



Juho Kähärä

Toiminnan- ja tuotannonohjausjärjestelmien asennuksen automatisointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

28.3.2022

Tiivistelmä

Tekijä:	Juho Kähärä
Otsikko:	Toiminnan- ja tuotannonohjausjärjestelmien asennuksen automatisointi
Sivumäärä:	33 sivua + 1 liite
Aika:	28.3.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Lehtori Vesa Ollikainen

Automaation tärkeys modernissa yhteiskunnassa kasvaa jatkuvasti, sillä automaatio vähentää itseään toistavien prosessien jatkuvaa manuaalista suorittamista, näihin prosesseihin vaadittua työtä voidaan täten hyödyntää toisaalla. Pitkälle hiottu automaatiojärjestelmä minimoi käyttäjältä vaadittavan työpanoksen ja täten vähentää prosessin alttiutta virheille.

Insinööriyön kohteena ovat ohjelmistoasennuksissa ja niiden automaatioissa kohdatavat haasteet sekä niiden ratkaisut.

Insinööriyön tavoitteena on tutustua ohjelmistoasennusten automatisoinnin toteutus-tapoihin, automatisaation mahdollisuuksiin ja potentiaalsiin haittoihin sekä kehittää toiminnan- ja tuotannonohjausjärjestelmien asennusta varten asennusprosessia automaatiovaivaksi työkalu.

Automaatiotyökalu suunniteltiin Windows-käyttöjärjestelmälle ja sen toteutuksessa käytettiin PowerShell-skriptejä.

Avainsanat: Automaatio, PowerShell, Ohjelmiston asennus

Abstract

Author: Juho Kähärä
Title: Automation of Installation of Manufacturing Execution and Enterprise Resource Planning Systems
Number of Pages: 33 pages + 1 appendix
Date: 28 March 2022

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Software Engineering
Supervisors: Vesa Ollikainen, Senior Lecturer

The importance of automation in modern society is ever-growing as it reduces the manual execution of repetitive tasks and allows redirecting the resources from these tasks to any other task. A highly developed automation system minimizes the amount of required user input and thus reduces the error susceptibility of the process.

The subjects of the thesis are the challenges and solutions involved in the installation of software and the automation of the software installation procedure.

The objective was to explore the different methods of implementation of automation, the opportunities, and potential drawbacks it may present, and to produce a tool for automating the installation of several manufacturing execution and enterprise resource planning systems.

The automation tool was designed for the Windows operating system and is composed of several PowerShell scripts.

Keywords: Automation, PowerShell, Software installation

Sisällys

Lyhenteet

1	Johdanto	1
2	Automaatio ohjelmistoympäristössä	2
2.1	Automaation vahvuudet	2
2.2	Automaation heikkoudet	2
2.3	Automoidun asennuksen toteutustavat	3
2.3.1	Komentorivi	3
2.3.2	Docker (kontittaja)	4
2.3.3	Package Manager (paketinhallintajärjestelmä)	4
2.4	Batch vai PowerShell?	5
2.5	Visuaalisella käyttöliittymällä varustetut ohjatut asennustoiminnot	6
2.6	CI/CD (Continuous Integration / Continuous Deployment tai Development)	7
3	Ongelman kuvaus	9
3.1	Työkalujen asennus	12
3.2	Oraclen ja Microsoftin tietokannat	12
3.3	Toiminnanohjausjärjestelmä	13
3.4	SSRS (SQL Server Reporting Services)	14
3.5	Tuotannonohjausjärjestelmät	15
3.6	Viimeistelyt	15
4	Projektin ympäristö	15
4.1	Testikäyttöjärjestelmäympäristö	15
4.2	PowerShell	16
4.3	Testiympäristö	17
5	Ohjelmistoasennuksen automaation toteutus	18
5.1	Palvelinasetukset	18
5.2	Työkalujen ja sovellusten asennuksen automaatio	18
5.3	Tuotannonohjausjärjestelmän asennuksen automaatio	21
5.4	SSRS	26
5.5	Oracle-tietokanta	26

5.6	Viimeistelyt	27
6	Lopputulos	27
6.1	Ohjelmistoasennuksissa esiintyvät huolimattomuusvirheet	28
6.2	Asennusten toimenpiteiden muistaminen	28
6.3	Ohjelmistoasennuksen monimutkaisuudesta aiheutuva tehottomuus	29
6.4	Automaatiotyökalun tehokkuuden analysointi kokonaisuutena	29
7	Yhteenveto	30
	Lähteet	32
	Liitteet	
	Liite 1: Asennusskriptejä kutsuva pääskripti	

Lyhenteet

- APEX: *Oracle Application Express*. Oraclen suunnittelema vähäisen koodin kehitysalusta Oraclen tietokantaa hyödyntäviä web-sovelluksia varten yritysympäristössä.
- CI/CD: *Continuous Integration / Continuous Deployment tai Development*. Ohjelmistopäivityksiin liittyvä prosessi, jossa automoidaan uusimman julkaisun testaus ja toimitus.
- CLI: *Command Line Interface*. Komentorivi tietokoneen ohjaamista varten.
- DB: *Database*. Tietokanta. Organisoitu kooste sähköisesti käsiteltävästä informaatiosta.
- ERP: *Enterprise Resource Planning*. Yrityksille suunnattu lukuisia eri toimintoja integroiva toiminnanohjausjärjestelmä.
- GUI: *Graphical User Interface*. Moderni visuaalisuuteen panostava käyttöliittymä, jota voidaan ohjata joko kursorilla tai näppäimistöllä.
- IDE: *Integrated Development Environment*. Ohjelmointiympäristö ohjelmien suunnittelua ja toteutusta varten.
- IP: *Internet Protocol*. Tietoliikennepakettien perille toimittamisesta huolehtiva verkkoprotokolla.
- MES: *Manufacturing Execution System*. Tehdasautomaatioon yhdistettävä tuotannonohjausjärjestelmä, jolla hallitaan tuotannon operatiivisia toimintoja sekä välitetään informaatiota toiminnanohjausjärjestelmille.

- MS-DOS: *Microsoft Disk Operating System*. Microsoftin kehittämä komentori-
vipohjaiseen käyttöliittymään perustuva tietokoneen levykäyttöjär-
jestelmä.
- MS-SQL: *Microsoft SQL Server*. Microsoftin tarjoama relaatiotietokantapalve-
lin.
- SA: *System Administrator*. Järjestelmänvalvoja.
- SSMS: *SQL Server Management Studio*. Microsoftin luoma Microsoft SQL-
palvelimen ylläpitoon tarkoitettu sovellus.
- SSRS: *SQL Server Reporting Services*. Microsoftin palvelin pohjainen
raportingenerointiohjelma.
- SQL: *Structured Query Language*. Relaatiotietokannan käsittelyyn kehi-
tetty standardoitu kyselykieli.
- TCP/IP: *Transmission Control Protocol / Internet Protocol*. TCP/IP on Inter-
netissä käytettävä kommunikaatioprotokollapino, joka tarjoaa kah-
den päätelaitteen välisen tiedonsiirtoyhteyden, pakettien järjestämi-
sen sekä lähettää mahdollisesti hukkuneet paketit uudelleen.
- TNS: *Transparent Network Substrate*. Toimii aliaksena Oraclen palvelin-
yhteysmerkkijonoon.

1 Johdanto

Automaatio on yksi ihmiskunnan tärkeimmistä tavoitteista sekä suuri osa modernia yhteiskuntaa. Mekaanisesti automaatiota on toteutettu jo vuosisatoja, mutta tietokoneiden yleistyttyä automaation mahdollisuudet ovat kasvaneet huomattavasti. Tässä insinööriyössä keskitytään automaatioon ohjelmistojen asentamisen, konfiguroinnin sekä päivittämisen kannalta.

Tämä insinööriyö on jaettu kahteen pääosioon, teoriaosuuteen sekä toteutukseen. Ensimmäisen osan (luvut 2 ja 3) tavoitteena on välittää lukijalle, kuinka automaatiota voitaisiin hyödyntää ohjelmistokokonaisuuksien asentamisessa etäpalvelimille. Aluksi syvennytään erilaisiin automaatiototeutuksiin ja siihen, mitä on tärkeitä ottaa huomioon työkalun suunnittelua varten. Tämän jälkeen aletaan pohtimaan konkreettisen toteutuksen skaalaa, tavoitetta ja toteutusta.

Toisen osan (luvut 4 ja 5) tavoitteena on esitellä lukijalle luvussa 3 kuvaillun ongelman ratkaisemisen vaatimat toimenpiteet.

Lopuksi vielä luvuissa 6 ja 7 analysoidaan insinööriyön onnistuneisuutta, kuinka hyvin se vertautuu alkuperäisiin suunnitelmiin, minkälaisia muutoksia työkalu koki kehitysvaiheessa ja pohditaan työkalun mahdollista jatkokehitystä.

Insinööriyön lopputuloksena on toiminnan- ja tuotannonohjausjärjestelmien asennustyökalu, jolla voidaan suorittaa vaihtelevan ohjelmistokokonaisuuden etäasennuksia sekä päivityksiä. Vaihtelevalla ohjelmistokokonaisuudella tarkoitetaan, että ohjelmistokokonaisuus koostuu useasta eri ohjelmasta, jotka ostetaan erikseen.

Asennusautomaatiotyökalu luodaan Elematic Oyj -nimisen yrityksen (1) käyttöön. Elematic Oyj on suomalainen betonielementtiteollisuuden tarpeisiin koneita, laitteita, elementtitehtaita ja ohjelmistoja valmistava ja toimittava yritys, jolla on yli 60 vuoden kokemus elementtikoneiden valmistuksesta.

Yrityksen tuotannosta noin 90 prosenttia on vientiä. Elematic Oyj kuuluu maailman johtaviin elementtikoneiden valmistajiin.

2 Automaatio ohjelmistoympäristössä

Tässä luvussa käsitellään tarkemmin automaation hyötyjä ja haittoja sekä käytötarkoituksia ja toteutuskeinoja ohjelmistoympäristössä.

2.1 Automaation vahvuudet

Automaatiolla pyritään vähentämään ihmisen huolimattomuudesta johtuvia virheitä ja tuottamaan tasaisempaa sekä usein myös korkeampaa laatua tehokkaammin. Vaikka automaatioprosessit vaativat kuitenkin ihmisen valvontaa toiminnan varmistamiseksi, on työtehtävä usein mielekkäämpää kuin toistuvan työn suorittaminen manuaalisesti päivästä toiseen.

Ohjelmistoasennukset ovat lähtökohdaltaan hyvin automatisoitavissa oleva prosessi, sillä tietokoneet pohjautuvat noudattamaan peräkkäisiä komentoja, joita käytännössä jokainen tietokoneella suoritettava toimenpide on.

Mitä useammin käyttöympäristöjä on yrityksellä asennettavana, sitä tärkeämpää prosessin automaatio on, sillä manuaaliset asennukset käyttävät huomattavasti enemmän työtunteja. Näin ollen tulee ajan myötä yritykselle kalliimmaksi kuin investoida prosessin automaatioon.

2.2 Automaation heikkoudet

Automaatiolla on myös omat heikkoutensa. Automaation kehittäminen sekä käyttöönotto vaatii suunnittelua sekä toteutusta. Tämä on hyvä ottaa huomioon prosessin automaatiota harkitessa. Mikäli prosessin automaatio vaatii enemmän vaivaa kuin sen toteuttaminen manuaalisesti, esimerkiksi vain kerran vuodessa suoritettava vartin toimenpide, ei se välttämättä ole sen arvoista.

Ohjelmistoasennusautomaation näkökulmasta ajatellen on pohdittava, onko valmiiksi jo mahdollisimman helppokäyttöisiksi luotuja ohjattuja asennustoimintoja tarpeellista automatisoida enempää.

Mikäli prosessin aikana tapahtuu virhe, voi virhe johtaa suurempiin menetyksiin kasvaneen tehokkuuden avustuksella ennen kuin virheellinen toiminto on saatu pysäytettyä. Tätä varten on hyvä lisätä tarkistuspisteitä, joissa prosessin suoritus keskeytetään operaatioiden välissä pyytäen käyttäjältä varmistusta jatkamisesta. Ohjelmistoja asennettaessa tämä pysäytys mahdollistaisi esimerkiksi komentoriville tulostettavan asennuslokin tarkistamisen potentiaalisten virheiden varalta, kun se ei etene tietokoneen suorittamalla nopeudella jatkuvasti eteenpäin.

Ohjelmistojen kannalta automaation suunnittelua heikentävät modernit, ihmiskäyttöön suunnitellut graafiset käyttöliittymät, ellei näille ole luotu hyvin dokumentoituja vaihtoehtoisia komentorivityökaluja.

2.3 Automoidun asennuksen toteutustavat

Automaatiota ohjelmistoympäristössä voi toteuttaa usealla eri keinolla. Seuraavaksi aion lyhyesti esitellä muutamia näistä eri keinoista automatisoida ohjelmistojen asennuksia.

2.3.1 Komentorivi

Perinteisin ohjelmistoautomaatio on komentorivillä (kuva 1) suoritettavat skriptit. Komentoriviskripteillä pystyy helposti toteuttamaan erilaisia operaatioita, kuten

purkamaan paketteja, kirjoittamaan tiedostoihin, siirtämään tiedostoja, mukautamaan esimerkiksi Windows-käyttöjärjestelmän asetuksia sekä ajamaan

```
C:\> Command Prompt  
Microsoft Windows [Version 10.0.19042.1645]  
(c) Microsoft Corporation. All rights reserved.  
C:\Windows\system32>_
```

sovelluksia.

Kuva 1. Komentorivi.

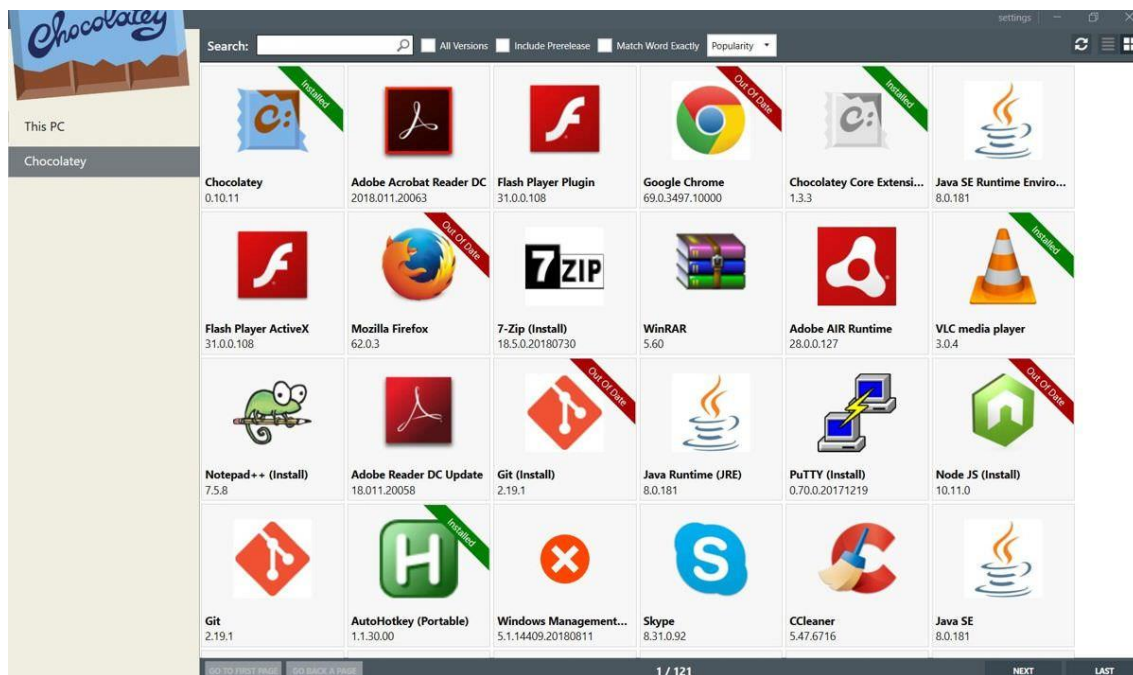
Etenkin Microsoftin .NET Framework (2) -ohjelmistokomponenttikirjastolla kehittämä seuraavan sukupolven komentotulkki PowerShell (3) mahdollistaa merkkipohjaisilla ohjelmilla (näistä puhutaan jatkossa skripteinä) käytännössä mitä vain, mitä visuaalisellakin käyttöliittymällä voi saavuttaa järjestelmää konfiguroitaessa.

2.3.2 Docker (kontittaja)

Dockerilla (4) voi toimittaa käyttöjärjestelmätasolla virtualisoituja ohjelmistoja säiliöiksi (container) kutsutuissa paketeissa. Tämä mahdollistaa ohjelmiston ajamisen erilaisten ympäristöjen välillä, sillä säiliö sisältää kaikki ohjelman vaatimat koodit, koodikirjastot, työkalut, asetukset ja niin edespäin.

2.3.3 Package Manager (paketinhallintajärjestelmä)

Paketinhallintajärjestelmä (kuva 2) on valikoima työkaluja, joilla voidaan automatisoida ohjelmistojen asennusta, päivitystä, konfigurointia sekä käyttöä. Paketinhallintajärjestelmät nopeuttavat ympäristön käyttöönottoa sekä varmistavat pakettien versionumeroiden olevan yhtenäisiä, mikäli tämä on tarpeellista.



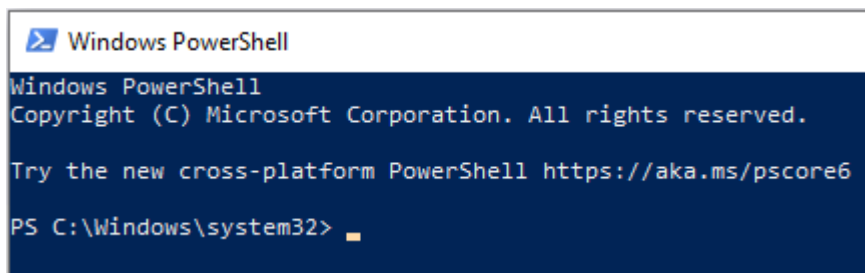
Kuva 2. Chocolatey-paketinhallintajärjestelmä Windowsille. (5.)

Kuvassa 2 esitellään Windowsille saatavilla oleva Chocolatey-paketinhallintajärjestelmän (6) visuaalinen käyttöliittymä.

2.4 Batch vai PowerShell?

Vuodesta 1987 lähtien on Windows-ympäristössä ollut käytössä cmd.exe-komentorivi (7), jolle voidaan kirjoittaa .bat-tiedostoja automatisoimaan erinäisiä operaatioita. Vaikkakin .bat-tiedostot ovat olleet kauan käytössä, se ei tarkoita, että ne olisivat kehittyneet sen kummemmin.

Tämän vuoksi nykypäivänä suosittelisin jokaiseen uuteen skriptiin mieluummin PowerShell-komentotulkkia (kuva 3), sillä kaikki batch-skripteillä suoritettavat toimenpiteet ovat vain osa PowerShellin kyvykkyydestä.

A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell". The terminal text includes: "Windows PowerShell", "Copyright (C) Microsoft Corporation. All rights reserved.", "Try the new cross-platform PowerShell https://aka.ms/pscore6", and the prompt "PS C:\Windows\system32>".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32>
```

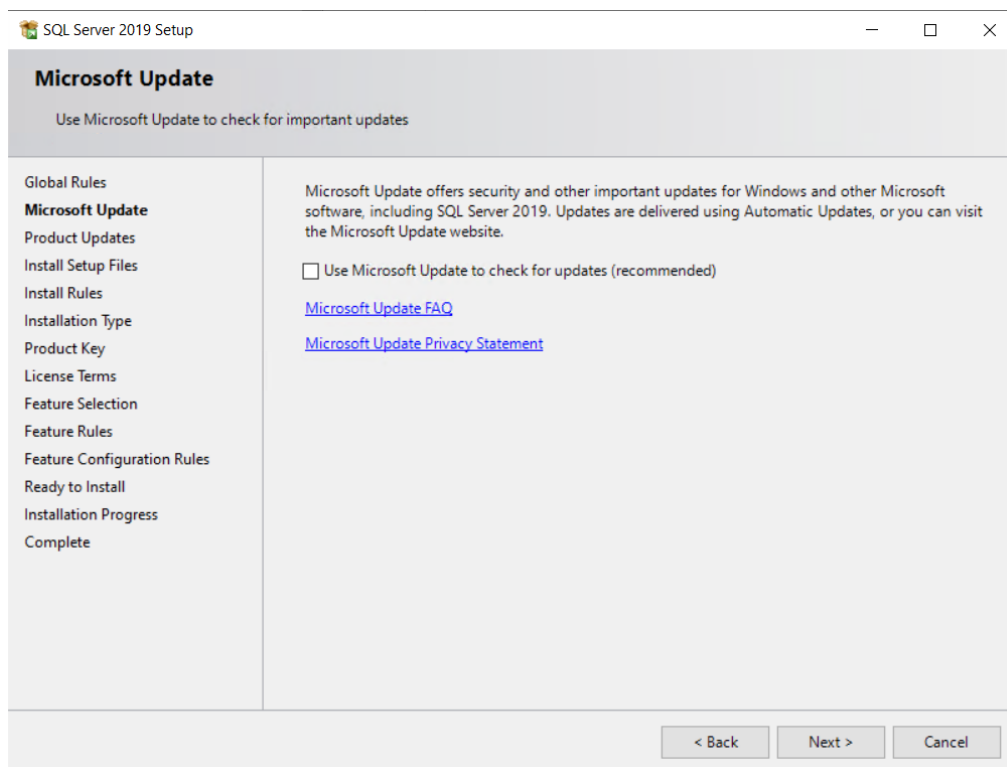
Kuva 3. Powershell-komentotulkki.

PowerShell on alun perin Windows-käyttöjärjestelmäympäristöä varten suunniteltu komentotulkki sekä skriptikieli, mutta se on nykyään saatavilla myös Unix-järjestelmille.

2.5 Visuaalisella käyttöliittymällä varustetut ohjatut asennustoiminnot

Edellisessä aliluvussa perehdyttiin komentorivillä suoritettavaan automaatioon. Seuraavaksi perehdymme sen niin kutsuttuun vastakohtaan eli visuaalisella käyttöliittymällä varustettuihin ohjattuihin asennustoimintoihin.

Käyttäjätasvällisyyttä huomioidessa tulisi automatisoituun asennustyökaluun liittää visuaalisella käyttöliittymällä varustettu ohjattu asennustoiminto (kuva 4), joka nimensä mukaisesti pyrkii hyödyntämään visuaalisen käyttöliittymän ominaisuuksia yksinkertaistamaan asennustoimenpidettä.



Kuva 4. Visuaalisella käyttöliittymällä varustettu ohjattu asennustoiminto.

Visuaalinen käyttöliittymä mielletään usein käyttäjäystävällisemmäksi prosessiksi, sillä asennusvalikkoa voi helposti navigoida hiiren vasemmalla painikkeella napsautellen, ja esimerkiksi sijaintien selaaminen on huomattavasti helpompaa, kun tietokoneen hakemistorakenne tulee selvästi esille klikatessa mahdollisesti esiintyvää ”Selaa”-painiketta.

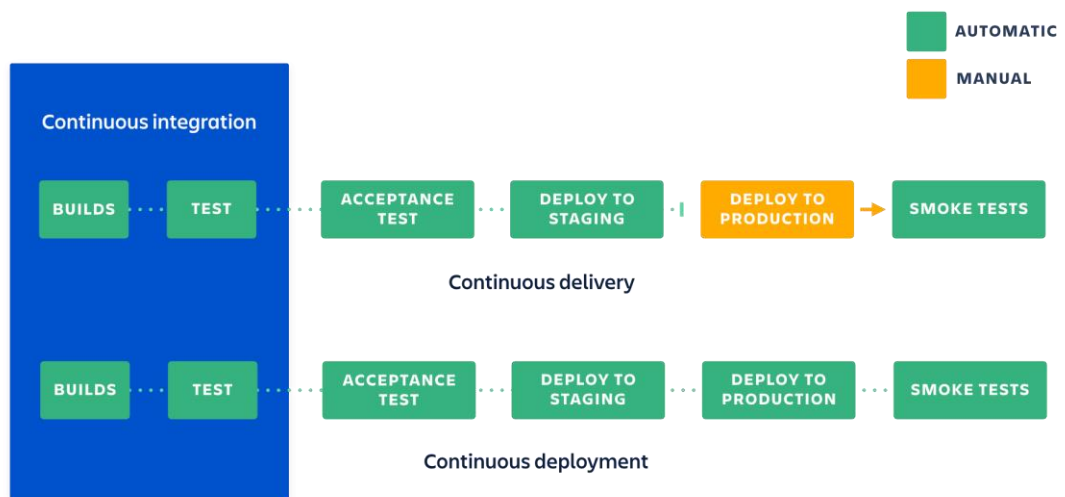
Sisimmiltään asennustavat eivät kuitenkaan loppupeleissä paljoa toisistaan eroa. Ohjatut asennustoiminnot keräävät käyttäjän valinnat konfiguraatitiedostoon ja suorittavat asennuksen samalla tavalla kuin se suoritettaisiin komentorivi-asennus suoritettaisiin.

2.6 CI/CD (Continuous Integration / Continuous Deployment tai Development)

Jatkuvassa käyttöönötossa (continuous deployment) (8) (kuva 5) ohjelmaversioiden siirtyessä versionhallintatyökalun päähaaraan (main branch) suoritetaan

sille automaattisesti ennalta määritetyt testit. Testien onnistuttua uusin versio asennetaan automaattisesti tuotantopalvelimelle.

Jatkuvaa käyttöönottoa varten tulee testauskäytäntöjen olla erinomaiset, mutta hyvänä puolena se mahdollistaisi usein tapahtuvat pienet ohjelmistopäivitykset harvojen suurempien päivitysten sijaan prosessin tehokkuuden vuoksi.



Kuva 5. Erot jatkuvan integraation, jatkuvan toimituksen ja jatkuvan käyttöönoton välillä. (9.)

Elematicin kannalta jatkuva käyttöönotto ei kuitenkaan ole toimiva vaihtoehto, sillä ohjelmistokehittäjien luomat ohjelmistoversiot eivät ole konfiguroitu asiakaspalvelimille suoraan lähetettäväksi, vaan konfigurointi täytyy suorittaa itse palvelimella. Elematicin asiakasyritysten tehtaiden toiminta on myös riippuvainen ohjelmiston toimivuudesta, joten sen odottamaton sulkeutuminen päivitysprosessin aikana voi tuottaa suuriakin ongelmia asiakasyrityksen tuotannossa. Päivityksien ajankohta on näin ollen aina sovittava asiakasyritysten kanssa etukäteen.

Jatkuva toimitus päivityksissä sen sijaan voisi toimiakin ajoitusten kannalta, mutta Elematicin tarjoaman ohjelmiston testaaminen vaatii visuaalisessa käyttöliittymässä navigointia, jonka vuoksi automatisoitu testaaminen voi tuottaa vaikeuksia jossain määrin. Tästä huolimatta CI/CD-menetelmän käyttöönottamisesta olisi mielestäni hyvä käydä keskustelua.

Ensimmäinen asennus tulee kuitenkin joka tapauksessa viedä manuaalisesti asiakasyrityksen palvelimelle, sillä sinne täytyy asentaa paljon eri ohjelmistoja ja luoda palvelinspesifejä konfiguraatioita. Nykyinen kehitysprosessi ei siis varsinaisesti tue CI/CD-menetelmän käyttöönottoa, se on sen verran suuri aihe, ettei tämä insinööri työ tule sitä käsittelemään tämän enempää.

3 Ongelman kuvaus

Uusille asiakkaille yrityksen tarjoaman ohjelmistokokonaisuuden asennus sekä päivitys vanhemmille asiakkaille on työläs prosessi. Tässä luvussa kuvataan asennustyökalun ratkaistavana oleva ongelma esittelemällä yksityiskohtaisesti asennuksessa manuaalista työtä vaativat työvaiheet. Toiminnan- ja tuotannon-ohjausjärjestelmien asentaminen riippuu asiakasyrityksen Elematicilta ostamasta ohjelmistokokonaisuudesta, joten tämä asennettavien sovelluksien valikoima tulee vaihtelemaan asiakkaalta toiselle.

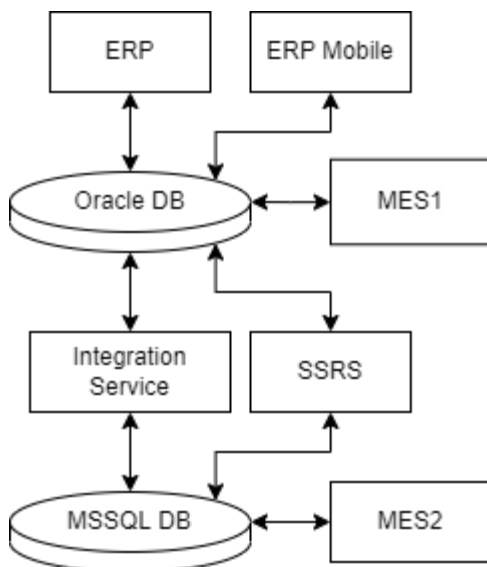
Asennusautomaatiotyökalun pohjimmainen tarkoitus on ratkaista kolme ohjelmistoasennuksiin liittyvää ongelmaa. Nämä kolme ongelmaa ovat manuaalisesta ohjelmistoasennuksesta mahdollisesti johtuvat huolimattomuusvirheet, asennusprosessin toimenpiteiden muistamisen hankaluus monimutkaisuuden vuoksi sekä kyseisen monimutkaisuuden aiheuttama asennusprosessin tehottomuus.

Kuvassa 6 esittelen asennusautomaatiotyökalun käsittelemät ohjelmistot ja niiden liitokset toisiinsa. Oraclen tietokannan kanssa keskustelevat ERP (Enter-

prise Resource Planning) eli toiminnanohjausjärjestelmä, toiminnanohjausjärjestelmän mobiiliversio, sekä yksi MES (Manufacturing Execution System) eli tuotannonohjausjärjestelmä (kuvassa 6 nimellä MES1).

Microsoftin tietokannan (MSSQL DB, Microsoft Structured Query Language Database) ja Oraclen tietokannan välissä keskustelevat SSRS (SQL Server Reporting Services) eli Microsoftin raportointipalvelinohjelmisto sekä Oraclen ja Microsoftin tietokantojen välinen integrointipalvelu.

Lopuksi vielä Microsoftin tietokannan kanssa keskustelee toinen tuotannonohjausjärjestelmä, johon kuvassa 6 viitataan nimellä MES2.



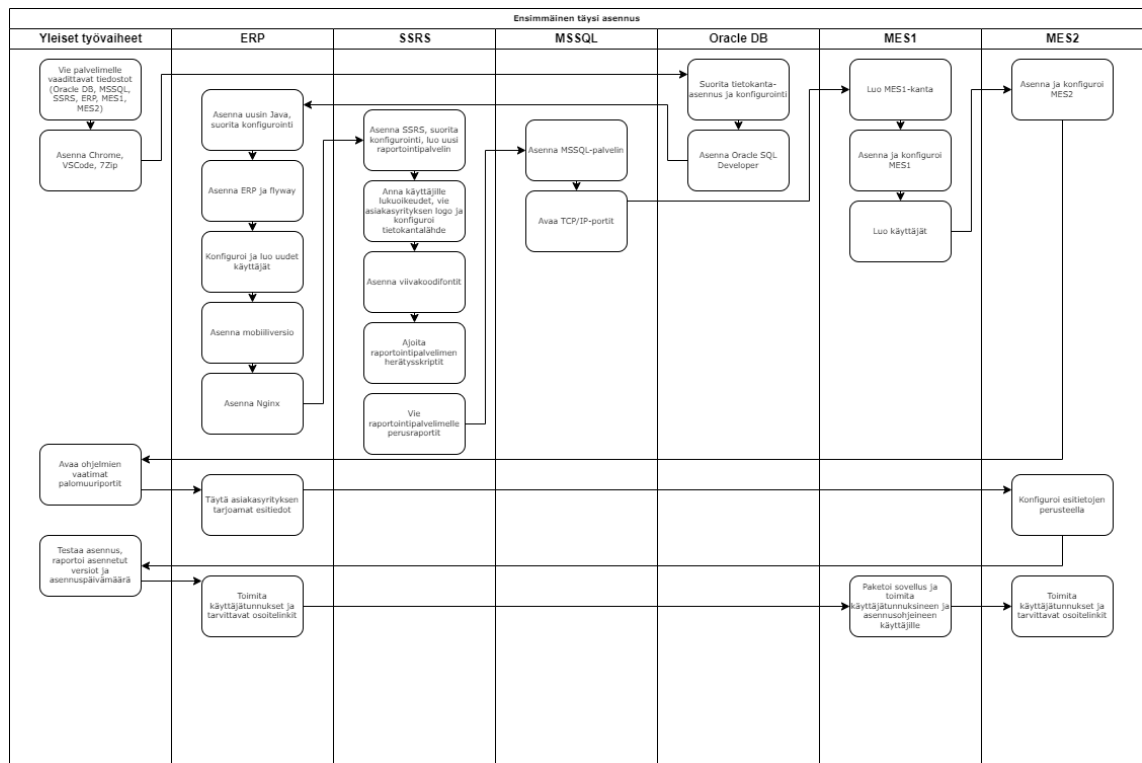
Kuva 6. Automaatiotyökalun käsittelemä ohjelmistokokonaisuus.

Kuten aiemmin tässä luvussa mainittiinkin, kuvassa 6 esiintyvän kaavion kuvaama ohjelmistokokonaisuus ei kuitenkaan kata kaikkia asennuksia, sillä Elematicin myymät toiminnan- ja tuotannonohjausjärjestelmät ovat ostettavissa erikseen. Jotkin asennukset voivat esimerkiksi sisältää ainoastaan toiminnanohjausjärjestelmäosuuden, jolloin asennus koostuu ainoastaan toiminnanohjausjärjestelmästä ja mahdollisesti siihen liittyvästä mobiiliversiosta, Oraclen tietokannasta sekä Microsoftin raportointipalvelinohjelmistosta.

Kuvan 6 kaavio kattaa ainoastaan tässä insinööriyössä käsiteltävän osuuden Elematicin ohjelmistosta yksinkertaisuuden vuoksi. Tuotenimet on jätetty pois, sillä ne eivät ole olennaisia tämän insinööriyön kannalta.

Kuvassa 7 esitellään ensimmäisen asennuksen vaatimat työvaiheet olettaen, että asiakas on ostanut koko ohjelmistokokonaisuuden. Suuri osa näistä työvaiheista on automatisoitavissa, mutta kuitenkin huomattava osuus jää silti manuaalisiksi toimenpiteiksi.

Yleensä koko asennukselle olisi hyvä varata aikaa useita tunteja, kuitenkin vähintään neljä. Automatisoimalla kaikki automatisoitavat työvaiheet uskoisin asennusprosessin keston putoavan alle kahteen tuntiin.



Kuva 7. Ensimmäisen asennuksen sisältämät työvaiheet koko ohjelmistokokonaisuuden ostaneelle asiakkaalle.

Mikäli asiakas ei ole ostanut jotain tämän insinööriyön käsittelemistä kolmesta toiminnan- ja tuotannonohjausjärjestelmästä, voi siihen liittyvät asennusvaiheet ohittaa.

3.1 Työkalujen asennus

Yrityksen työntekijät joutuvat aika ajoin suorittamaan erilaisia palvelimen ylläpitotoimenpiteitä asiakaspalvelimilla etäyhteyden välityksellä, joten palvelimelle on hyvä asentaa usein käytetyt työkalut valmiiksi. Osaa näistä työkaluista tarvitaan myös ohjelmistokokonaisuuden asennuksen yhteydessä.

Palvelimille asennetaan usein seuraavat yleiset työkalut:

- Google Chrome -verkkoselain
- Visual Studio Code -tekstieditori
- 7zip-tiedostonpakkaus- ja arkistointiohjelma
- Nginx-web-palvelin
- Oracle SQL Developer IDE (Integrated Development Environment).

Tämä vaihe tulee suorittaa ainoastaan ensimmäisen asennuksen yhteydessä, mutta esimerkiksi toiminnan- ja/tai tuotannonohjausjärjestelmiä päivitettäessä olisi hyvä kyetä mahdollisesti päivittämään työkalut uusimpiin versioihin, mikäli niitä on saatavilla.

3.2 Oraclen ja Microsoftin tietokannat

Oraclen tietokanta kuuluu jokaiseen asennukseen, sillä tämä toimii yhteisenä tietokantana lähes kaikkien Elematicin käyttämien sovelluksien tiedolle. Oraclen tietokannan visuaalisella käyttöliittymällä varustetun ohjatun asennustoiminnon voi ohittaa hyödyntämällä Oraclen tarjoamaa db_install.rdp-responssitiedostoa.

Responssitiedostot ovat hiljaisissa asennusoperaatioissa käytettyjä konfiguraatiotiedostoja, jotka sisältävät samoja asetustietoja kuin mitä määritellään visuaalisella käyttöliittymällä varustetuissa ohjatuissa asennustoiminnoissa.

Tämän lisäksi prosessiin kuuluu myös komentoriviä vaativia toimenpiteitä, kuten esimerkiksi palvelinnimen haku, TNS-kuuntelijan (Transparent Network

Substrate listener) käynnistys, tietokannan ja kuuntelijan avaaminen ja toimivuuden testaaminen sekä palvelinnimen lyhenteen asettaminen jatkotoimenpiteiden helpottamista varten.

Loput ohjelmista, eli yksi toiminnanohjausjärjestelmä sekä Microsoftin raportointipalvelinohjelmisto ottavat yhteyden Microsoftin relaatiotietokantapalvelimeen.

Microsoft SQL Serverin asennus on hyvin samankaltainen prosessi Oraclen tietokannan kanssa. Se sisältää myös komentorivin avulla ohitettavan visuaalisen ohjatun asennustoiminnon. Microsoftin kannan kanssa ei kuitenkaan ole pakko käyttää Oraclen tapaan responssitiedostoa, vaan parametrit voidaan myös syöttää suoraan asennusohjelmaa ajaessa komentoriviargumentteina.

3.3 Toiminnanohjausjärjestelmä

Yrityksen tarjoaman toiminnanohjausjärjestelmän asennus vaatii viimeisimmän Javan asennusta palvelimelle, konfigurointia, asennuskriptien ajamista, Flyway-asennuksen (tietokannan migraatiotyökalu), osoitteiden asettamista palvelinspesifeiksi sekä asennuksen onnistumisen testausta.

Yrityksen toiminnanohjausjärjestelmä on toteutettu Oraclen Apex-työkalulla (10), joten näin ollen se on verkossa ajettava sovellus perinteisen työpöytäsovelluksen sijaan.

Oraclen Apex-työkalu on suunniteltu tarjoamaan yrityksille vähäkoodista alustaa yrityksen luomille sovelluksille tietokantayhteydellä.

Toiminnanohjausjärjestelmän mobiiliversion asennus tapahtuu siirtämällä tuotantoon valmistettu asennuspaketti toiminnanohjausjärjestelmän yhteydessä luotuun webapps-alihakemistoon.

3.4 SSRS (SQL Server Reporting Services)

SSRS:än visuaalisen asennustoiminnon voi myös ohittaa jo tähän mennessä tuttuun tapaan komentorivin kautta. Kuvassa 8 esitellään asennuksen yhteydessä käytettäviä parametreja. Kyseinen ikkuna tulee esiin kirjoittaessa parametriksi /help asennustiedostoa ajaessa.



Kuva 8. SSMS (SQL Server Management Studio) -asennuksen parametrien apunäyttö.

Komentoriviä tarvitaan myös asiakasyrityksen logon viemiseen oikeaan sijaan, viivakoodifontteja asennettaessa sekä raportointipalvelinohjelmiston herätyskriptejä vietäessä Windowsin tehtävien ajoitukseen.

Raportointipalvelinohjelmisto tarvitsee herätyskriptejä, sillä se menee yön aikana lepotilaan, josta herääminen kestää useita minuutteja. Herätyskriptit ajetaan automaattisesti aamulla viiden aikaan, jotta kun asiakkaat haluavat ajaa raportteja, ei heidän tarvitse odotella palvelimen käynnistymistä.

3.5 Tuotannonohjausjärjestelmät

Yrityksellä on kahdelle eri tuotekategorialle omat tuotannonohjausjärjestelmät, joilla on omat eriävät asennusprosessinsa. Kummatkin vaativat paljon asetustiedostojen läpikäyntiä palvelinosoitteita muutettaessa kehityspalvelimen osoitteista asiakaspalvelimen osoitteisiin.

3.6 Viimeistelyt

Tilattujen ohjelmistojen asennuksen jälkeen täytyy vielä avata palomuriin ohjelmistojen käyttämät portit. Sisäistä kirjanpitoamme varten tulee raportoida palvelimelle asennettujen ohjelmistojen versionumerot sekä asennuspäivämäärä. Asennuksen yhteydessä potentiaalisesti ylimääräiseksi jääneet tiedostot tulisi siivota pois. Lopuksi ohjelmistojen loppukäyttäjille täytyy luoda käyttäjätunnukset ja toimittaa kyseiset tunnukset sekä asiakkaiden omille koneille asennettavat sovellukset ja mahdollisesti kiinnostavat verkko-osoitteet ohjeineen sähköpostitse.

4 Projektin ympäristö

Tämän luvun tarkoituksena on perehtyä projektin lopputuloksen luomista varten päädyttyihin työkaluihin ja käyttöjärjestelmäympäristöihin.

4.1 Testikäyttöjärjestelmäympäristö

Projektin toteutus kohdistui täysin Windows Server -käyttöjärjestelmäryhmään, sillä kyseistä käyttöjärjestelmäympäristöä kaikki asiakaspalvelinasennuksemme käyttävät poikkeuksetta. Tämän vuoksi PowerShell oli looginen vaihtoehto projektin skriptejä varten. Vaikka vähintäänkin suuren osan skripteistä olisi voinut luoda MS-DOS:n aikakauden komentojonoilla (batch), ei nykypäivänä mielestäni ole järkeä luoda näillä uusia ratkaisuja. PowerShell on modernimpi ja kykenee vähintäänkin samoihin toimenpiteisiin, mutta tarvittaessa myös paljon muuhunkin.

4.2 PowerShell

PowerShell on Microsoftin alun perin Windows-käyttöjärjestelmille kehittämä komentotulkki ja skriptikieli muun muassa erilaisten toimintojen automaatiota varten.

PowerShellin erikoisuuksista on cmdlets:it, jotka ovat komentoja PowerShell-ympäristössä. Nämä komennot on nimetty verbi-substantiivi-käytännöllä, jonka vuoksi PowerShell on itseään dokumentoiva kieli. Pitkien komentonimien sijasta komennoista voi myös halutessaan käyttää aliaksia, mikäli käyttäjä pitää enemmän bash-tyyppisestä, paljon lyhenteitä hyödyntävästä estetiikasta. Seuraavassa PowerShell-otteessa on esimerkki PowerShellin Get-ChildItem-komennosta ja sen bash-tyylisestä aliaksesta ls.

```
PS C:\Elematic\Install\SSRS> Get-ChildItem
    Directory: C:\Elematic\Install\SSRS

Mode                LastWriteTime         Length Name
----                -
d-----           19.4.2022          6.17          Viivakoodi
-a-----           24.10.2019         23.17       560361832 SSMS-Setup-ENU.exe
-a-----           12.8.2020          2.04       110536768 SSRS.exe
-a-----           7.2.2022           6.20           3662 SSRS Restart.xml
-a-----           7.2.2022           6.20           3808 SSRS Wakeup.xml

PS C:\Elematic\Install\SSRS> ls
    Directory: C:\Elematic\Install\SSRS

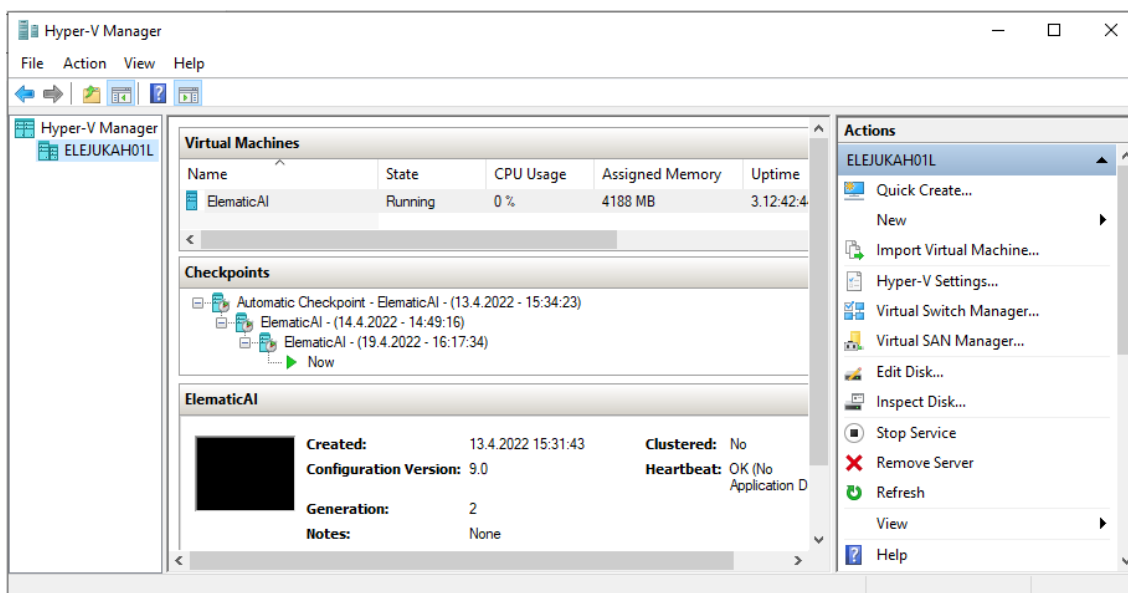
Mode                LastWriteTime         Length Name
----                -
d-----           19.4.2022          6.17          Viivakoodi
-a-----           24.10.2019         23.17       560361832 SSMS-Setup-ENU.exe
-a-----           12.8.2020          2.04       110536768 SSRS.exe
-a-----           7.2.2022           6.20           3662 SSRS Restart.xml
-a-----           7.2.2022           6.20           3808 SSRS Wakeup.xml
```

Kuva 9. PowerShell-komennon Get-ChildItem ja sen aliaksen ls vertailua.

PowerShell tukee myös Windows-komentoa dir, jolla saa luotua samanlaisen listan kuin komennoilla ls ja Get-ChildItem. Henkilökohtaisesti suosittelisin skriptejä kirjoittaessa kuitenkin pysyttämään pidemmissä komentonimiversioissa, mutta komentorivin kautta suoraan ilman skriptejä työskennellessä aliakset ovat oikein käytännöllisiä.

4.3 Testiympäristö

Asennusautomaatiotyökaluja luodessa niitä olisi myös hyvä testata. Päädyin valitsemaan testiympäristöksi kuvassa 4 näkyvän Microsoftin Hyper-V-virtuaalisattorin (kuva 10). Virtuaalikoneeseeni asensin Windows Server 2022 -käyttöjärjestelmän ja loin asennuksen valmistumispisteestä varmuuskopion. Tällä tavoin voin tarvittaessa aina palata takaisin lähtöpisteeseen nopeasti, esimerkiksi kun haluan suorittaa järjestelmäkokonaisuuden asennuksen alusta alkaen uudelleen varmistaakseni työkalun toimivuuden sekä käydäkseni prosessia läpi kehitysvaiheessa.



Kuva 10. Hyper-V Manager.

Hyper-V:n käyttöönotto on yksinkertaista. Esimerkiksi PowerShellillä järjestelmänhaltijana ajettaessa se otetaan käyttöön seuraavalla komennolla:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

Kuvan 10 näkymässä on virtuaalikoneeni jo käynnissä, mutta uuden luominen tapahtuisi Action-valikon alta valitsemalla New -> Virtual Machine ja seuraamalla tästä aukeavaa ohjattua asennustoimintoa.

Luomani automaatiokriptit lataan versionhallintapalvelimelle, josta voin ne aina helposti käydä kloonaamassa palatessani aiempaan palautuspisteeseen.

5 Ohjelmistoasennuksen automaation toteutus

Tässä luvussa perehdytään automaatiokriptien luomiseen. Automaatiokriptit ovat merkkipohjaisia ohjelmia, joiden tarkoituksena on vähentää eri toimenpiteiden manuaalista suorittamista automaatiota hyödyntäen.

5.1 Palvelinasetukset

Ensimmäisenä suoritettava skripti hakee palvelimelle spesifit asetukset tarkastelemalla muun muassa palvelimen IP-osoitteen ja isäntänimen (hostname) sekä skriptin ajosijainnin, jonka perusteella se luo oletuksena toimivan hakemiston asennuksille (useimmiten C:\Elematic\Install).

Haettuaan nämä oletusasetukset skripti kysyy käyttäjältä vielä varmistuksen tietojen kelvollisuudesta, käyttäjä voi niitä tarpeen vaatiessa vielä muuttaa. Tämän jälkeen tiedot talletetaan asetustiedostoon, josta muut skriptit sitten hakevat tarvitsemaansa informaatiota.

5.2 Työkalujen ja sovellusten asennuksen automaatio

Jokaiselle palvelimelle asennetaan samat usein käytetyt työkalut ja sovellukset. Tämä vaihe on pakollinen uusien asiakkaiden kohdalla, mutta esimerkiksi toiminnan- ja/tai tuotannonohjausjärjestelmäpäivityksien yhteydessä olisi hyvä kyetä tällä skriptillä pitämään myös työkalut ajan tasalla.

Ratkaistavana tässä on kaksi vaihetta: kuinka asennussovellukset haetaan ja kuinka itse asennus hoidetaan.

Ensimmäistä vaihetta varten päätin toistaiseksi yksinkertaisesti vain ladata asennussovellukset lokaalisti tallentamaani valmiiseen pakettiin, joka viedään

asiakkaan koneelle uusia asennuksia suoritettaessa. Optimaalisesti skripti noutaisi verkosta viimeisimmän version asennettavaksi, mutta tästä saatu hyöty ei ole tarpeeksi korkea priorisoidakseni sitä tässä vaiheessa.

Jälkimmäistä vaihetta varten kirjoitin PowerShell-skriptin, joka lukee kansion sisällön ja jokaisen .exe-tiedoston kohdalla kysyy käyttäjältä, ajetaanko kyseinen tiedosto. Käyttäjän vastatessa positiivisesti skripti optimaalisesti syöttäisi asennussovellukselle /silent-parametrin, jolla saataisiin asennus suoritettua taustalla ilman tarvetta navigoida visuaalisen käyttöliittymän asennusprosessia sekä menettämättä fokusta skriptistä.

Tämä kuitenkin osoittautui yllätyksellisesti oletettua monimutkaisemmaksi, sillä kaikki asennussovellukset eivät hyväksykään /silent-parametria. Kyseinen parametri toimi moitteettomasti ainoastaan Google Chromea asennettaessa aiemmin annetuista esimerkeistä. Visual Studio Code suostui sillä asentumaan, mutta toi näytölle asennuksen edistymisikkunan. Täysin hiljaista asennusta varten Codelle täytyi parametriksi syöttää /verysilent. 7zip taas ei suostunut toimimaan ollenkaan kummallakaan näistä vaihtoehdoista vaan vaati parametrisen /S.

Kuten esimerkikoodissa 1 näkyy, päätin luopua parametrin käytöstä kokonaan, sillä tämän ongelman ratkaiseminen ei ole korkein prioriteetti projektiani ajatellen. En koe mieluisaksi vaihtoehdoksi kovakoodata sopivia parametrejä jokaiselle mahdolliselle sovellukselle erikseen.

```
$path = '.\sovellukset\'
Get-ChildItem -Filter $path*.exe | ForEach-Object {
    $installer = $PSItem.BaseName+$PSItem.Extension
    $confirmation = Read-Host 'Install' $installer 'y/[n]?'
    if ($confirmation -eq 'y') {
        Write-Host 'Installing' $installer
        & $path$installer
    } else {
        Write-Host 'Skipping' $installer
    }
}
```

Esimerkkikoodi 1. Asennussovellusten järjestelmällinen läpikäynti ForEach-Object-silmukalla.

Esimerkkikoodi 1 käy läpi kaikki hakemistossa ".\sovellukset\" olevat .exe-tiedostot, kysyy käyttäjältä varmistuksen ja toimii sen mukaisesti joko käynnistäen asennussovelluksen tai ohittaen asennuksen.

Syöttämällä Get-ChildItem-komennolle parametrin -Filter, voi komennon palauttamia tuloksia rajata. Jokerimerkki (*) korvaa mitä tahansa merkkijonoa, joten tässä tapauksessa rajaus \$path*.exe palauttaa kaikki sijainnissa .\sovellukset\ olevat .exe-tiedostot.

ForEach-Object-silmukalla kyetään suorittamaan toimenpiteitä jokaisen tuloksen kohdalla. PowerShellissä aktiivisena olevaan tulokseen ForEach-Object-silmukassa viitataan muuttujanimellä \$PSItem. Vaihtoehtoisesti voi PSItemin tilalla myös käyttää alaviivamerkkiä. En kuitenkaan koe sitä yhtä luettavaksi, joten suosittelen käyttämään PSItemia.

Read-Host-komennolla luetaan käyttäjältä syötettä, joka tallennetaan if-lausekkeessa -eq-operaattorilla vertailtavaan muuttujaan. Vertailuun käytettävä -eq-operaattori yllätti, sillä siitä ei ole PowerShellin tähän mennessä yleiseltä luettavuuskäytännöltä tuntuva vastinetta "-equals".

Esimerkkikoodin 1 rivillä 7 käytettävä &-merkki toimii kutsumaoperaattorina, jolla voidaan ajaa monimutkaisesti nimettyjä tiedostoja. Esimerkiksi tässä tapauksessa ajettavien asennusohjelmien nimi ja sijainti muodostetaan yhdistämällä muuttujia, joten sellaisenaan PowerShell ei tiedä, kuinka toimia.

Viimeisimpänä komentona skriptissä käytetään Write-Host-komentoa, jolla voidaan syöttää tekstiä käyttäjän ruudulle.

Tämän skriptin etenemistä komentorivillä esitellään kuvassa 11.

```
PS C:\Elematic\Install> .\install_programs.ps1

Install 7z2107-x64.exe y/[n]?: y
Installing 7z2107-x64.exe
Install ChromeSetup.exe y/[n]?: n
Skipping ChromeSetup.exe
Install VSCodeSetup-x64-1.66.2.exe y/[n]?:
Skipping VSCodeSetup-x64-1.66.2.exe
```

Kuva 11. Työkalunasennusskriptin eteneminen komentorivillä.

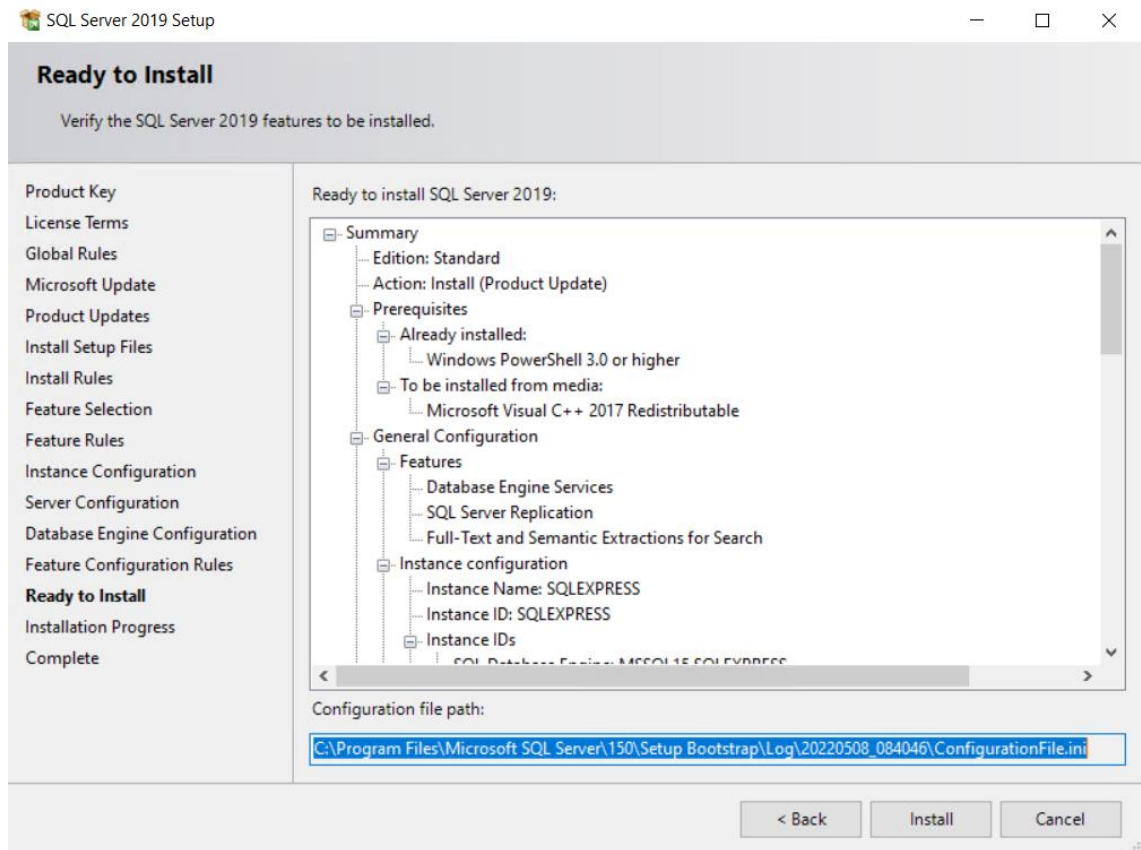
Hakasulkeilla ilmaistaan oletusvastaus, joka hyödynnetään, mikäli käyttäjä painaa enter-näppäintä syöttämättä mitään arvoa. Vaikka vaihtoehdoiksi annetaan kirjaimet y ja n, ei niistä oikeasti tarkasteta kuin vain y. Kaikki muu ymmärretään negatiivisena vastauksena.

5.3 Tuotannonohjausjärjestelmän asennuksen automaatio

Edellisessä aliluvussa tarkasteltiin työkalujen ja sovellusten asennuksen automaatiota. Seuraavaksi tutustutaan Elematicin asiakasyrityksille tarjoaman tuotannonohjausjärjestelmän asennusprosessin automaatioon.

Ensimmäisenä asennetaan tuotannonohjausjärjestelmän käyttämä tietokanta. Tämän hiljaiseen asentamiseen komentorivin kautta on kätevä oikotie asetustiedostoa tulevaisuuden asetuksia varten. Ensin suoritetaan asennus käyttäen visuaalisella käyttöliittymällä varustettua ohjattua asennustoimintoa, valitaan halutut asetukset ja edetään kohtaan "Ready to Install".

Tässä vaiheessa aukeaa kuvassa 12 esitelty yhteenvetonäkymä valituista asetuksista. Näkymän alareunassa viitataan konfiguraatitiedoston sijaintiin.



Kuva 12. SQL Server -ohjelmiston ohjattu asennus.

Avaamalla tämän tiedoston (kuva 13) saadaan selville kaikki tarpeelliset parametrit hiljaista asennusta varten.

```

C: > Program Files > Microsoft SQL Server > 150 > Setup Bootstrap > Log > 20220508_084046 > ConfigurationFile.ini
1 ;SQL Server 2019 Configuration File
2 [OPTIONS]
3 IACCEPTSQLSERVERLICENSETERMS="True"
4 ; By specifying this parameter and accepting Microsoft R Open and Microsoft R Server te
5 IACCEPTPYTHONLICENSETERMS="False"
6 ; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE. This is a required
7 ACTION="Install"
8 ; By specifying this parameter and accepting Microsoft R Open and Microsoft R Server te
9 IACCEPTROPENLICENSETERMS="False"
10 ; Specifies that SQL Server Setup should not display the privacy statement when ran fr
11 SUPPRESSPRIVACYSTatementNOTICE="False"
12 ; Use the /ENU parameter to install the English version of SQL Server on your localized
13 ENU="True"
14 ; Setup will not display any user interface.
15 QUIET="True"
16 ; Setup will display progress only, without any user interaction.
17 QUIETSIMPLE="False"
18 ; Parameter that controls the user interface behavior. Valid values are Normal for the
19 UIMODE="Normal"
20 ; Specify whether SQL Server Setup should discover and include product updates. The val
21 UpdateEnabled="True"
22 ; If this parameter is provided, then this computer will use Microsoft Update to check
23 USEMICROSOFTUPDATE="True"
24 ; Specifies that SQL Server Setup should not display the paid edition notice when ran
25 SUPPRESSPAIDEDITIONNOTICE="False"
26 ; Specify the location where SQL Server Setup will obtain product updates. The valid v
27 UpdateSource="MU"

```

Kuva 13. MSSQL-konfiguraatitiedosto (12).

Kuvassa 13 näkyvässä konfiguraatitiedostossa oleva rivi 3 täytyy lisätä sinne käsin, sillä asennusohjelma ei näyttänyt sitä sinne itse syöttävän. Asennusohjelma ei myöskään aseta järjestelmänvalvojan salasanaa, joten tämä tulee myös lisätä tiedostoon parametrina "SAPWD" (System Administrator Password).

Hiljaista asennusta varten rivillä 14 näkyvän parametrin "QUIET" arvoksi on syötettävä "True" ja rivillä 19 näkyvän parametri "UIMODE" tulee poistaa. Lopuksi täytyy vain ottaa tiedosto talteen ja viitata siihen uutta asennusta tehdessä.

Esimerkkikoodissa 2 ajetaan asennussovellus mukautetulla konfiguraatitiedostolla sekä tuoteavaimella.

```

. .\server_settings.ps1
E:\setup.exe /CONFIGURATIONFILE='.\mssql_config.ini' /PID=$productKey

```

Esimerkkikoodi 2. Asennussovelluksen ajaminen konfiguraatitiedostolla ja tuoteavaimella.

Esimerkkikoodin 2 ensimmäisellä rivillä ensimmäisenä merkinä olevaa pistettä kutsutaan termeillä "source" tai "dot operator" (11). Pisteoperaattorilla saadaan tuotua vieraan tiedoston käyttämät muuttujat sekä funktiot käyttöön toisessa tiedostossa. Tällä tavoin saadaan server_settings.ps1-tiedostoon määritelty muuttuja tuoteavaimelle käyttöön ajettaessa asennustiedostoa.

Esimerkkikoodissa 3 esitellään tietokantapalvelimen porttien konfigurointia. Tämän prosessin automaatio jäi hieman puolitiehen ajan puutteesta johtuvan priorisoinnin vuoksi, joten skripti avaa konfiguraatio-ohjelman ja tulostaa ruudulle muistion toteutettavista toimenpiteistä.

```
C:\Windows\SysWOW64\mmc.exe /32 C:\Windows\SysWOW64\SSM15.msc
Write-Host ""SQL Server Configuration Manager
    -> SQL Server Network Configuration
    -> Protocols for SQLEXPRESS
    -> Enable Named Pipes and TCP/IP
    -> Double click on TCP/IP
    -> IP Addresses
    -> Remove all TCP Dynamic port numbers
    -> Set all TCP ports to 1433 (or whatever)""
```

Esimerkkikoodi 3. SQL Server Configuration Managerin avaaminen ja manuaali konfigurointi.

Oletuksena asennus vaatii paljon manuaalista konfiguraatitiedostojen läpikäyntiä sekä muun muassa pakettien purkamista ja Windows-palveluiden luomista. Konfiguraatitiedostojen konfiguroinnin automaatio vaatii PowerShell-skriptin, joka hakee palvelimen IP-osoitteen sekä palvelinnimen aliluvussa 5.1 mainitusta asetustiedostosta ja käy konfiguraatitiedostot läpi korvaten vakiona olevat localhost-isäntänimet palvelinspesifeillä asetuksilla.

Esimerkkikoodissa 4 esiteltävä skripti käy läpi tiivistettyjä kansioita ja purkaa ne omaan sijaintiinsa.

```
$path = '.\FM\'
$destination_path = 'C:\Elematic\FM\'
Get-ChildItem -Filter $path*.zip | ForEach-Object {
    $confirmation = Read-Host 'Extract' $PSItem.BaseName 'y/[n]?'
}
```

```

    if ($confirmation -eq 'y') {
        Write-Host 'Extracting package' $PSItem.BaseName
        Expand-Archive $path$PSItem -DestinationPath $destination_path\'$PSItem.BaseName -Force
    } else {
        Write-Host 'Skipping' $PSItem.BaseName
    }
}

```

Esimerkkikoodi 4. Tiivistettyjen kansioden purkamista varten kehitetty skripti.

Esimerkkikoodissa 4 esitellään kansioden purkamista PowerShellin avustuksella. Skriptin eteneminen muistuttaa hyvin paljon edellisessä aliluvussa esiteltyä sovellusten asentamiseen luotua skriptiä. Sovellusten asennustiedostojen suorittamisen sijaan tämän skriptin tarkoituksena on löytää ja purkaa hakemistoja.

Kuvassa 14 esitellään purkuskriptin kulkua.



Kuva 14. Hakemistoja puretaan seuraavaa skriptiä varten.

Purettuaan hakemistot lopulliseen sijaintiinsa täytyy konfiguraatioskriptin täydentää vielä IP-osoitteet asetustiedostoista ja palvelinskriptin luoda Windows-palvelut (esimerkkikoodi 5).

```

$path = 'C:\Elematic\FM\'
Write-Host '-----Windows service creation-----'
Get-ChildItem $path | ForEach-Object {
    $confirmation = Read-Host 'Create a service for' $PSItem.BaseName
    'y/[n]?'
    if ($confirmation -eq 'y') {
        Write-Host 'Creating service' $PSItem.BaseName
        $service = Get-ChildItem $path$PSItem -Filter *Service.exe
        $servicePath = "$path$PSItem\$service"
    }
}

```

```

$params = @{
    Name = 'Elematic.FM.'+$PSItem.BaseName
    BinaryPathName = $servicePath
    StartupType = 'Automatic'
}
New-Service @params
Start-Service -Name $params.Name
} else {
    Write-Host 'Skipping' $PSItem.BaseName
}
}
}

```

Esimerkkikoodi 5. MES (Manufacturing Execution System) -järjestelmän vaatimien Windows-palveluiden luomisen automaatiota ForEach-Object-silmukalla.

Päivitettäessä aikaisempaa asennusta tulisi edellä mainittujen vaiheiden lisäksi ottaa tietokannasta varmuuskopio SSMS (SQL Server Management Studio) -sovellusta hyödyntäen sekä siirtää tuotannonohjausjärjestelmän testaamista varten luotu lisenssitiedosto vanhasta versiosta uuteen.

5.4 SSRS

Raportointipalvelinohjelmiston asennuksen voi suorittaa myös hiljaisena komentorivin kautta. Raportointipalvelin täytyy joka aamu erikseen herättää, sillä muuten päivän ensimmäisen raportin avaamisessa kestää hyvin kauan. Näin ollen Windowsin tehtävien ajoitukseen on vietävä kaksi skriptiä suorittamaan kyseinen operaatio. Raportointipalvelinohjelmisto vaatii myös ylimääräisten fonttien asennuksen viivakoodeja varten.

5.5 Oracle-tietokanta

Oracle-tietokannan asennuksen voi suorittaa hiljaisesti komentoriviltä hyödyntämällä kuvassa 5 näkyvää Oraclen tarjoamaa "db_install.rsp"-responssitiedostoa (13). Responssitiedosto sisältää täytettävät parametrit jokaiselle ohjatussa

asennustoiminnossa kysytylle asetukselle. Parametrien täyttämisen jälkeen kyseinen tiedosto syötetään argumentiksi setup.exe-tiedostoa ajaessa, jolloin asennus suoritetaan taustalla.

```
#####
## Copyright (c) Oracle Corporation 1998,2019. All rights reserved.##
##
## Specify values for the variables listed below to customize
## your installation.
##
## Each variable is associated with a comment. The comment
## can help to populate the variables with the appropriate
## values.
##
## IMPORTANT NOTE: This file contains plain text passwords and
## should be secured to have read permission only by oracle user
## or db administrator who owns this installation.
##
#####

#-----
# Do not change the following system generated value.
#-----

oracle.install.responseFileVersion=/oracle/in-
stall/rspfmt_dbinstall_response_schema_v19.0.0
```

Kuva 15. Ote Oraclen responssitiedostosta.

5.6 Viimeistelyt

Mobiilisovelluksen asentaminen on yksinkertainen paketin siirto Apexin alla olevaan webapps-hakemistoon. Käyttäjille tulisi luoda tunnukset sähköpostiosoitteen perusteella. Tämänkin prosessin voisi automatisoida, generoida salasanat ja lähettää tunnukset sähköpostitse.

6 Lopputulos

Tässä luvussa tarkastellaan lopputulosta ja sitä, kuinka se vastaa alkuperäisiä odotuksia.

Automaatiotyökalun oli tarkoitus ratkaista kolme ohjelmistoasennuksiin liittyvää ongelmaa, jotka ovat ohjelmistoasennuksissa potentiaalisesti esiintyvät huolimattomuusvirheet, asennuksia suorittamalta henkilöltä vaadittu suoritettujen toimenpiteiden muistaminen sekä asennusoperaation monimutkaisuuden aiheuttama tehottomuus.

Seuraavaksi tarkastellaan työkalun pätevyyttä jokaisen näiden kolmen ongelman ratkaisussa yksitellen.

6.1 Ohjelmistoasennuksissa esiintyvät huolimattomuusvirheet

Ennen asennustyökalua oli ohjelmistoasennuksia suorittavan henkilön vastuulla täydentää palvelinspesifit hakemistosijainnit sekä IP-osoitteet ja isäntänimet oikeisiin konfiguraatitiedostoihin. Palomuriin täytyi avata oikeat portit.

Työkalu hakee osoitteet automaattisesti hyödyntäen tietokoneelta saatua informaatiota ja osaa itse täydentää konfiguraatitiedostoihin vaaditut muutokset, joten teoriassa asennuksesta johtuvien virheiden tulisi kadota kokonaan.

6.2 Asennusten toimenpiteiden muistaminen

Jokainen asiakasyrityksen palvelimelle asennettava ohjelmisto piti hakea eri sijainnista ja monesti ne koostuivat useammasta eri komponentista. Esimerkiksi selaimessa toimivan toiminnanohjausjärjestelmän asennus vaati myös Nginx-web-palvelimen asennusta. Raportointipalvelinohjelmiston asennus vaati viiva-koodifonttien asentamista sekä herätyskriptien asettamista Windowsin tehtävien ajoitukseen. Konfiguraatitiedostoja on useita ja kaikki niistä ovat kaikkien muiden tiedostojen seassa, ja niin edelleen.

Toimenpiteiden muistaminen ulkoa vaatii paljon kertausta, mutta näitä asennuksia ei välttämättä suoriteta läheskään joka viikko, joten käytännössä ohjelmistojen asennuksesta selviää ainoastaan kattavilla muistiinpanoilla. Mikäli jonkun

muun kuin useimmiten asennuksia suorittavan henkilön täytyisi suorittaa asennus, tulisi näiden muistiinpanojen olla helposti muidenkin ymmärrettävissä.

Asennustyökalu yhdistää kaikki nämä toimenpiteet yhdestä skriptistä ajettavaksi kokonaisuudeksi ja välittää tiedon suoritetuista toimenpiteistä skriptin ajajalle, joten käytännössä muistettavaa ei juurikaan ole, kunhan ajaja muistaa viedä asiakaspalvelimelle asennuksen vaatimat paketit ja lähettää skriptin luomat tunnukset sähköpostitse ohjelmistojen käyttäjille.

6.3 Ohjelmistoasennuksen monimutkaisuudesta aiheutuva tehottomuus

Työvaiheiden suuri määrä sekä monimutkainen luonne tekee Ohjelmiston asennusoperaatiosta ajallisesti hyvin tehottoman prosessin. Tätä on hyvä ajatella myös siltä kantilta, että ohjelmistoasennuksen käyttämä aika ei välttämättä pysähdy täysin siihen, kun asennus on ensimmäisen kerran suoritettu, sillä potentiaalisten huolimattomuusvirheiden korjaaminen asennuksen jälkeen tulisi laskea mukaan tähän ajankäytön analysointiin.

Huolimattomuusvirheiden minimointia ja toimenpiteiden yhteen ryhmittelyä käsiteltiin jo aiemmissa aliluvuissa 6.1 ja 6.2, joten niitä ei ole tarpeellista tässä erikseen mainita, vaikka niilläkin on vaikutusta asennustyökalun tehokkuuteen. Mutta niihin lisäten, asennustyökalu onnistuu täysin ohittamaan kolme graafisella käyttöliittymällä varustettua ohjattua asennustoimintoa, jolla on hyvinkin suuri vaikutus ajankulutuksen minimointia ajatellen.

6.4 Automaatiotyökalun tehokkuuden analysointi kokonaisuutena

Insinööriyön lopputuloksena syntyi hyvä lähtökohta ohjelmistoasennusautomaatiolle. Työkalu on helposti mukautettavissa modulaarisen luonteensa vuoksi, jossa jokainen toiminto on paloitetu omaksi tiedostokseen ja lisää tehokkuutta jokaisella kolmella tässä luvussa aiemmin mainitulla osa-alueella.

Tästä huolimatta automaatiotyökalussa on vielä paljon varaa jatkokehitykselle. Asennettavista työkaluista olisi hyvä saada automaattisesti ladattua viimeisin versio, skriptien ajon aikaiset komentorivitulosteet olisi hyvä tallentaa lokitiedostoihin myöhempää tarkastelua varten. Osa suoritettavista toimenpiteistä jäi automatisoimatta, kuten TCP/IP:n (Transmission Control Protocol / Internet Protocol, Internetissä käytettävä tietoliikenneprotokollapino) porttien konfigurointi.

Mikäli asennettavien sovelluksien rakenne tulee tulevaisuudessa muuttumaan, se voi potentiaalisesti aiheuttaa ongelmia työkalun kanssa. Esimerkkinä tästä voisi olla esimerkiksi muutos tiedostojen nimeämiskäytännössä.

Tällaisilta ongelmilta on kuitenkin mahdollista välttyä sopimalla yrityksen sisäisesti yhtenäisistä käytännöistä, joita noudatetaan. Mikäli niiden muuttuminen on välttämätöntä, tulisi tätä varten keskustella siitä etukäteen ja tehdä yhdessä jatkosuunnittelua.

Dockerin käyttöönottoa harkittiin insinööriyön suunnitteluvaiheessa, mutta aika-
tauluun liittyvistä syistä se päätettiin toistaiseksi jättää pois insinööriyön käsittelemästä laajuudesta.

7 Yhteenveto

Insinööriyön pyrkimyksenä oli välittää lukijalle, kuinka automaatiota on mahdollista hyödyntää ohjelmistoasennuksien suorittamiseen sekä järjestelmän konfigurointiin Windows-käyttäjärjestelmäympäristössä. Aluksi, luvuissa 2 ja 3, esiteltiin automaation hyötyjä sekä haittoja, pohdittiin eri lähestymistapoja automaatioon ja määriteltiin ratkaistava ongelma.

Seuraavaksi luvuissa 4 ja 5 käytiin läpi työkalun suunnittelua sekä toteutusta esitellen suoritettut toimenpiteet tarkasti.

Lopuksi luvussa 6 vielä analysoitiin lopputuloksen onnistuneisuutta, vertailtiin sitä alkuperäiseen visioon, tuotiin esille kehitysvaiheessa tapahtuneita suunnitelmanmuutoksia ja määriteltiin työkalun puutteet sekä toimenpiteet jatkokehitykselle.

Insinööriyön aikana yllätti, että hyvin monet sovellusasennukset tarjoavat hiltajaista asennusvaihtoehtoa visuaalisen käyttöliittymän sijaan.

Asennusautomaatiotyökalua luotaessa tuli myös hyvin esille ohjelmistoversioiden julkaisuprosessiin liittyviä hankaluuksia. Ohjelmistoversioiden julkaisun tulisi olla standardoitu prosessi, sekin itsessään kaipaa vielä selvästi jatkokehitystä automaatiotyökalua paranneltaessa.

CI/CD-menetelmän käyttöönottoa olisi hyvä vakavasti pohtia ja ohjelmistotuotantoprosesseja mukauttaa siihen sopiviksi. Täytyy myös perehtyä syvemmin sekä Dockerin tarjoamiin kontteihin että Chocolateyn tai vastaavien paketinhallintatyökalujen käyttöönottoa.

Lähteet

- 1 Elematic-yrityssivut. Verkkoaineisto. Elematic. <<https://www.elematic.com/>> Luettu 22.4.2022.
- 2 Dotnet-tuotesivu. Verkkoaineisto. Microsoft. <<https://dotnet.microsoft.com/en-us/>> Luettu 6.5.2022.
- 3 PowerShell-dokumentaatio. Verkkoaineisto. Microsoft. <<https://docs.microsoft.com/fi-fi/powershell/scripting/how-to-use-docs?view=powershell-5.1>> Luettu 26.4.2022.
- 4 Docker-tuotesivu. Verkkoaineisto. Docker. <<https://www.docker.com/>> Luettu 7.5.2022.
- 5 Chocolatey-käyttöliittymä. Hongkiat. <<https://assets.hongkiat.com/uploads/batch-install-batch-update-windows-apps/chocolatey-gui-package-manager-windows.jpg>> Luettu 8.5.2022.
- 6 Chocolatey-tuotesivu. Verkkoaineisto. Chocolatey Software. <https://chocolatey.org/> Luettu 8.5.2022.
- 7 Difference between cmd vs PowerShell vs bash. Verkkoaineisto. ServerTribe. <<https://www.servertribe.com/difference-between-cmd-vs-powershell-vs-bash/>> Luettu 6.5.2022.
- 8 Continuous integration vs delivery vs deployment. Verkkoaineisto. Atlassian. <<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>> Luettu 7.5.2022.
- 9 Erot eri CI/CD-termien välillä. Atlassian. Ladattava tiedosto. <<https://waccdn.atlassian.com/dam/jcr:b2a6d1a7-1a60-4c77-aa30-f3eb675d6ad6/ci%20cd%20asset%20updates%20.007.png?cdnVersion=336>> Luettu 7.5.2022.
- 10 Apex-tuotesivu. Verkkoaineisto. Oracle. <<https://apex.oracle.com/en/>> Luettu 30.4.2022.
- 11 Source or dot operator. Verkkoaineisto. SS64. <<https://ss64.com/ps/source.html>> Luettu 9.5.2022.

- 12 Install SQL Server from the command prompt. Verkkoaineisto. Microsoft. <<https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server-from-the-command-prompt?view=sql-server-ver15>> Luettu 29.4.2022.
- 13 Installing and configuring Oracle database using response files. Verkkoaineisto. Oracle. <https://docs.oracle.com/cd/B19306_01/install.102/b15660/app_nonint.htm> Luettu 29.4.2022.

Asennusskriptejä kutsuva pääskripti

```
. ./server_settings.ps1

# Tools
& ./install_programs.ps1

# Oracle DB
$confirmation = Read-Host 'Install Oracle DB? y/[n]: '
if ($confirmation -eq 'y') {
    & ./install_oracle.ps1
} else {
    Write-Host 'Skipping Oracle DB installation.'
}

# ERP
$confirmation = Read-Host 'Install ERP? y/[n]: '
if ($confirmation -eq 'y') {
    & ./install_java.ps1
    & ./install_apex.ps1
    & ./install_mobile.ps1
    & ./install_nginx.ps1
    & ./install_ssrs.ps1
    & ./install_barcodes.ps1
    & ./install_task_scheduler_scripts.ps1
} else {
    Write-Host 'Skipping ERP installation.'
}

# MES1
$confirmation = Read-Host 'Install MES1? y/[n]: '
if ($confirmation -eq 'y') {
    & ./install_sqlserver.ps1
    & ./install_mes1.ps1
    & ./install_oracle.ps1
} else {
    Write-Host 'Skipping MES1 installation.'
}

# MES2
$confirmation = Read-Host 'Install MES2? y/[n]: '
if ($confirmation -eq 'y') {
    & ./install_mes2.ps1
} else {
    Write-Host 'Skipping MES2 installation.'
}
```

```
# Open firewall ports
$confirmation = Read-Host 'Open firewall ports' $firewall-
Ports '? y[n]: '
if ($confirmation -eq 'y') {
    & ./set_firewall_rules.ps1
} else {
    Write-Host 'Skipping firewall rule additions.'
}
```