevan lainkaan huonoja puolia ja ominaisuuden hyödyntämistä pidettiin hyvänä ratkaisuna.

Muutoksesta syntyvien hyötyjen todettiin ylittävän haitat ja järjestelmän rakennetta päätettiin muuttaa. Kuvasta 7 selviää alunperin suunnitellun ja päivitetyn ohjelmistojen asennusprosessin väliset erot. Samassa yhteydessä jouduttiin myös toteamaan, että integraatiojärjestelmä tulisi jäämään prototyypiksi Active Directory -ohjelmistoryhmistä johtuvan epävarmuuden takia, eikä projektissa syntyvää järjestelmää voitaisi välttämättä ottaa tuotantokäyttöön sellaisenaan. Active Directory -ryhmien vuoksi jouduttiin myös luopumaan vanhan työaseman ohjelmistotietojen hyödyntämisestä uuden työaseman käyttöönotossa, koska vain murto-osalle tuetuista ohjelmistoista luotaisiin ryhmät ja kyseisen ominaisuuden testaaminen olisi hankalaa kuvatun rajoitteen puitteissa.



**Kuva 7.** Muutos järjestelmäintegraatioon.

#### 4.4.5 Microsoft Deployment Tool – Active Directory -integraatio

Kun käyttöjärjestelmän asennus on käynnistetty MDT:llä, voidaan asennusprosessin etenemistä seurata MDT-palvelimelta. Työaseman asennusta tarkkaileva palvelu kertoo perustietoja asennuksessa olevasta tietokoneesta ja asennuksen etenemisestä. Näihin perustietoihin sisältyy esimerkiksi työaseman toimialuenimi, asennuksen eteneminen prosenteissa ja asennukseen liittyviä aikatietoja. Asennusta tarkkailevan palvelun dataa voidaan lukea Powershellin avulla ja tämän datan avulla voidaan käynnistää työasemalle tilattujen ohjelmistojen asennuksen toteuttava prosessi. [52]

PS-skriptin toiminnallisuutta laajennettiin ensiksi tarkkailemaan MDT:n työasemien asennuksia seuraavaa prosessia, joka vaati vain muutamia uusia rivejä PS-skriptiin. Tämän jälkeen PS-skriptiin kehitettiin Active Directoryä käsitteleviä funktioita, joiden avulla pystytään tarkistamaan onko työasema liitetty toimialueeseen sekä lisäämään työasemia Active Directoryn ohjelmistoryhmiin. Powershellin Active Directory -moduulin toiminnallisuus on hyvin dokumentoitu ja pitkälti tämän kattavan dokumentaation ansiosta tarvittavat funktiot onnistuttiin kehittämään yllättävän lyhyessä ajassa.

Asennuksessa olevia työasemia tarkkaileva PS-skripti toimii seuraavanlaisesti:

1. Tarkista asennuksessa olevat työasemat MDT:n seurantapalvelusta
2. Tarkista, onko työasema liitetty toimialueeseen
3. Nouda työaseman tilaustietue ServiceNow'sta
4. Lue asennettavat ohjelmistot laitetilaustietueelta
5. Lisää työasema laitetilaustietueen ohjelmistoja vastaaviin Active Directory - ohjelmistoryhmiin

Käyttöjärjestelmien asennusta tarkkailevaa PS-skriptiä päätettiin suorittaa MDT-palvelimella viiden minuutin välein ajastettuna tehtävänä. Mikäli tarkkailuskriptiä olisi haluttu suorittaa jatkuvatoimisesti, olisi sitä varten täytynyt kehittää erillinen PS-skripti tai järjestelmäprosessi, joka olisi tarkkaillut asennuksia tarkkailevan PS-skriptin tilaa mahdollisten kaatumisten tai jäätymisten varalta. Työasemien asennusvaiheen tarkistaminen ei myöskään ollut aikakriittinen operaatio, sillä työaseman toimialueeseen liittämisen ja asennuksen valmistumisen välillä on yli kymmenen minuutin viive ja tarkkailuskriptin suoritus kestää yksittäisen työaseman kohdalla muutamia sekunteja. Mikäli yhtäaikaisten asennusten määrä kasvaa satoihin rinnakkain asennattaviin työasemiin, Active Directory ohjelmistoryhmäliitoksiin voi syntyä merkittäviä viiveitä, mutta tällaisessa skaalassa on syytä pohtia koko laitetilaus- ja asennusinfrastruktuurin uudelleen rakentamista.

#### **4.4.6 Active Directory – LANDESK Management Suite -integraatio**

Tilaajayrityksen kaikissa käyttöjärjestelmän asennussekvensseissä asennetaan työasemalle Landeskin agenttiohjelmisto, jonka avulla työasemaa pystytään hallinnoimaan etänä Landeskin Core-palvelinohjelmiston kautta. Tilaajayrityksessä tätä agenttia käytetään pääsääntöisesti kahteen tehtävään: työasemien tilaa kuvaavien tietojen keräämiseen ja työasemille asennettujen ohjelmistojen ylläpitämiseen. Agentin avulla voidaan myös suorittaa ajastettuja tehtäviä työasemilla ja näiden tehtävien avulla voidaan esimerkiksi asentaa ohjelmistoja. Landeskin ajastettuihin tehtäviin pystytään ohjelmoimaan konditionaalista logiikkaa ja tämä mahdollistaa tilattujen ohjelmistojen asentamisen automaattisesti. [50]

Konstruktiota varten kehitetyt Landeskin tehtävät suoritetaan tunnin välein, joka on pienin mahdollinen tehtävälle asetettava suoritusväli, jotka tarkistavat onko tehtävän mukainen ohjelmisto asennettu koneelle ja mikäli ohjelmisto puuttuu koneelta, tarkistetaan kuuluuko kyseinen työasema ohjelmiston mukaiseen Active Directory ryhmään. Jos koneelle ei ole asennettu tehtävän mukaista ohjelmistoa, mutta se kuuluu ohjelmistoryhmään, käynnistetään ohjelmiston asennus verkkoasemalta.

Jokaista tilaajayrityksen tukemaa ohjelmistoa varten täytyy luoda oma Landesk-tehtävä, joka olisi ollut työläs urakka. Kuten aiemmin on mainittu, Active Directoryyn luotiin projektin aikana ryhmät vain viidelle ohjelmistolle ja täten myös Landeskiin luotiin vain näitä ryhmiä vastaavat asennustehtävät, jotta järjestelmää päästiin testaamaan mahdollisimman nopeasti. Koska Landeskin ajastettuja tehtäviä pystytään suorittamaan minimissään tunnin välein tarkoitti se, että ohjelmistoasennukset käynnistyivät keskiarvoisesti puolen tunnin viiveellä siitä kun työasema oli liitetty ohjelmistoryhmään. Ylimääräinen viive ohjelmistojen asennuksessa oli epätoivottavaa, mutta koska asennukset pystyttiin suorittamaan agentin avulla turvallisesti ilman valvontaa, havaittu viive todettiin hyväksyttäväksi projektin kannalta.

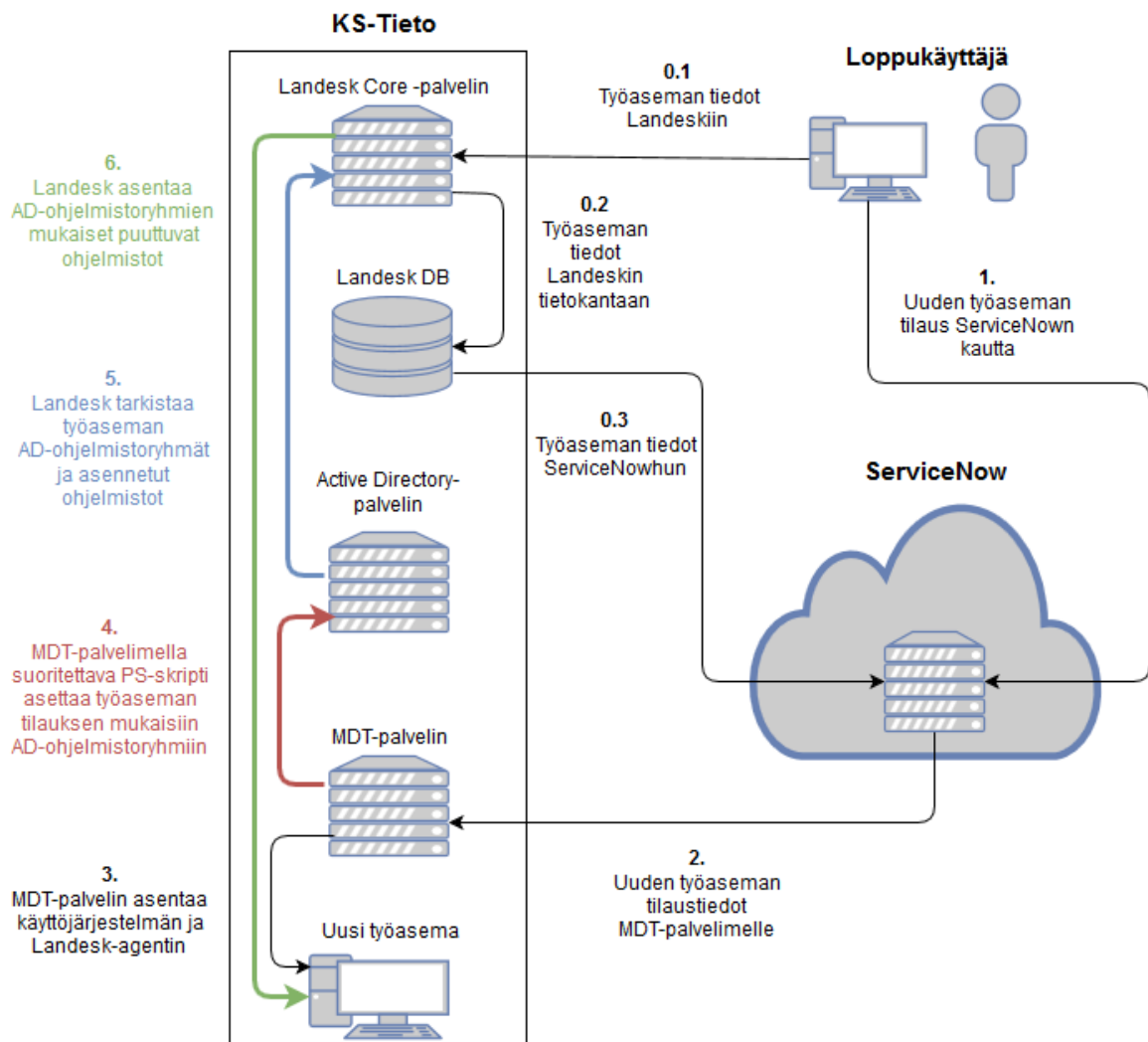
#### **4.4.7 Active Directory – ServiceNow -integraatio**

Työasemille asennetut ohjelmistot oli tarkoitus tarkistaa kertaluontoisesti Landeskin tietokannasta ja ohjelmistot, jotka asennettaisiin ilman ServiceNow-tilausta, eivät tällöin näkyisi automaattisesti ServiceNow'n tietokannassa. Koska asennettuja ja asennettavia ohjelmistoja tulnaisiin seuraamaan Active Directory -ryhmien avulla oli järkevää noutaa Active Directoryn ohjelmistoryhmätietueet ServiceNow'hun, etenkin kun ServiceNow'sta löytyy valmis työkalu LDAP-integraatiota varten, jota Active Directory tukee [54].

Tilaaajayrityksessä käytettiin jo ServiceNow'n LDAP-tietueita noutavaa ominaisuutta, mutta ominaisuuden implementaatio oli suoritettu kolmannen osapuolen toimesta, joten projektissa täytyi selvittää miten LDAP-tietueiden noutaminen suoritetaan. Olemassa olevasta LDAP-tietueita noutavasta toteutuksesta pystyttiin selvittämään yleiset parametrit LDAP-operaatiota varten ja varsinainen kehitystyö ei lopulta vaatinut muuta kuin oikean muotoisen LDAP-kyselyn luominen sekä kyselyn tulosten liittämisen oikeisiin ServiceNow'n tietueisiin.

#### **4.4.8 Toteutunut integraatiojärjestelmä**

Järjestelmä saatiin projektin kymmenennen viikon päätteeksi siihen pisteeseen, että ServiceNow'n kautta tehdyn työasematilauksen yhteydessä tilattujen ohjelmistojen asennukset pystyttiin suorittamaan automaattisesti tietyn rajoitteen sisällä. Tämä rajoite liittyi Active Directory -ohjelmistoryhmiin, joita luotiin projektin yhteydessä viidelle ohjelmistolle ja täten vain näiden ohjelmistojen asentaminen pystyttiin suorittamaan automaattisesti. Kuvasta 8 selviää toteutetun järjestelmän rakenne.



**Kuva 8.** Toteutunut järjestelmäintegraatio.

Käytössä olevien työasemien asennetut ohjelmistot haetaan Landeskistä ServiceNow’hun vain kerran, koska hakuun käytettävän SQL-kyselyn ylläpito on työlästä ja myöskin siksi että ServiceNow’n työasemille asennettujen ohjelmistojen tietokantataulu pystytään pitämään ajantasalla järjestelmän muiden komponenttien avulla.

MDT-palvelimella suoritettava PS-skripti tarkkailee työasemia, joissa on käyttöjärjestelmän asennus käynnissä, kun PS-skripti havaitsee, että asennuksessa oleva uusi työasema on liitetty toimialueeseen hakee se ServiceNow’sta tiedot työasemalle tilatuista ohjelmistoista. Tämän jälkeen PS-skripti liittää työaseman ohjelmistojen mukaisiin Active Directory -ryhmiin ja merkitsee työaseman käsitellyksi lokitiedostoon.

Kun käyttöjärjestelmän asennus on päättynyt ja työasema on käynnistynyt uudestaan, työasemalle käyttöjärjestelmäasennuksen yhteydessä asennettu Landesk-agentti ottaa yhteyden Landesk-keskuspalvelimeen. Landesk-keskuspalvelin antaa tietyin väliajoin agentille tehtäviä suoritettavaksi, joihin lukeutuu ohjelmistoasennuksista vastaavat tehtävät. Ohjelmistoasennustehtäviä suorittaessa Landesk-agentti tarkistaa, onko työasemalle asennettu tehtävän mukainen ohjelmisto ja kuuluuko työasema asennustehtävässä määritettyyn Active Directory -ryhmään. Mikäli työasemalle ei ole asennettu agentin suorittaman asennustehtävän mukaista ohjelmistoa, mutta työasema kuuluu asennustehtävän mukaiseen ohjelmistoryhmään, asentaa agentti kyseisen ohjelmiston.

Ydinprosessin ulkopuolisena vaiheena Active Directoryn ohjelmistoryhmätiedot noudetaan ServiceNow'hun ajastetuin välein sen varalta, että työasemille asennetaan ohjelmistoja ilman ServiceNow'n kautta tehtyjä tilauksia. Jos ohjelmistoja asennetaan ilman ServiceNow'n kautta tehtyä tilausta, täytyisi ServiceNow'n ohjelmistotaulua päivittää manuaalisesti, mikäli Active Directory -ohjelmistoryhmätietojen ei noudettaisi ServiceNow'hun säännöllisin väliajoin.

## 5 KONSTRUKTION ARVIOINTI

Projektissa kehitetyn järjestelmän eksplisiittisenä tavoitteena oli automatisoida uusille työasemille tilattujen ohjelmistojen asentaminen ja tämän tavoitteen täytyessä seurauksena olisi työasemien käyttöönottoon vaadittujen henkilötyötuntien vähentyminen. Voidaan siis todeta, että projektin todellisena tavoitteena oli työaseman käyttöönottokustannusten pienentäminen ja kehitetyn järjestelmän vaikutusta työaseman käyttöönoton kustannuksiin täytyi täten mitata jollakin tavalla.

Ohjelmistoasennusten automatisoinnin vaikutusta työaseman käyttöönoton kustannuksiin testattiin mittaamalla aikaa, joka kului asennusoperaatioon, johon sisältyy uuden työaseman käyttöjärjestelmän ja tilattujen ohjelmistojen asentaminen. Asennusoperaatioita suoritettiin projektissa kehitetyn järjestelmän avulla sekä ilman järjestelmää ja yksittäiseen asennusoperaatioon käytettyä aikaa mitattiin operaation kokonaiskeston sekä siihen käytettyjen henkilötyötuntien mukaan. Varsinainen asennusoperaatioon kuluva ajan mittaus suoritettiin sekuntikellon avulla manuaalisesti, joten mitatuissa ajoissa voi olla muutamien sekuntien virheitä. Mahdollisten mittausvirheiden merkitystä voidaan kuitenkin pitää vähäisenä asennusoperaatioon kuluva henkilötyömäärän kannalta, koska käytännössä operaation keston tulee vaikuttamaan myös sen suorittava työntekijä, jonka vaikutus on todennäköisesti paljon radikaalimpi operaation keston kannalta kuin mitä mahdolliset mittausvirheet vaikuttavat näiden testien tuloksiin.

Asennusoperaatioon kulunutta aikaa mitattiin siitä hetkestä lähtien kun työasema kytkettiin päälle. Työaseman päälle kytkemisen jälkeen Windows PE täytyi käynnistää manuaalisin toimenpitein ja odottaa, että Windows PE on latautunut käyttövalmiiksi. Kun Windows PE oli käyttövalmis sen valikoista valittiin työasemalle asennettavan käyttöjärjestelmän MDT-asennussekvenssi ja asennussekvenssin alkuvalikkoon syötettiin manuaalisesti ServiceNow'n tilausprosessin generoima työaseman toimialuenimi, jonka jälkeen käyttöjärjestelmän automaattinen asennussekvenssi käynnistyi. Asennussekvenssin käynnistyttyä kirjattiin ylös väliaika, koska asennussekvenssin käynnistymiseen vaaditut vaiheet vaativat yhtäjaksoista manuaalista työtä, jonka määrää testissä haluttiin mitata. Käyttöjärjestelmän asennuksen päätyttyä työasema käynnistyi uudelleen



käyttöjärjestelmään ja tässä välissä kirjattiin toinen väliaika, jotta ohjelmistojen asentamiseen käytetty aika saatiin määritettyä mahdollisimman tarkasti. Testin viimeisenä vaiheena työasemalle tilatut ohjelmistot asennettiin joko kehitetyn järjestelmän avulla automaattisesti tai manuaalisesti seuraamalla tilattujen ohjelmistojen listaa. Kun kaikki ohjelmistot olivat asentuneet työasemalle, ajanmittaus pysäytettiin ja koko asennusoperaatioon kulunut aika kirjattiin ylös.

Asennusoperaatioissa asennettaviksi ohjelmistoiksi valittiin viisi tilaajayrityksen työasemissa yleisesti käytettävää ohjelmistoa, jotka selviävät Taulukosta 3. Yleisyyden lisäksi asennettavaksi valikoituneille ohjelmistoille ei ollut muita valintakriteerejä, koska yksittäisen ohjelmiston asentamiseen kuluva aika ei pystytä etukäteen arvioimaan kovinkaan tarkasti, eikä ohjelmiston asentamiseen kuluvaan aikaan pystytä käytännössä vaikuttamaan lainkaan. Testeissä ei mitattu yksittäisten ohjelmistojen asennusaikoja, jotta tarkkailtavien muuttujien määrä pysyisi mahdollisimman pienenä.

**Taulukko 3.** Asennusoperaatioiden testaustiedot.

<b>Tietokoneen valmistaja</b>	<b>Tietokoneen malli</b>	
Fujitsu	Celsius W510	
<b>Käyttöjärjestelmän tuottaja</b>	<b>Käyttöjärjestelmä</b>	
Microsoft	Windows 7 SP1	
<b>Ohjelmiston tuottaja</b>	<b>Ohjelmiston nimi</b>	<b>Ohjelmiston tavukoko</b>
Adobe	FlashPlayer	20 Mt
Adobe	Reader DC	49 Mt
Oracle	Java	49 Mt
Paessler	PRTG Network Monitor	179 Mt
TeliaSonera	CallGuide	8 Mt

Mitattavia asennusoperaatioita suoritettiin viisi kertaa kehitetyn järjestelmän kanssa ja toiset viisi kertaa asentamalla tilatut ohjelmistot käsin ja mittaustulokset selviävät Taulukoista 4 ja 5. Mitattujen asennusoperaatioiden määrä jäi vähäiseksi, joka johtui hyvin pitkälti siitä, että järjestelmä saatiin mittauskelpoiseen tilaan vasta projektin toiseksi

viimeisen viikon päätteeksi ja itse mittausmenetelmä kehitettiin vasta projektin viimeisellä viikolla. Yksittäiseen asennusoperaatioon kului valmisteluineen aikaa tunnista kahteen tuntiin riippuen siitä, suoritettiinko operaatio kehitetyn järjestelmän avulla vai ilman järjestelmää ja projektiin varattu aika alkoi käydä vähiin. Kun nämä kymmenen asennusoperaatiota oli saatu suoritettua keskustelimme projektin käytännön ohjaajien kanssa mittauksen kattavuudesta ja totesimme että saatujen mittauksien avulla pystyttiin muodostamaan riittävän hyvä kuva projektissa kehitetyllä järjestelmällä saavutettavista hyödyistä sekä haitoista.

**Taulukko 4.** Manuaaliset ohjelmistoasennukset, ajat hh:mm:ss muodossa.

	<b>Käyttöjärjestelmä-asennuksen käynnistäminen</b>	<b>Käyttöjärjestelmä-asennuksen päättymisen</b>	<b>Ohjelmisto-asennusten päättymisen</b>	<b>Ohjelmisto-asennuksiin kulunut aika</b>
	00:03:43	00:25:20	00:31:59	00:06:39
	00:03:54	00:26:31	00:33:27	00:06:56
	00:03:57	00:27:12	00:34:27	00:07:15
	00:03:38	00:25:26	00:31:28	00:06:02
	00:03:30	00:25:41	00:30:26	00:05:45
<b>Keskiarvo</b>	<b>00:03:44</b>	<b>00:26:07</b>	<b>00:32:50</b>	<b>00:06:43</b>

**Taulukko 5.** Automaattiset ohjelmistoasennukset, ajat hh:mm:ss muodossa.

	<b>Käyttöjärjestelmä-asennuksen käynnistäminen</b>	<b>Käyttöjärjestelmä-asennuksen päättymisen</b>	<b>Ohjelmistoasennusten päättymisen</b>
	00:04:01	00:26:41	01:04:07
	00:03:57	00:26:15	00:43:11
	00:03:24	00:25:03	01:18:34
	00:03:42	00:25:57	00:47:30
	00:03:34	00:25:46	00:56:45
<b>Keskiarvo</b>	<b>00:03:46</b>	<b>00:25:56</b>	<b>00:58:01</b>

Asennusoperaatioiden alussa suoritettavat manuaaliset vaiheet vaativat keskimäärin 3 min 45 s ja nämä vaiheet täytyi suorittaa sekä kehityksen järjestelmän avulla, että ilman järjestelmää suoritetuissa asennusoperaatioissa.

Manuaalisesti suoritettavat asennusoperaatiot kestivät 32 min 50 s ja niissä käytettiin ohjelmistojen asentamiseen 6 min 43 s manuaalista työtä. Yksittäisen ohjelmiston asentaminen vaatii yleensä vain asennusohjelman käynnistämisen, mutta koska ohjelmistojen asentaminen kestää usein vain joitain minuutteja ei asennusten väliin jäävää aikaa pystytä käytännössä hyödyntämään muihin tehtäviin. Tästä johtuen kaikkien yksittäiselle työasemalle tilattujen ohjelmistojen asentamista käsitellään mittaustuloksissa yhtäjaksoisena työsuorituksena. Testiin valittujen ohjelmistojen asentamiseen vaaditun manuaalisen työn määrä jäi melko vähäiseksi. Tilaaajayritys toimittaa asiakkailleen vuosittain yli 500 uutta työasemaa, josta voidaan karkeasti johtaa että järjestelmän avulla pystytään vuosittain säästämään ainakin  $500 * 6 \text{ min } 47 \text{ s} = 56,5$  henkilötyötuntia. Kuten aiemmin todettiin eri ohjelmistojen asennusaikoja pystytään arvioimaan vain suurpiirteisesti ja projektin käytännön ohjaajien empiiristen havaintojen mukaan on erittäin todennäköistä, että tilattujen ohjelmistojen asentamiseen kuluu keskiarvoisesti enemmän aikaa kuin mitä mitatut arvot osoittavat. Manuaalisten asennusoperaatioiden mittaustuloksissa on huomion arvoista myös käyttöjärjestelmän asentamiseen kuluva aika 22 min 16 s. Koska käyttöjärjestelmän asentamiseen kuluva aika on suhteellisen lyhyt, seuraa siitä, että käyttöjärjestelmän asentamisen aikana ei välttämättä ehdi suorittamaan muita työtehtäviä, joka vaikuttaa työntekijän tehokkuuteen negatiivisesti. Mikäli käyttöjärjestelmän asentamiseen kuluva aika mukaanluetaan asennukseen käytettävään työaikaan, muuttuu vuositasolla säästettävien työtuntien määrä 242,5 tuntiin, joka on jo puolitoista henkilötyökuukautta.

Järjestelmän avulla suoritettuihin asennusoperaatioihin kulunut mitattu aika 58 min 1 s, on valitettavan epätarkka operaatioiden kokonaiskeston suhteen mitatun suureen korkean varianssin vuoksi. Asennusoperaatioiden kokonaiskeston suuri varianssi johtuu Landeskin keskuspalvelimen ohjaamien ajastettujen tehtävien pienimmästä mahdollisesta suoritusvälistä, joka on yksi tunti. Tunnin pituisesta tehtävien ajastusvälistä seuraa, että järjestelmän avulla suoritettavat asennusoperaatiot kestävät keskiarvoisesti puoli tuntia

pidempään kuin manuaalisesti suoritettavat operaatiot. Tarkasteltaessa järjestelmän tuottaman viiveen vaikutusta työasemien käyttöönottoprosessiin, saattaa olla järkevintä tarkastella järjestelmällä suoritettavia asennusoperaatioita huonoimman tapauksen mukaan, jolloin manuaalisten asennusoperaatioiden kokonaiskestoihin voitaisiin yksinkertaisesti lisätä yksi tunti operaatioiden kokonaiskeston arvioimista varten.

## 6 POHDINTA JA JOHTOPÄÄTÖKSET

Tilaaajayrityksen ohjelmistoasennusten automatisointi käyttöjärjestelmän asennuksen yhteydessä oli tämän diplomityön aloituspisteenä, eli työssä tuotettava konstruktiio oli tiedossa jo ennen hypoteesin kehitystä. Tästä johtuen työssä kehitetty ja testattu hypoteesi on melko suppea ja yksiulotteinen. Työn tutkimusongelmaa voidaankin pitää pikemminkin käytännöllisenä kuin tieteellisenä ja tämän vuoksi kirjallisen työn keskipisteenä onkin integraatioprojektin toteutuksen kuvaus. Projektin kuvausta itsessään on täten aiheellista käsitellä tapaustutkimuksellisenä raakadatana, jota tutkimalla on mahdollista havaita COTS-ohjelmistointegraatioiden suunnitteluun ja toteutukseen vaikuttavia tekijöitä. Toisaalta konstruktion testauksesta saadut mittaustulokset antavat viitteitä ohjelmistoasennusten automatisoinnin kannattavuudesta, mutta konstruktion prototyypiluonteesta sekä mittausten suppeudesta johtuen projektin tuloksena syntyneen järjestelmän takaisinmaksuajan arviointi on ongelmallista.

Konstruktion kehitystä kuvaavassa kappaleessa tuodaan esiin useita ongelmia ja konsepteja, jotka vaikuttivat konstruktion toteutukseen yleensä negatiivisesti. Suuri osa toteutukseen hidastavasti vaikuttaneista tekijöistä olivat odottamattomia ja tekijöiden yllätyksellisyys johtui pääsääntöisesti virheellisistä oletuksista projektin suunnitteluvaiheessa. Virheelliset oletukset puolestaan johtuivat liian suuripiirteisestä projektin suunnittelusta.

Työn konstruktion kehitys jaettiin sitä suunniteltaessa vaiheisiin, mutta alustava suunnitelma sisälsi useita oletuksia ohjelmistojen välisten integraatioiden suhteen. Esimerkiksi LANDESK Management Suiten ja ServiceNow'n välinen integraatio onnistuikin hyvin pitkälti oletusten mukaisesti, mutta ServiceNow'n ja Microsoft Deployment Toolin välisen integraation kohdalla oletukset osoittautuivat täysin vääriksi. Suuripiirteisen projektin toteutussuunnitelman lisäksi konstruktion kehittäjällä ei varsinaisesti ollut aiempaa kokemusta vastaavanlaisista projekteista, eikä kehittäjä edes ollut käyttänyt integraation kohteena olleita ohjelmistoja. Näistä seikoista johtuen, oli selvää jo ennen integraatioprojektin aloittamista, että projektin aikana tulisi vastaan ongelmatilanteita, jotka täytyisi ratkaista ad-hoc -tyylisesti.

Integraatioprojektin läpiviemiseen varattu aika oli jälkikäteen tarkasteltuna erittäin lyhyt ja konstruktion valmistuminen edes prototyyppiasteelle yhdessätoista viikossa oli jonkinasteinen saavutus. Projektille varattu aika oli lyhyt, koska oletuksena oli että ohjelmistojen väliset integraatiot olisivat onnistuneet ohjelmistoja konfiguroimalla, ohjelmistojen liitännäisiä hyödyntämällä ja mahdollisesti projektissa olisi täytynyt tuottaa joitakin lyhyitä skriptejä. ServiceNown ja MDT:n välinen integraatio täytyi kuitenkin toteuttaa pelkkien PowerShell-skriptien avulla, joita tuotettiin yli tuhannen rivin verran, ja tämä vaikutti projektin aikatauluun merkittävästi. Työssä tuotettu konstruktio saatiin kehitettyä prototyyppiasteelle noin 350:ssä työtunnissa ja konstruktion kaikkien osien kehittäminen ja testaaminen tuotantokäyttöön vaatisi vielä vähintään 150 työtuntia lisää.

## **6.1 Havainnot COTS-ohjelmistointegraatioiden suunnittelusta**

Tässä luvussa toistuvana teemana on oletuksien vaikutus COTS-ohjelmistointegraatioiden käytännön toteutukseen ja tehokas keino oletuksien välttämiseksi on perusteellisen suunnittelu- ja selvitystyön tekeminen. Projektin aikana sekä sen jälkeen tehtyjä havainnot hyödyntämällä kehitettiin Taulukon 6 mukainen tarkistuslista, jonka tarkoituksena on helpottaa vastaavanlaisten integraatioprojektien kokonaisvaltaista suunnitteluprosessia. Taulukon 6 sisältöä käsitellään tarkemmin taulukon jälkeisissä alaluvuissa.

**Taulukko 6.** Tarkistuslista COTS-ohjelmistointegraatioiden suunnittelua varten.

<b>Selvitä</b>	<b>Huomioi</b>
Integroitavien ohjelmistojen määrä	<i>Mitä useampia ohjelmistoja, sitä enemmän projekti vaatii kehitystä ja testausta.</i>
Integraatioiden suunnat	<i>Mihin suuntaan tietoja siirretään? <math>A \rightarrow B</math> integraatio ei ole yhtä kuin <math>B \rightarrow A</math> integraatio.</i>
Ohjelmistojen rajapinnat ja rajapintojen käyttäminen	<i>Onko ohjelmistoilla rajapinnat ja miten ne on dokumentoitu? Vaatiiko integraatio tiedostojen manipulaatiota ja millaisessa formaatissa tiedostot ovat? Onko integraatiot toteutettavissa valmiiden työkalujen avulla? Vaatiiko integraatio ohjelmointia?</i>
Tietojen käsittely	<i>Staattisesti muotoiltuja tietoja helppo käsitellä. Monimuotoisten tietojen käsittely voi vaatia monimutkaisempaa logiikkaa. Tarkista ohjelmoinnin tarve.</i>
Tietoturva	<i>Tarvitseeko siirrettäviä tietoja erikseen suojata? Vaikuttavatko yrityksen tietoturvakäytännöt käytettäviin ratkaisuihin?</i>
Ylläpito	<i>Kuinka paljon integraatio vaatii aktiivista ylläpitoa? Miten mahdolliset muutokset rajapinnoissa vaikuttavat järjestelmään? Liittyykö tietojen käsittelyyn ylläpitoa vaativia artefakteja?</i>
Ohjelmistojen korvaaminen	<i>Voidaanko integraatio, tai osaintegraatioita välttää käyttämällä vaihtoehtoisia ohjelmistoja?</i>

### **6.1.1 Integroitavien ohjelmistojen määrä**

Integroitavien ohjelmistojen määrä vaikuttaa suoraan integraatioprojektin kehitystä ja testausta varten varattavaan aikaan sekä kehityskustannuksiin. Integroitavien ohjelmistojen määrästä ei kuitenkaan pystytä tarkasti arvioimaan projektin kestoa tai sen kustannuksia, koska useat muut listassa esiintyvät tekijät vaikuttavat integroitavan järjestelmän kehittämiseen. Selvittämällä mitkä ohjelmistot ovat integraation kohteena, projekti on mahdollista jakaa selkeisiin vaiheisiin kuten tässä työssä kuvatussa projektissa tehtiin.

### **6.1.2 Integraatioiden suunnat**

Integraation suunta vaikuttaa ohjelmistoparin integraation toteutukseen merkittävästi. On mahdollista, että tiedon välittäminen ohjelmistosta A ohjelmistoon B onnistuu ohjelmistoon B rakennettujen valmiiden työkalujen tai liitännäisten avulla, mutta tiedon välittäminen ohjelmistosta B ohjelmistoon A vaatii ratkaisun toteuttamista ohjelmoimalla. Esimerkiksi työn konstruktiota kehitettäessä ilmeni että tietojen noutaminen useista kolmannen osapuolen ohjelmistoista ServiceNow'hun on toteutettavissa ServiceNow'n sisäänrakennettujen työkalujen avulla, mutta tietojen noutaminen ServiceNow'sta kolmannen osapuolen ohjelmistoon vaatii lähes poikkeuksetta erikseen ohjelmoitavan ratkaisun, jonka avulla ServiceNow'n ohjelmointirajapintaa hyödynnetään.

### **6.1.3 Ohjelmistojen rajapinnat ja rajapintojen käyttäminen**

Ohjelmistojen rajapintojen tutkiminen ja miten niitä hyödynnetään on integraation teknisen toteutuksen kannalta selvitystyön keskeisin vaihe. Onko ohjelmistolla ylipäätään rajapinta, jonka avulla sen tietoja voidaan manipuloida? Jos ohjelmistolla on rajapinta, niin onko rajapinnan ominaisuudet dokumentoitu riittävän hyvin, jotta integraatio pystytään toteuttamaan ennustettavassa ajassa? Mikäli rajapintaa ei ole, niin onko mahdollista manipuloida ohjelmiston käyttämiä tiedostoja jotka sisältävät haluttuja tietoja? Jos tiedostojen manipuloiminen on mahdollista, niin onko tiedosto jossakin yleisen standardin mukaisessa formaatissa, vai täytyykö tiedoston rakenne mahdollisesti selvittää takaisinmallinnuksen avulla. Edellämainitut ominaisuudet ovat integraation passiivisen osapuolen, eli ohjelmiston jonka tietoja pyritään manipuloimaan, ominaisuuksia ja tietojen



manipulaatiolla tarkoitetaan tässä yhteydessä tietojen noutamista ohjelmistosta, tietojen syöttämistä ohjelmistoon tai ohjelmiston tietojen muokkaamista. Integraation aktiivisessa osapuolella, eli ohjelmistossa jonka avulla tietoja pyritään manipuloimaan, täytyy myös olla mekanismeja manipulaation suorittamiseksi. Mekanismeilla puolestaan tarkoitetaan joko ohjelmiston sisältämiä valmiita työkaluja, joiden avulla integraatio voidaan toteuttaa, tai tukea jollekin ohjelmointikielelle, jotta integraatio voidaan toteuttaa ohjelmoimalla räätälöity ratkaisu.

Kaikilla ohjelmistoilla, jotka toimivat työn konstruktiossa passiivisinä osapuolina, on hyvin dokumentoidut rajapinnat ja ilman rajapintoja konstruktio olisi tuskin valmistunut edes prototyyppiasteelle aikataulussa. Rajapintojen tai niiden puuttumisen vaikutusta integraation tulee kuitenkin arvioida tapauskohtaisesti. Rakenteeltaan yksinkertaisen tiedoston manipulointi voi olla joissakin tapauksissa helpompaa kuin rajapinnan käyttäminen, mutta todennäköisesti ohjelmiston rajapinnan hyödyntäminen, etenkin jos sen dokumentaatio on selkeää ja kattavaa, on lähes poikkeuksetta paras vaihtoehto integraation toteuttamiseen.

Pääosa ohjelmistoparin integraation kehitystyöstä kohdistuu integraation aktiiviseen ohjelmistoon. Olennaista on selvittää onko integraatio toteutettavissa aktiivisen ohjelmiston sisäänrakennettujen työkalujen tai liitännäisten avulla ja vaatiiko integraatio ohjelmointia.

Aktiivisen ohjelmiston työkaluilla toteutettavan integraation eduiksi voidaan laskea, että integraation toteutus on suoraviivaista ja työkalut itsessään ovat testattuja ohjelmistotuottajan puolesta. Haittapuolena valmiiden työkalujen käyttämisessä on se, että niiden ominaisuuksiin ei juurikaan voi vaikuttaa ja tämän takia on varmistettava, että ominaisuudet täyttävät integraation vaatimukset. Ohjelmoitavan ratkaisun etuna on sen joustavuus, mutta haittapuolena on, että ratkaisun kehittäminen on usein hitaampaa kuin valmiiden työkalujen avulla ja ratkaisun testaaminen vaatii myös aikaa ennen kuin se voidaan ottaa käyttöön.

#### **6.1.4 Tietojen käsittely**

Ohjelmistointegraation tavoitteena on mahdollistaa tietojen siirtäminen ohjelmistojen välillä, ja jotta ohjelmisto pystyy hyödyntämään sille välitettyjä tietoja, on tiedot prosessoitava ohjelmistolle sopivaan muotoon. Staattisten tietojen, joiden sisältämä informaatio on aina samanlaista, prosessointi on melko triviaali toimenpide, jota varten pystytään kehittämään ratkaisu yleensä suhteellisen lyhyessä ja ennakoitavassa ajassa. Sisällöltään muuttuvien tietojen käsitteleminen vaatii monimutkaisempaa logiikkaa, joka usein tarkoittaa räätälöidyn ratkaisun kehittämistä ohjelmoimalla ja ratkaisun kehitysjan arvioiminen on haastavampaa kuin staattisen tiedon käsittelyn suhteen.

#### **6.1.5 Tietoturva**

Tietoturvallisuuden vaikutukset integraation toteuttamiseen eivät välttämättä ole ilmeisiä suunnitteluvaiheessa. Ohjelmistojen rajapintojen käyttäminen vaatii usein jonkinlaista autentikointia ennen tietojen välittämistä ja rajapinnat siten estävät tietojen päätyksen ei toivotuille tahoille. Ohjelmoimalla tuotettujen ratkaisujen kohdalla tietoturvaan tulee kiinnittää erityistä huomiota, koska esimerkiksi yrityksen tietoturvakäytännöt voivat vaikuttaa integraatiossa käytettäviin teknisiin ratkaisuihin. Esimerkkinä tilaajayrityksen tietoturvakäytännöt vaikuttivat työn konstruktion ohjelmien asennuksen yhteydessä, koska PS-skriptien etäsuorittaminen ei ollut sallittua ja alunperin suunniteltu ratkaisu tilattujen ohjelmistojen asentamiseen jouduttiin lopulta hylkäämään. Tiedonsiirtojen autentikointi, siirtojen salaus ja tietoturvakäytäntöjen vaikutukset tuleekin varmistaa etukäteen, jotta tietoturva ei synnyttäisi odottamattomia ongelmia integraation kehitysprosessin aikana.

#### **6.1.6 Ylläpito**

Integraation tuloksena syntyvän järjestelmän ylläpitäminen voi muodostua merkittäväksi kulueraiksi järjestelmän elinkaaren aikana ja ylläpidon tarvetta kannattaakin arvioida etukäteen. Integraation kohteena olevien ohjelmistojen muutoslokeja tarkastelemalla on mahdollista arvioida integraatorajapintojen ja niitä käyttävien mekanismien vakautta muutoksien suhteen. Rajapintojen ja niiden hyödyntämismekanismien arvioidun vakauden perusteella voidaan puolestaan arvioida ylläpidon tarvetta niiden suhteen pidemmällä

aikavälillä. Välitettävien tietojen käsittelemiseen tulee todennäköisesti liittymään ylläpidettäviä artefakteja ja tällaisten artefaktien määrän minimoiminen huolellisen suunnittelun avulla on mahdollista vaikuttaa kehitettävän järjestelmän ylläpitotarpeisiin.

### **6.1.7 Ohjelmistojen korvaaminen**

Integraatiota suunniteltaessa on syytä myös selvittää, onko ohjelmistojen vaatima integraatio vältettävissä vaihtoehtoisia ohjelmistoja käyttämällä, etenkin jos järjestelmä vaatii useamman kuin kahden ohjelmiston välistä integraatiota. Mikäli tilaajayrityksessä olisi käytetty Microsoft System Center Configuration Manageria työasemien hallinnoimiseen LANDESK Management Suiten sijasta, olisi projekti ollut todennäköisesti paljon yksinkertaisempi toteuttaa. SCCM:n avulla on muun muassa mahdollista pitää kirjaa työasemien ohjelmistoista, asentaa ohjelmistoja työasemille sekä käynnistää ja ohjata käyttöjärjestelmien asennuksia [55]. Näiden ominaisuuksien myötä tarvittujen integraatioiden määrä olisi laskenut viiden yksisuuntaisen integraation sijasta yhteen kaksisuuntaiseen integraatioon, koska SCCM:n avulla olisi voitu suorittaa kaikki ohjelmistoasennusten vaatimat operaatiot lukuunottamatta työaseman ja sen ohjelmistojen tilaamista. Työaseman ja sen ohjelmistojen tilaaminen olisi täytynyt tässäkin tapauksessa toteuttaa ServiceNow'n avulla ja tietojen noutaminen ServiceNow'sta SCCM:ään olisi täytynyt toteuttaa ServiceNow'n verkko-ohjelmointirajapintoja hyödyntämällä. Toinen vaihtoehtoinen toteutustapa olisi ollut LANDESKin tarjoaman Service Desk -pilvipalvelun käyttäminen ServiceNow'n sijasta. LANDESK Management Suiten ja LANDESK Service Deskin välinen integraatio on hyvin dokumentoitu, ja tällainen integraatio mahdollistaisi asiakkaiden tilaamien ohjelmistojen asentamisen automaattisesti milloin tahansa. Näitä mahdollisia käytettävien ohjelmistojen vaihdoksia mietittäessä on kuitenkin huomioitava niistä seuraavia muita vaikutuksia. Mikäli LANDESK Management Suite ja MDT korvattaisiin SCCM:llä täytyisi huomioida vaihdoksesta syntyvät integraatiokustannukset sekä uuden työkalun käyttöön liittyvien koulutusten kustannukset. Jos taas ServiceNow vaihdettaisiin LANDESK Service Deskiin integraatio- ja koulutuskustannusten lisäksi tilaajayrityksen asiakkaiden perehdyttäminen uuden palvelualustan käyttämiseen, sekä ServiceNow'hun integroitujen ohjelmistojen uudelleen implementointi uuteen alustaan, jotka voivat muodostua erittäin merkittäviksi kustannuseriksi.

### **6.1.8 Integraation kustannusten ja kannattavuuden arvioiminen**

Ottamalla huomioon Taulukossa 6 esiintuodut tekijät on mahdollista muodostaa selkeä kokonaiskuva suunniteltavasta integraatioprojektista ja integraation osa-alueista. Taulukon 6 mukaisesti tehtävän selvityksen avulla voidaan arvioida integraatioprojektin kehityskustannuksia suhteellisen tarkasti, mutta pelkkien kehityskustannusten avulla ei kuitenkaan pystytä arvioimaan integraation kannattavuutta. Integraation tavoitteista riippuen, integraation kannattavuuden arvioiminen voikin olla erittäin haastavaa.

Automatisoitaessa toimintoja ohjelmistointegraation avulla on lähes itsestään selvää, että toimintojen yksikkökustannukset laskevat, koska niiden suorittamiseen ei tällöin vaadita henkilötyötunteja. Toimintojen automatisoinnin kannattavuutta arvioitaessa tulisi mitata kuinka usein automatisoitavia toimintoja suoritetaan ja kuinka paljon toimintojen suorittaminen vaatii resursseja. Esimerkiksi tässä työssä tuotetulla konstruktiolla saavutettavia kustannushyötyjä olisi pystytty arvioimaan melko tarkasti selvittämällä kuinka usein mitäkin ohjelmistoa tilataan työaseman tilauksen yhteydessä, sekä mittaamalla jokaisen yksittäisen ohjelmiston asentamiseen vaadittu työmäärä. Kun kaikkien ohjelmistojen keskiarvoiset asennusajat tiedettäisiin ja ohjelmistojen tilausyleisyydet olisivat selvillä, niin näiden tietojen avulla voitaisiin laskea keskiarvo asennusoperaation vaatimasta työmäärästä. Tämän havainnon mukainen tutkimus olisikin ollut hyvä suorittaa ennen varsinaisen projektin aloittamista, koska sen avulla olisi saatu hyvä arvio varsinaisen integraatioprojektin kannattavuudesta.

Mikäli integraation tavoitteena on mahdollistaa jonkin uuden toiminnon suorittaminen, kuten esimerkiksi uudentyyppisen tiedon kerääminen, kannattavuuden arviointi saattaa jäädä subjektiiviseksi. Uudentyyppisen toiminnon kohdalla luultavasti oletetaan, että uusi toiminto joko suoraan tehostaa siihen liittyviä muita toimintoja tai että uuden toiminnon avulla voidaan analysoida ja tehostaa muita toimintoja epäsuorasti. Tällaisessa tapauksessa haasteeksi muodostuu vaikutusten suuruusluokan arviointi, jonka paikkansapitävyyttä voi olla mahdotonta varmistaa ennen integraation toteuttamista.

## **6.2 Jatkotutkimus**

COTS-ohjelmistointegraatioiden kehityskustannusten ja etenkin kannattavuuden arviointi on tapauskohtaista, koska integraatioihin vaikuttavien tekijöiden määrä voi vaihdella paljon eri integraatioiden välillä. Potentiaalisia jatkotutkimuksen aiheita olisikin tässä kappaleessa esitetyn tarkistuslistan kehittäminen ja testaaminen sekä menetelmien kehittäminen ohjelmistointegraatioiden kannattavuuden systemaattista arviointia varten.

## 7 YHTEENVETO

Tämän työn tarkoituksena oli automatisoida Kaakkois-Suomen Tieto Oy:n suorittamat ohjelmistoasennukset uusien työasemien käyttöönoton yhteydessä. Työssä tuotetun konstruktion, eli useiden KS-Tiedon käyttämien ohjelmistojen keskinäisen integraation, tavoitteena oli vähentää työasemien käyttöönottoon käytettäviä henkilötyötunteja. Mittaustulosten mukaan työn tavoite täyttyi, joten työtä voi pitää siltä osin onnistuneena, mutta ero manuaalisiin ohjelmistoasennuksiin jäi yllättävän vähäiseksi tulosten kriittisimmän tulkinnan mukaan.

Työn keskipisteenä oli haastavaksi osoittautuneen COTS-ohjelmistointegraatioprojektin kuvaus. Projektin kuvauksen sekä johtopäätöksissä esitetyn ohjelmistointegraatioiden suunniteluvaihetta varten kehitetyn tarkistuslistan avulla pystytään toivottavasti varautumaan COTS-ohjelmistointegraatioissa mahdollisesti ilmeneviin ongelmatilanteisiin. Huomionarvoinen havainto työn tuloksia pohdittaessa oli miten tämän projektin kannattavuutta olisi pystytty arvioimaan tarkemmin ennen varsinaisten ohjelmistointegraatioiden kehittämistä analysoimalla asennettavien ohjelmistojen asennusaikoja. Ohjelmistointegraatioiden kannattavuutta tulee kuitenkin arvioida tapauskohtaisesti, koska integraatioiden tuloksena syntyvillä järjestelmillä tavoiteltavia hyötyjä on hankalaa yleistää. Menetelmien kehittäminen ohjelmistointegraatioiden kannattavuuden arvioimiseksi jäikin avoimeksi kysymykseksi, jonka tutkimisesta voisi syntyä merkittäviä käytännön hyötyjä.

Tuotetusta konstruktiosta oli suoraa hyötyä tilaajayritykselle, koska sen osia on sittemmin otettu onnistuneesti tuotantokäyttöön. Seuraavana vaiheena olisi jatkokehittää konstruktion kaikki osat tuotantovalmiiksi ja tarkkailla kuinka merkittäviksi konstruktion avulla saavutettavat hyödyt muodostuvat pidemmällä aikavälillä.

## LÄHTEET

1. Google, Google Scholar, Noudettu 2.9.2016, <https://scholar.google.fi/>
2. Association for Computing Machinery, ACM Digital Library, Noudettu 2.9.2016, <http://dl.acm.org/>
3. Institute of Electrical and Electronics Engineers, IEEE Xplore, Noudettu 2.9.2016, <http://ieeexplore.ieee.org/Xplore/home.jsp>
4. Google, Google hakukone, Noudettu 2.9.2016, <https://www.google.fi/>
5. Dearle, A. "Software deployment, past, present and future." 2007 Future of Software Engineering." IEEE Computer Society, 2007.
6. Tallman, O. H. "Project Gabriel: Automated software deployment in a large commercial network." Digital Technical Journal 7 (1995): 56-56.
7. Carzaniga, A., Fuggetta, A., Hall, R. S., Heimbinger, D., Van Der Hoek, A., & Wolf, A. L. "A characterization framework for software deployment technologies." COLORADO STATE UNIV FORT COLLINS DEPT OF COMPUTER SCIENCE, 1998.
8. Dolstra, E. "The purely functional software deployment model." Väitöskirja, Utrecht University, 2006.
9. Dolstra, E., De Jonge, M., & Visser, E. "Nix: A Safe and Policy-Free System for Software Deployment." Proceedings of LISA '04: Eighteenth Systems Administration Conference (2004): 79-92.
10. Juve, G., & Deelman, E. "Automating application deployment in infrastructure clouds." Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on. IEEE, 2011.
11. Hasselbring, W. "Information system integration." Communications of the ACM 43.6 (2000): 32-38.
12. Land, R., & Crnkovic, I. "Existing approaches to software integration—and a challenge for the future." Integration 40 (2004): 58-104.
13. Boehm, B. W., & Abts, C. "COTS integration: Plug and Pray?" Computer 32.1 (1999): 135-138

14. Vigder, M., Gentleman W. M., & Dean, J. "COTS Software Integration: State of the art." NRC-CNRC Report, National Research Council Canada, 1996.
15. Abts, C., Boehm B. W., & Clark, E. B. "COCOTS: A COTS software integration lifecycle cost model-model overview and preliminary data collection findings." Proceedings of the 11<sup>th</sup> ESCOM Conference (2000): 325-333.
16. Rosa, W., Packard, T., Krupanand, A., Bilbro, J. W., & Hodal M. M. "COTS integration and estimation for ERP." Journal of Systems and Software 86.2 (2013): 538-550.
17. Microsoft, TechNet Library, Noudettu 4.9.2016, <https://technet.microsoft.com/library>
18. ServiceNow, ServiceNow Product Documentation, Noudettu 4.9.2016, <http://wiki.servicenow.com>
19. Lukka K. Konstruktiivinen tutkimusote, Metodix, Noudettu 7.9.2016, <https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/>
20. Radatz, J., Geraci A., & Katki, F. "IEEE standard glossary of software engineering terminology." IEEE Std 610.12 (1990).
21. Bygstad, B., Nielsen, P. A., & Munkvold, B. E. "Four integration patterns: a socio-technical approach to integration in IS development projects." Information Systems Journal, 20.1 (2010), 53-80.
22. Microsoft, TechNet, SQL Server Language Reference, Transact-SQL Reference, Noudettu 29.10.2016, [https://technet.microsoft.com/en-us/library/ms189826\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms189826(v=sql.90).aspx)
23. Internet Engineering Taskforce, Request for Comments: 4511, Lightweight Directory Access Protocol (LDAP): The Protocol, Noudettu 29.10.2016, <https://tools.ietf.org/html/rfc4511#section-1>
24. World Wide Web Consortium, Extensible Markup Language, Noudettu 30.10.2016, <https://www.w3.org/XML/>
25. Yakimovich, D., Bieman, J. M., & Basili, V. R. "Software architecture classification for estimating the cost of COTS integration." Software Engineering, Proceedings of the 1999 International Conference (1999): 296-302. IEEE.
26. Park, J., & Sudha R. "Information systems interoperability: What lies beneath?" ACM Transactions on Information Systems (TOIS) 22.4 (2004): 595-632.



27. Kaakkois-Suomen Tieto Oy, Noudettu 10.8.2016, <http://www.kstieto.fi/>
28. ServiceNow, ServiceNow Platform datasheet, Noudettu 2.7.2016, <https://www.servicenow.com/content/dam/servicenow/documents/datasheets/ds-servicenow-platform.pdf>
29. ServiceNow, ServiceNow Product Documentation, List of Plugins, Noudettu 15.7.2016, [https://wiki.servicenow.com/?title=List\\_of\\_Plugins#gsc.tab=0](https://wiki.servicenow.com/?title=List_of_Plugins#gsc.tab=0)
30. LANDESK, Products, Management Suite, Noudettu 10.7.2016, <http://www.landesk.com/products/management-suite/>
31. Microsoft, Products, Windows Server 2012 R2 datasheet, Noudettu 4.8.2016, [http://download.microsoft.com/download/D/2/C/D2CDA5BA-E440-4A50-A418-5362291156C1/Windows\\_Server\\_2012\\_R2\\_Datasheet.pdf](http://download.microsoft.com/download/D/2/C/D2CDA5BA-E440-4A50-A418-5362291156C1/Windows_Server_2012_R2_Datasheet.pdf)
32. Microsoft, Developer Network, SQL Server, Features and Tools Overview (SQL Server 2008), Noudettu 4.8.2016, [https://msdn.microsoft.com/en-us/library/bb500397\(v=sql.105\).aspx](https://msdn.microsoft.com/en-us/library/bb500397(v=sql.105).aspx)
33. Microsoft, TechNet, Microsoft Deployment Toolkit (MDT), Using the Microsoft Deployment Toolkit, Noudettu , <https://technet.microsoft.com/en-us/library/dn759415.aspx>
34. Microsoft, TechNet, Microsoft Deployment Toolkit, MDT 2012 Update 1 Documentation, Using the Microsoft Deployment Toolkit, Noudettu 10.7.2016, <http://download.microsoft.com/download/b/3/a/b3a89fae-f7bf-4e7c-b208-223b991e9c30/Optional%20-%20MDT%202012%20Print-Ready%20Documentation.zip>
35. Microsoft, Developer Network, Directory Services, Active Directory Domain Services, Noudettu 11.7.2016, [https://msdn.microsoft.com/en-us/library/aa362244\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa362244(v=vs.85).aspx)
36. Microsoft, TechNet, Active Directory, Introduction to the Active Directory Schema, Noudettu 12.7.2016, <https://technet.microsoft.com/en-us/library/cc961756.aspx>
37. Microsoft, TechNet, Active Directory, Active Directory Domain Hierarchy, Noudettu 12.7.2016, <https://technet.microsoft.com/en-us/library/cc978009.aspx>

38. Microsoft, Developer Network, Directory Services, About Active Directory Domain Services, Noudettu 12.7.2016, [https://msdn.microsoft.com/en-us/library/aa772142\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa772142(v=vs.85).aspx)
39. Microsoft, TechNet, Active Directory, Active Directory Administration with Windows PowerShell, Noudettu 12.7.2016, <https://technet.microsoft.com/en-us/library/dd378937.aspx>
40. Microsoft, Developer Network, Directory Services, Lightweight Directory Access Protocol, Noudettu 12.7.2016, [https://msdn.microsoft.com/en-us/library/aa367008\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa367008(v=vs.85).aspx)
41. Microsoft, Developer Network, PowerShell, Noudettu 21.7.2016, <https://msdn.microsoft.com/en-us/powershell/mt173057.aspx>
42. Microsoft, TechNet Articles, Popular PowerShell Modules, Noudettu 22.7.2016, [https://social.technet.microsoft.com/wiki/contents/articles/4308\\_popular-powershell-modules.aspx](https://social.technet.microsoft.com/wiki/contents/articles/4308_popular-powershell-modules.aspx)
43. Microsoft, TechNet, Blogs, Hey, Scripting Guy! Blog, What Is PowerShell, Noudettu 22.7.2016, <https://blogs.technet.microsoft.com/heyscriptingguy/2015/01/02/what-is-powershell/>
44. ServiceNow, ServiceNow Product Documentation, MID Server, Noudettu 9.5.2016, [http://wiki.servicenow.com/index.php?title=MID\\_Server](http://wiki.servicenow.com/index.php?title=MID_Server)
45. ServiceNow, ServiceNow Product Documentation, Data Sources, JDBC Data Types, Noudettu 10.5.2016, [http://wiki.servicenow.com/index.php?title=Data\\_Sources#JDBC\\_Types](http://wiki.servicenow.com/index.php?title=Data_Sources#JDBC_Types)
46. ServiceNow, ServiceNow Product Documentation, Direct Web Services, Noudettu 13.5.2016 , [http://wiki.servicenow.com/index.php?title=Direct\\_Web\\_Services#gsc.tab=0](http://wiki.servicenow.com/index.php?title=Direct_Web_Services#gsc.tab=0)
47. MicroSoft, Developer Network, Windows PowerShell 5.0, Invoke-WebRequest, Noudettu 19.5.2016, <https://technet.microsoft.com/en-us/library/hh849901.aspx>
48. Microsoft, Developer Network, Security, Windows Data Protection, Noudettu 19.5.2016, <https://msdn.microsoft.com/en-us/library/ms995355.aspx>
49. LANDESK, White Papers, Accelerating Your Distributed Environment with LANDESK® Systems Management, Noudettu 20.5.2016,

<http://rs.landesk.com/whitepapers/accelerating-your-distributed-environment-LSI-1088.pdf>

50. LANDESK, LANDESK Help Center, Scheduled tasks help, Noudettu 20.5.2016, [https://help.landesk.com/topic/index/enu/ldms/9.5/content/windows/tasks\\_help.htm](https://help.landesk.com/topic/index/enu/ldms/9.5/content/windows/tasks_help.htm)
51. Microsoft, TechNet, Microsoft Deployment Toolkit (MDT), Microsoft Deployment Toolkit Samples Guide, Noudettu 3.6.2016, <https://technet.microsoft.com/en-us/library/dn781089.aspx>
52. Microsoft, TechNet, Blogs, Michael Niehaus' Windows and Office deployment ramblings, Troubleshooting MDT 2012 Monitoring, Noudettu 10.6., <https://blogs.technet.microsoft.com/mniehaus/2012/05/10/troubleshooting-mdt-2012-monitoring/>
53. Microsoft, TechNet, Windows Server, Active Directory Cmdlets in Windows PowerShell, Noudettu 18.6.2016, <https://technet.microsoft.com/en-us/library/ee617195.aspx>
54. ServiceNow, ServiceNow Product Documentation, LDAP Integration, Noudettu 18.6.2016, [http://wiki.servicenow.com/?title=LDAP\\_Integration#gsc.tab=0](http://wiki.servicenow.com/?title=LDAP_Integration#gsc.tab=0)
55. Microsoft, Cloud Platform, System Center Configuration Manager Datasheet, Noudettu 3.10.2016, [http://download.microsoft.com/download/5/D/B/5DBEBA38-8D5D-4119-B2E8-B8369B74BF43/system\\_center\\_configuration\\_manager\\_and\\_microsoft\\_intune\\_datasheet.pdf](http://download.microsoft.com/download/5/D/B/5DBEBA38-8D5D-4119-B2E8-B8369B74BF43/system_center_configuration_manager_and_microsoft_intune_datasheet.pdf)