

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Arnika Rästa 175928IDDR

# **Andmete migratsiooniprotsessi automatiseerimine AS-i Synerall näitel**

Diplomitöö

Juhendaja: Priit Rospel  
MSc

Tallinn 2021

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Arnika Rästa

17.05.2021

## **Annotatsioon**

Probleemideks on andmemigratsiooni komponentide (SSIS pakett ja migratsioonirakendus) käsitsi käivitamine ja käsitsi uuendamine mitmetes erinevates serverites. Selle tagamiseks peab olema töötajatel ligipääs serveritesse. Tegemist on manuaalse protsessiga, mis võtab töötajatel aega ja põhjustab inimvigu. Inimvigadeks on komponentide käivitamise unustamine ja komponentide uuendamisel mõne serveri unustamine.

Töö eesmärgiks on automatiseerida komponentide uuendamist ja migratsiooniprotsessi tööd nii, et töötajad ei pea uute versioonide paigaldamisega tegelema ega komponente käsitsi käivitama.

Töö tulemuseks toimub komponentide uuenduste paigaldamine loodud konveierite poolt ning komponentide käivitamiseks kasutatakse serverisse lisatud töid. Komponentidele lisatakse andmete olemasolu ja valmiduse kontrollid. Säätetud tööaeg suunatakse tootearendusse, ning inimvigadest tekkinud olukordade lahendamisele ei pea kulutama töötajad lisa-aega.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 9 peatükki, 5 joonist, 1 tabelit.

## **Abstract**

### **Data Migration Process Automation Based on AS Synerall Example**

The problems handled in the thesis are – manual activation of data migration components (SSIS package and migration application) and deployment of updates into several target servers. This requires the employees to have access to all target servers. Manual activation wastes employees work time and causes human errors. For example, in some servers components are not up to date and migrations are delayed because of employee forgetfulness.

The goal of this thesis is to automate migration component updates and migration process from start to end. Thus, reducing the risk of human error and reducing employees work time related to updating components and executing data migration process.

As a result of the work component updates are performed by build and deploy pipelines. Migration components are activated by tasks in servers. The overall migration process starts when files are provided and continues automatically if critical errors are not found. Spare work time can be directed for product development work. There is no need to spend extra time on correcting human errors.

The thesis is in Estonian and contains 31 pages of text, 9 chapters, 5 figures, 1 tables.

## Lühendite ja mõistete sõnastik

|                     |   |
|---------------------|---|
| Andmevoog           | <i>Data flow</i> – programmi käskude täitmise jada  |
| CIS                 | <i>Customer Information System</i> – kliendi andmete haldamise tarkvara   |
| CSV                 | <i>Comma Separated Values</i> – komaga eraldatud väärtuste fail   |
| JSON                | <i>JavaScript Object Notation</i> – inimloetava teksti failiformaat andmete edastamiseks ja hoidmiseks  |
| Konveier            | <i>Pipeline</i> – jadamisi protsesside ahel   |
| Käsuvoog            | <i>Control flow</i> – programmi andmetega seotud tegevuste jada   |
| Mestima             | <i>Merge</i> – mitme andmekogumi ühesuguseks ühendamine   |
| Pidev edastamine    | <i>Continuous Delivery</i> – koodiuuendustest pidevalt käivitatava rakenduse loomise tava   |
| Pidev integratsioon | <i>Continuous Integration</i> – versioonihalduse kasutamise tava koodi hoidmiseks ja kohaliku koodi uuendamiseks  |
| Pidev tarne         | <i>Continuous Deployment</i> – rakenduse käivitatava versiooni pidev kasutusse andmise tava   |
| Põhirakendus        | Synerall toote üks rakendustest – kliendiinfosüsteemi (CIS) ja arvelduse veebipõhine platvorm   |
| Pärandsüsteem       | Varasemalt kasutusel olnud süsteem, mida ei täiustata ega uuendata  |
| SSDT                | <i>Microsoft SQL Server Data Tools</i> – Visual Studio-ga integreeritav arendustööriist SQL Server ja Azure SQL andmebaaside ning Integration Services pakettide loomiseks    |
| SSIS                | <i>SQL Server Integration Services</i> või <i>Integration Services</i> – Microsofti SQL Server andmebaasi tarkvara komponent andmete liigutamiseks, laadimiseks ja muutmiseks |
| SSMS                | <i>SQL Server Management Studio</i> – rakendus SQL Serveri ja andmebaaside haldamiseks  |
| TFVC                | <i>Team Foundation Version Control</i> – koodi versioonihaldus Azure DevOps Server-is   |
| Võõrvõti            | <i>foreign key</i> – võti, millega määratakse seos kahe tabeli andmete vahel  |
| Ühendusstring       | <i>Connection string</i> – string ehk sõne mille abil saab andmeallikaga ühenduda   |

## Sisukord

|   |    |
|---|----|
| 1 Sissejuhatus .....  | 10 |
| 2 Syneralli olemasolev andmete migratsiooniprotsess.....                                    | 12 |
| 2.1 Synerall AS.....  | 12 |
| 2.2 Eeldused migratsiooni üles panekuks.....  | 12 |
| 2.3 Olemasoleva migratsiooni süsteemi protsess ning selle töövoo kitsaskohad.....           | 13 |
| 2.4 Komponentide uuendamise kitsaskohad .....   | 16 |
| 2.5 Migratsiooni tulemus kliendile.....   | 17 |
| 3 Teoreetiline alus .....   | 18 |
| 3.1 CI/CD ehk Continuous Integration, Continuous Delivery ja Continuous<br>Deployment ..... | 18 |
| 3.1.1 Pidev integratsioon ehk Continuous Integration .....                                  | 18 |
| 3.1.2 Pidev edastamine ehk Continuous Delivery.....   | 19 |
| 3.1.3 Pidev tarne ehk Continuous Deployment.....  | 19 |
| 3.2 Azure DevOps Server.....  | 19 |
| 3.3 SQL Server Integration Services .....   | 20 |
| 3.4 SSIS projekti kokku ehitamise ja paigaldamise võimalused.....                           | 21 |
| 3.4.1 SQL Server Data Tools (SSDT).....   | 21 |
| 3.4.2 SSIS DevOps Tools.....  | 22 |
| 3.5 SSIS paketi käivitamise võimalused.....   | 22 |
| 3.5.1 SQL Server Agent .....  | 23 |
| 3.5.2 Task Scheduler .....  | 24 |
| 4 Analüüs.....  | 25 |
| 4.1 Nõuded lahendusele.....   | 25 |
| 4.2 Migratsioonirakenduse automaatse uuendamise lahendus .....                              | 25 |
| 4.3 SSIS projekti automaatse uuendamise lahenduse valik.....                                | 26 |
| 4.4 SSIS paketi käivitamise lahenduse valik .....   | 27 |
| 5 Projekti planeerimine.....  | 29 |
| 5.1 SSIS konveierite loomine ja seadistamine.....   | 29 |

|  |    |
|--|----|
| 5.2 SSIS paketi automaatne käivitamine ning andmefailide olemasolu kontrolli lisamine.....             | 29 |
| 5.3 Migratsioonirakenduse uuenduste automatiseerimine ja automaatne käivitamine .....                  | 31 |
| 6 Teostus.....   | 34 |
| 6.1 SSIS projekti viimine versioonihaldusesse ja konveierite loomine automaatseks uuendamiseks.....    | 34 |
| 6.2 SSIS failide olemasolu kontrolli ja automaatne käivitamine.....                                    | 35 |
| 6.3 Migratsioonirakenduse konveierite loomine automaatseks uuendamiseks .....                          | 36 |
| 6.4 Migratsioonirakenduse täiendamine ja automaatne käivitamine.....                                   | 36 |
| 7 Tulemus .....  | 37 |
| 8 Tulevikuplaanid.....   | 38 |
| 9 Kokkuvõte .....  | 39 |
| Kasutatud kirjandus .....  | 41 |
| Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks ..... | 43 |
| Lisa 2 – Paigaldamise konveieri keskkondade järjekorra seadistamine testkeskkonnaga .....              | 44 |
| Lisa 3 – SQL Server Agent töö loomisel SSIS paketi käivitamise tööülesande lisamine SSMS-i kaudu.....  | 45 |

## **Jooniste loetelu**

|   |    |
|---|----|
| Joonis 1. Migratsiooniprotsessi andmevoo diagramm. ....                   | 13 |
| Joonis 2. Migratsiooni esimese etapi SSIS protsessi voodiagramm.....      | 15 |
| Joonis 3. Migratsiooni teise etapi protsessi voodiagramm. ....            | 16 |
| Joonis 4. Uuendatud migratsiooni esimese etapi protsessi voodiagramm..... | 31 |
| Joonis 5. Uuendatud migratsiooni teise etapi protsessi voodiagramm.....   | 33 |



## **Tabelite loetelu**

|  |    |
|--|----|
| Tabel 1. SSIS paigaldamise vahendite võrdlus. .... | 27 |
|--|----|

## 1 Sissejuhatus

Synerall AS pakub kommunaalteenuste ettevõtetele toodet nimega Synerall, milleks on kliendiinfosüsteem ja arvelduse platvorm koos iseteenindusportaaliga. Toote kasutusse võtmiseks on testfaasis vajalik teostada kliendi andmete migratsioon ehk andmete üle viimine Syneralli andmebaasi. Klient võtab andmed eelnevalt kasutusel olevast süsteemist. Andmemigratsioon võimaldab kliendil olemasolevaid andmeid Synerallis kasutada ja nendega edasi töötada.

Andmemigratsiooni komponentideks on SSIS pakett ja .NET konsoolirakendus ehk migratsioonirakendus. Migratsiooniprotsess jaguneb kaheks osaks. Esimeses osas sisestatakse kliendi andmed migratsiooni andmebaasi SSIS paketi või kliendi enda poolt. Teises osas võtab migratsioonirakendus andmed migratsioonibaasist, kontrollib andmete vastavust ärioloogikale ning sisestab need põhirakenduse andmebaasi.

Andmemigratsiooniga on seotud kaks suuremat probleemi: migratsiooniprotsessis on palju manuaalseid tegevusi ja kõikide migratsiooni komponentide uuendamine toimub käsitsi. Need tekitavad olukordi, kus inimvea tõttu on serverites kasutusel erinevad versioonid või protsess ei viida lõpuni, kuna komponentide käivitamine ununeb. Lisaks ei ole võimalik migratsioonirakendust uuendada ilma serverisse ligipääsuta.

Töö eesmärgiks on lihtsustada Syneralli arendajate tööprotsesse ja vähendada migratsiooni haldusele kuluvat aega. Migratsiooniprotsessi automatiseeritakse nii, et Syneralli töötaja ei pea migratsiooni komponente käsitsi käivitama. Arendajate tehtud muudatuste kasutusse võtmiseks ei ole vajadust kõikidesse serveritesse käsitsi eraldi paigaldamist teha.

Migratsiooniprotsessi manuaalseteks tegevusteks on Syneralli töötaja poolt SSIS paketi ja migratsioonirakenduse eraldi käsitsi käivitamine. Selleks on vajalik töötaja ligipääs kõikide migreeritavate klientide serveritesse. Migratsioonirakendus on komponent, mida uuendatakse arendaja poolt rakenduse binaarfailide asendamisega igas serveris. SSIS paketti uuendatakse Visual Studio paigaldamise viisardi kaudu igale serverile eraldi.

Lahenduse valimisel peab arvestama, et klientide serverites kasutatakse Windows Server 2016 või uuemat operatsioonisüsteemi ning Microsoft SQL Server 2016 tööriista või selle uuemat versioon.

Töö käigus selgitatakse välja, mis põhjustab protsesside manuaalse käivitamise ning uuenduste käsitsi paigaldamise. Autor otsib puuduste likvideerimiseks võimalikke lahendusi ja võrdleb nende sobivust oodatava lahenduse saavutamiseks.

Töö koosneb seitsmest osast. Kõigepealt kirjeldatakse olemasoleva andmemigratsiooni olukorda ja probleeme. Seejärel tutvustatakse erinevaid lahendusi. Analüüsis võrreldakse lahendusi üksteisega ja nende sobivust probleemide lahendamiseks. Peale analüüsis lahenduste välja valimist koostatakse plaan, kuidas neid lahendusi kasutusse võtma hakatakse. Planeerimise järgselt kirjeldatakse realiseeritud teostust ning selle tulemusi. Lõpus tuuakse välja tulevikuplaanid, kuidas migratsiooniprotsessi edasi arendada.

## **2 Syneralli olemasolev andmete migratsiooniprotsess**

Käesolevas osas tutvustatakse Syneralli andmemigratsiooni olemasolevat lahendust. Tuuakse välja mis komponentidest see koosneb, kuidas protsess töötab ning millised on edaspidi lahendatavad probleemid ja puudujäägid.

### **2.1 Synerall AS**

Synerall AS pakub energiaettevõtetele loodud toodet nimega Synerall, mis koosneb kahest rakendusest. Esimeseks rakenduseks on veebipõhine kliendiinfosüsteemi (CIS) ja arvelduse platvorm, mida edaspidi nimetatakse põhirakenduseks. Teiseks rakenduseks on veebipõhine iseteenindusportaal. Toote kasutajateks on elektrik, gaasi või kaugkütet pakkuvad kommunaalteenuste ettevõtted. Toode on eelkõige mõeldud Põhjamaade turule, hetkel on Synerall kasutusel Eestis, Soomes ja Inglismaal. Enamik kliente on Soomes [1].

Arendajad kasutavad töö tegemiseks Microsofti Visual Studio (versioon 2017 ja 2019) integreeritud programmeerimiskeskonda. Toote koodi ja versioonide haldamiseks kasutatakse Azure DevOps Server-it, endine Team Foundation Server. Syneralli kliendiinfosüsteem ja iseteenindusportaal on .NET rakendused. Klientide serverites on Windows Server 2016 või uuem operatsioonisüsteem [2].

Käesolevas diplomitöös ei käsitleta Synerall toote põhilisi tööprotsesse. Keskendutakse Syneralli klientide andmete migreerimise töövoogudele ja migratsiooni tulemuste jälgimisele.

### **2.2 Eeldused migratsiooni üles panekuks**

Andmete migratsioon tähendab kliendi andmete liigutamist pärandsisüsteemi andmebaasist põhirakenduse andmebaasi. Migreeritavateks andmeteks on eelnevalt kasutatavast süsteemist kasutajad, kontaktandmed, mõõtepunktid, mõõtjad, lepingud, hinnakirjad ja arveldusinfo. Andmemigratsioone teostatakse pidevalt testkeskkonnas, et parandada

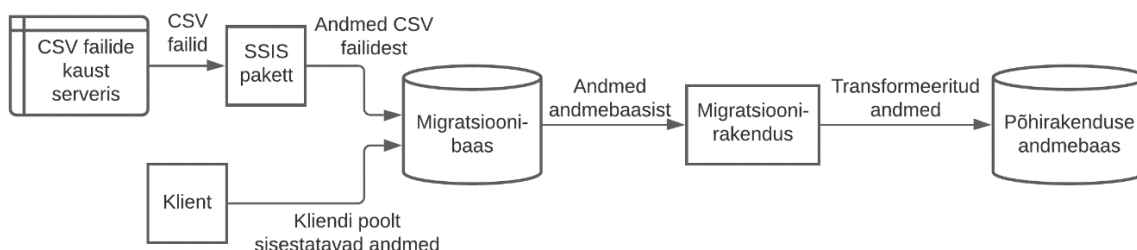
kliendi poolt ette valmistatud sisendandmeid. Migratsiooni käigus kontrollitakse andmete vastavust andmebaasi väljade andmetüüpidele ja Synerall rakenduse äriloogikale. Lõpptulemusena on andmed kantud rakenduse andmebaasi ja klient saab neid kasutama hakata [3].

Syneralli kasutamiseks on kliendil eraldi serverid andmebaasi ja põhirakenduse jaoks. Iseteenindusportaal võib asuda, kas eraldiseisvas serveris või põhirakendusega samas serveris. Käesolevas töös käsitletakse andmebaasi ja põhirakenduse serverit. Andmebaasi serveris asub migratsioonibaas ja SSIS (*SQL Server Integration Services*) pakett kliendi andmete liigutamiseks migratsioonibaasi. Põhirakenduse serveris asub Synerall toode ning migratsioonirakendus. Migratsioonirakendusega kontrollitakse andmete vastavust äriloogikale. Kontrollid läbinud andmed sisestatakse põhirakenduse andmebaasi [4].

Syneralli andmebaasi serverisse on paigaldatud Microsoft SQL Server 2016 või sellest uuem versioon ja andmete migratsiooni jaoks SSIS ning SSDT (*Microsoft SQL Server Data Tools*). On loodud tühi põhirakenduse andmebaas koos andmebaasi struktuuri ja vajalike andmebaasi kasutajatega [4].

## 2.3 Olemasoleva migratsiooni süsteemi protsess ning selle töövoo kitsaskohad

Andmemigratsiooni eelduseks on tühi migratsiooni andmebaas, millel on teistsugune andmestruktuur kui põhirakenduse baasil. Arendaja loob selle olemasolevat SQL skripti käivitades. Migratsioonirakenduseks on .NET konsoolirakendus, mis kontrollib migratsioonibaasis olevate andmete vastavust äriloogika reeglitele ja sisestab need andmed põhirakenduse andmebaasi (vt Joonis 1, lk 13) [2].



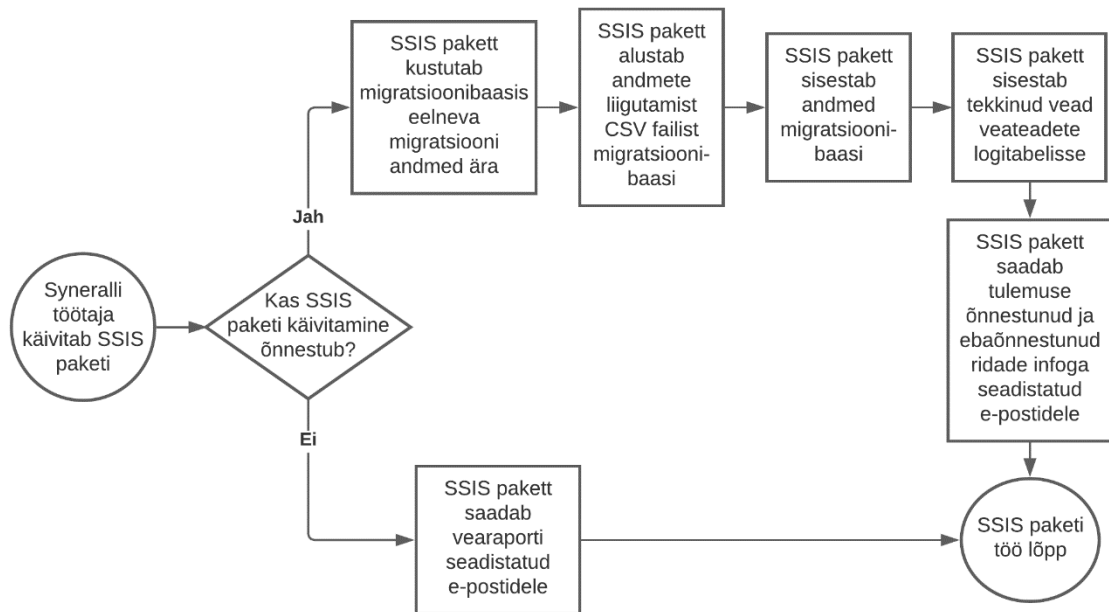
Joonis 1. Migratsiooniprotsessi andmevoo diagramm.

Migratsioon koosneb kahest etapist – esimene on andmete sisestamine migratsiooni andmebaasi ja teine on migratsiooni andmebaasist andmete üle viimine pöhirakenduse andmebaasi [3].

Esimest etappi on võimalik teostada kahel viisil:

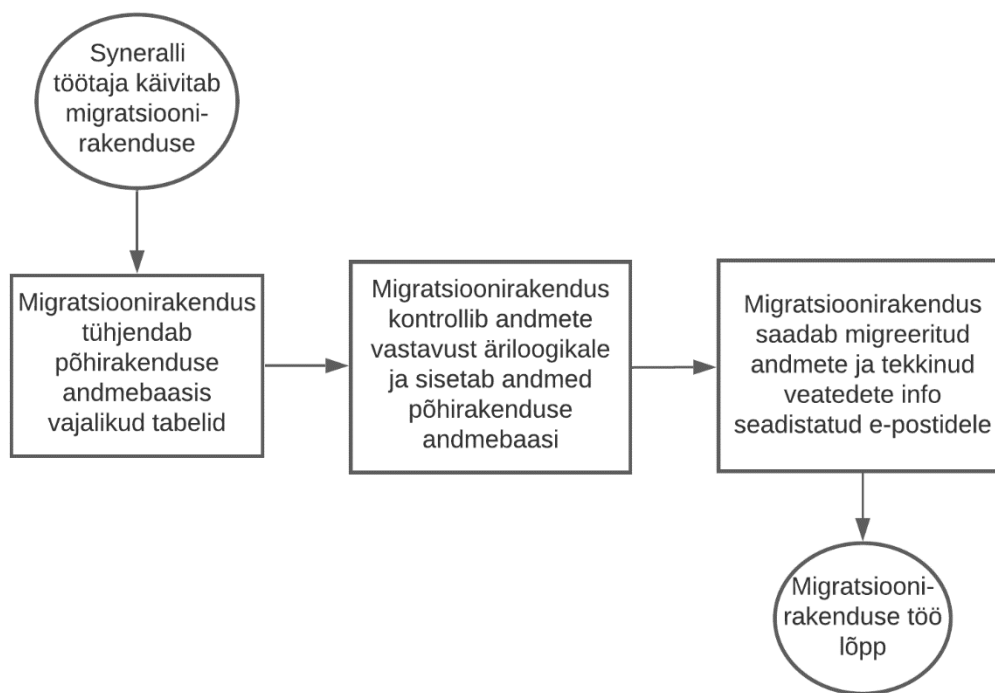
- Kliendid liigutavad enda valitud meetodil andmed pärandsüsteemist migratsiooni andmebaasi. Kliendile tutvustatakse migratsiooni andmebaasi ülesehitust.
- Klientidele antakse dokumentatsioon mille alusel CSV (*Comma Separated Values*) faile täidetakse. Kliendid ekspordivad pärandsüsteemi andmed failidesse ja panevad need serveri kausta. SSIS pakett võtab kaustas olevad failid ja sisestab need migratsioonibaasi [3].

Kõik Syneralli kliendid on kasutanud teist varianti ehk CSV failidega andmete üle toomist. Iga migratsioonibaasi tabeli kohta on üks CSV fail. Failis on esimeseks reaks tabeli veergude nimed ja järgnevateks ridadeks on tabeli andmed. Väärtused on failis eraldatud semikoolonitega. CSV failide olemasolul käivitab Syneralli töötaja käsitsi andmebaasi kataloogis asuva SSIS paketi, mis liigutab andmed failidest andmebaasi (vt Joonis 2 lk 15). Andmete töötlemisel kontrollitakse andmete vastavust andmebaasi struktuurile. SSIS paketi töö käigus tekkivad vead salvestatakse migratsioonibaasi veateadete tabelisse, ning saadetakse määratud Syneralli töötajatele ja kliendile e-kirjaga. Ainult kontrolli läbinud andmed sisestatakse SSIS-i poolt migratsiooni andmebaasi [3].



Joonis 2. Migratsiooni esimese etapi SSIS protsessi voodiagramm.

Migratsiooni järgmiseks etapiks käivitab Syneralli töötaja migratsioonirakenduse peale SSIS paketi töö lõppu (vt Joonis 3 lk 16). Migratsioonirakenduse käivitamisel kustutatakse eelmise migratsiooni andmed põhirakenduse andmebaasis. Tühjendatavad tabelid on seadistatud migratsioonirakenduses. Rakendus kontrollib migreeritavate andmete vastavust äriloogikale ning normaliseerib andmeid (näiteks aadressid). Kliendile ei saadeta kokkuvõtet migratsioonirakenduse töö tulemusest, kuna veakirjeldused on liiga tehnilised. Tulemus saadetakse seadistatud Syneralli töötajatele e-kirjana [3].



Joonis 3. Migratsiooni teise etapi protsessi voodiagramm.

## 2.4 Komponentide uuendamise kitsaskohad

Syneralli töötajate jaoks on migratsiooni kõige probleemsemaks kohaks komponentide uuenduste käsitsi paigaldamine igasse serverisse. SSIS paketti uuendatakse iga migratsioonibaasi muudatuse puhul. Migratsioonirakendust uuendatakse tihedamini. Uuenduste paigaldamine nõuab arendaja aega, ligipääsu kõikidesse serveritesse ning tekitab olukordi, kus kõikides serverites ei kasutata samasuguseid versioone [5].

Migratsioonirakenduse kood asub versioonihalduses koos põhirakendusega. Migratsioonirakenduse serverisse lisamiseks tõstab arendaja programmi failid käsitsi serveri kausta. Sama tehakse rakenduse uuendamisel. Uuendamiseks asendab arendaja kliendi serveris olevad migratsioonirakenduse binaarfailid arendaja kohalikus arvutis olevate uuendatud binaarfailidega. Iga kliendi serveri jaoks peab seda eraldi tegema [2].

SSIS paketi kõige uuem versioon asub kindla serveri kaustas, versioonihaldusesse seda lisatud ei ole. SSIS paketi üles seadmine ja uue versiooni paigaldamine käib samuti käsitsi. Iga migratsiooni andmebaasi struktuuri muutus vajab SSIS paketi muutmist, millega tegeleb üks konkreetne töötaja. Arendaja annab talle muudatuste info edasi peale igit andmebaasi struktuuri muutust. Seejärel muudab SSIS paketiga tegelev töötaja



paketti vastavalt infole ja paigaldab uue versiooni Visual Studio viisardi kaudu vajalikesse andmebaasidesse [2].

## **2.5 Migratsiooni tulemus kliendile**

Kliendile saadetakse andmemigratsiooni esimese osa tulemused automaatselt SSIS paketi töö lõpus. Klient saab teada migreeritud tabelite loetelu, edukate ja ebaõnnestunud ridade arvu ning andmete sisestamisel tekkinud vigade nimekirja. Vigadeks on näiteks andmetüübi mittedobivus, kohustusliku välja väärtuse puudumine, võõrvõtme (ingl *foreign key*) viidatava teise tabeli kirje puudumine ja duplikaatväärtuse või duplikaatrea olemasolu [3].

Kliendid on varasemalt väljendanud soovi näha migratsiooni andmekvaliteedi ajalugu. Selline võimalus puudub hetkel, kuna migratsiooni lõpptulemusi ei talletata kusagile. Lõplikud migratsiooni teise osa tulemused, mis koosnevad tehnilistest veateadetest, saadetakse ainult e-posti teel Syneralli töötajatele. Syneralli tarkvaraarhitekt koostab käsitsi umbkaudse andmekvaliteedi ülevaate statistika, põhinedes e-kirjades olevale infole ja saadab selle klientidele. See on ajakulukas, ehamugav ja veaohklik. Klientide soov oleks kohe peale andmemigratsiooni näha lõpptulemust [5].

## 3 Teoreetiline alus

Käesolevas osas tuuakse välja erinevad probleemide lahendused. Tutvustatakse lähemalt lahendustes kasutatavaid tööriistu ja rakendusi ning nende kasutusvõimalusi.

### 3.1 CI/CD ehk Continuous Integration, Continuous Delivery ja Continuous Deployment

Komponentide automaatseks halduseks kasutatakse laialt levinud tavasid. Nende koondnimetus on CI/CD, mis koosneb kolmest tavast: pidev integratsioon (ingl k *Continuous Integration*), pidev edastamine (ingl k *Continuous Delivery*) ja pidev tarne (ingl k *Continuous Deployment*) [6].

#### 3.1.1 Pidev integratsioon ehk Continuous Integration

Ühtse koodibaasi peal töötavate arendajate puhul võib tekkida olukord, kus mitu inimest muudavad samu koodiridu. Esimene neist lõpetab oma töö ära ja viib oma muudatused ühtsesse koodibaasi – teised töötavad oma kohaliku versiooni peal edasi ilma kõige uuemate muudatusteta. Konflikt tekib, kui mõlema muudatused on seotud sama koodiga. Nii võivad eelnevad muudatused minna kaduma ja ühe poole muudetud funktsionaalsus ei tööta enam õigesti või rakendust ei ole enam võimalik kokku ehitada [6].

Pidev integratsioon on tava, kus kontrollitakse koodimuudatuste kokku sobimist rakenduse kõige uuema versiooniga ja muudetud koodi kokku ehitamise õnnestumist. Seda lahendatakse versioonihaldusega – rakenduse põhiharu koodi hoitakse ühtses kohas. Põhiharusse mestitakse (ingl k *merge*) kõikide arendajate muudatused. Arendaja uuendab kohalikku koodi põhiharust pidevalt – seda peaks tegema nii tööülesande vahetamisel kui ka keset pikemat pooleli olevat tööd [6].

Kohalike koodimuudatuste põhiharusse üles laadimisel kontrollitakse, et mestitavad muudatused ei läheks konflikti põhiharus oleva koodiga. Arendaja saab kohest tagasisidet koodi mitesobivuse kohta ning peab need vead likvideerima. Lisaks kontrollitakse uue koodi kokku ehitamise õnnestumist. Ebaõnnestumise puhul antakse veateade ja järgmisi toiminguid ei käivitata kuni viga parandatakse [6], [7].

### 3.1.2 Pidev edastamine ehk Continuous Delivery

Koodimuudatuste üles laadimine ja nende kontrollimine ei tähenda seda, et neid muudatusi on võimalik realselt kasutada. Kasutusse andmiseks peab kontrollima, et rakendus on kasutuskõlblik ning siis saab koostada sellest uue versiooni [6].

Pidev edastamine tähendab, et põhiharu uuendatud koodist luuakse uus rakenduse paigaldatav versioon. Selle jaoks seadistatakse konveier (ingl k *pipeline*), kus rakendus kompileeritakse, käivitatakse erinevat tüüpi teste (n. komponendi-, integratsiooni- ja kasutajaliidesetestid) ja pakitakse rakendus kokku. Lõpptulemuseks on rakendusest uus versioon, mida saab kasutusse anda või ebaõnnestumise korral veareport ning testide tulemused [6].

### 3.1.3 Pidev tarne ehk Continuous Deployment

Peale kasutusele võetava versiooni loomist ei ole see kohe lõppkasutajatele kättesaadav. Selleks peab rakendusest loodud uue versiooni paigaldama keskkonda, kus rakendus on kasutusel [6].

Pideva tarne puhul toimetatakse rakenduse uus versioon lõppkasutajale kasutamiseks. Rakendust võib paigaldada mitmesse keskkonda ja iga keskkonna puhul saab seadistada paigalduse aega. Keskkondadel saab määrata paigaldamise järjekorra. Näiteks üks keskkond on testkeskkond kuhu paigaldatakse rakendus esimesena ja alles selle õnnestumise korral alustatakse paigaldamist järgnevasse keskkondadesse. Sellega tuvastatakse rakenduse keskkonna ühilduvusest tulenevaid probleeme [8].

## 3.2 Azure DevOps Server

Azure DevOps Server, endine Team Foundation Server, on üks CI/CD lahendus. See katab järgnevad CI/CD vajadused: ühine koodihaldus, muudatuste kontrollimine, testimine, paigaldatava versiooni loomine ja selle kasutusse andmine. Lisaks on Azure DevOps Server-il integratsioon Visual Studio rakendustega [9].

Koodihalduseks on kaks võimalust – TFVC (*Team Foundation Version Control*) ja Git. TFVC on tsentraliseeritud versioonihaldus, ehk rakenduse kood koos versioonidega asub ühes ühtses serveris. Selle kaudu uuendavad kõik arendajad oma kohalikke muudetavaid versioone. Git on hajus versioonihaldus, ehk kood asub ühtses serveris, aga sellest saavad

kõik arendajad luua enda koodihoidla kohalikus arvutis, mis ei vaja ühtse serveri ühendust. Sellisel juhul asuvad kõik versioonid ja kogu versioonide ajalugu kohalikus arvutis. Arendaja uuendab tihedamini kohalikku koodihoidlat ja mingi aja tagant mestib muudatused ühtsesse serverisse. Synerall põhirakenduse ja migratsioonirakenduse projektid asuvad juba TFVC-s [10].

Azure DevOps Server-is on võimalus kasutada konveiereid rakenduse uuendamiseks. Konveierid jagunevad suures-joones kaheks: kokku ehitamise konveierid ja paigaldamise konveierid [11].

Kokku ehitamise konveierit saab käivitada peale koodimuudatuste tegemist käsitsi või seadistada automaatselt käivituma versioonihalduselt tuleva muudatuse märguande peale. Selle käigus ehitatakse rakendus kokku, jooksutatakse testid ning luuakse paigaldatav rakenduse versioon [11].

Rakenduse paigaldamise konveier võtab käivitatava rakenduse, jooksub olemasolevaid teste ja paigaldab toote uue versiooni seadistatud keskkonda. Lisaks saab määrata kuna täpsemalt paigaldamine teostatakse ja mis kasutajatel on õigus paigaldamist alustada. Paigaldamist saab seadistada iga keskkonna jaoks eraldi: peale muudatuse kokku ehitamist, fikseeritud intervalli tagant või ainult manuaalselt. Konveieri käivitamise ajaloost on värviliste kastidega näha, mis keskkonda konkreetne versioon paigaldati [11].

Mõlema konveieri jaoks saab seadistada erinevaid töid, millega rakendust ehitatakse kokku ja testitakse. Tööde seast valitakse sobivaim olenevalt hallatavast rakendusest ja testimiseks kasutatavast raamistikust [11].

### **3.3 SQL Server Integration Services**

SSIS (*SQL Server Integration Services*) on Microsofti SQL Server andmebaasi tarkvara komponent. Seda kasutatakse andmete välja eraldamiseks, laadimiseks ja muutmiseks. Näiteks on võimalik andmeid võtta erinevatest failidest ja viia andmed ühte või mitmesse sihtpunkti – selleks võib olla andmebaas või uus fail [12].

Ühes SSIS lahenduses saab olla mitu projekti, mille sees asuvad paketid. Lahenduse loomiseks ja muutmiseks on võimalik kasutada Visual Studio lisaprogrammi SSDT. SSIS

projekti asukohaks võib olla failisüsteem või andmebaasi SSIS kataloog. Kataloogi puhul saab projekti vaadata ja seadistada SSMS (*SQL Server Management Studio*) rakendusega [12].

SSIS paketid koosnevad kolmest osast: käsuvoog (ingl k *control flow*), andmevoog (ingl k *data flow*) ja ühendustest. Käsuvoog ei tegele andmetega, sellega määratakse ära mis tööd, kuna ja mis järjekorras käivitatakse. Andmevoog määrab kust kuhu andmed liiguvad ning kuidas andmeid töödeldakse ja muudetakse. Ühenduse jaoks määratakse ühendusstring (ingl k *connection string*), mille kaudu saavad kõik tööd andmetele ligi. See ühendab paketi andmeallikaga, milleks võib olla andmebaas, XML fail või mõni tekstifail nagu CSV [13].

Käsuvoole saab lisada erinevaid tüüpi töid ja neid paigutada gruppidesse konteineritesse ning tsüklitesse. Andmetega tegelev SSIS paketil peab olema käsuvoos vähemalt üks andmevoogu tüüpi töö, millele lisatakse andmevoogu andmete liigutamise ja töötlemise kohta. Tööde sisuks võib veel olla päringud, skriptid või funktsioonid. SSIS paketi käivitamisel pannakse käsuvoogu tööd seadistatud järjekorras käima [13].

### **3.4 SSIS projekti kokku ehitamise ja paigaldamise võimalused**

Järgnevalt uuritakse SSIS projekti uuenduste paigaldamise võimalusi ja kuidas neid kasutada. Vaadeldavateks lahendusteks on SSDT ja SSIS DevOps Tools.

#### **3.4.1 SQL Server Data Tools (SSDT)**

Visual Studio-ga integreeritav tööriist SQL Server Data Tools (SSDT) võimaldab SSIS projekte muuta ja paigaldada. Paigaldamiseks ei ole vaja eraldi õiguseid, seda saab teha igäiks kellel on SSIS projekt ja sihtkoha andmed. Paigaldamise eelduseks on eelnevalt loodud ja seadistatud sihtkoha kataloog kuhu projekti paketid paigaldatakse [14].

Visual Studios saab projektile seadistada parameetreid *project.params* failis, ülejäänud seadistamine tuleb teha SSMS rakenduse kaudu andmebaasi kataloogis. Projekti paigaldamine käib Visual Studios teenuse paigaldamise viisardi kaudu. Viisardis peab sisestama sihtkoha ning autentimise viisi [14].

### 3.4.2 SSIS DevOps Tools

Azure DevOps Serveri mõlema konveieri jaoks on võimalik seadistada töid SSIS projektide jaoks. Tööde kasutamiseks on vajalik Azure DevOps Marketplace-ist paigaldada SSIS DevOps Tools laiend. Antud laiend on tasuta ning sellega on võimalik luua SSIS kokku ehitamise, paigaldamise ja kataloogi seadistamise töid. Selle jaoks peab SSIS projekt asuma versioonihalduses [15].

Kokku ehitamise konveierile saab lisada ainult ühte SSIS kokku ehitamise tööd. Seda kasutatakse ühe või mitme SSIS projekti kokku ehitamiseks projektifailidest. Kokku ehitamise töö jaoks on vaja seadistada projekti asukoht. Selle järgi otsitakse üles projektifail, mida kasutatakse kokku ehitamisel. Tulemuseks on ispac faililaiendiga kokku ehitatud fail, mida kasutatakse paigaldamiseks [15].

Rakenduse paigaldamise konveierile saab lisada SSIS paigaldamise tööd ja kataloogi seadistamise tööd. Paigaldamise töö võtab sisendiks kokku ehitatud ühe või mitu projekti ja paigaldab need soovitud failikataloogi või andmebaasi. Selleks tuleb seadistada projekti paigaldatava algfaili asukoht ja sihtkoht kuhu see üles laetakse. Sihtkoha jaoks on vaja lisada sihtkoha tüüp ning server, fail või kataloog, kuhu antud projektifail salvestatakse. Sihtkoha tüüp määrab, kas paigaldamise kohaks on faili süsteem või andmebaasi SSIS kataloog. Andmebaasi kataloogi puhul tuleb lisada ka autentimise viis, milleks saab olla Windowsi kasutaja, SQL Serveri kasutaja või parooliga autentimine [15].

SSIS kataloogi seadistamise töö võtab seadete faili sisse ja rakendab need sihtkoha kaustale, projektile või keskkonnale. Seadistused on JSON (*JavaScript Object Notation*) formaadis, mis lisatakse tööle tekstina või määratakse töös faili asukoht projektis. Tööle lisatakse sihtkoha server ja autentimise viis – samamoodi nagu paigaldamise töös. Rakendatud seadistused võetakse tagasi paigaldamise ebaõnnestumisel. JSON failis määratakse seadistusi kataloogile, kaustale ja projektile [15].

## 3.5 SSIS paketi käivitamise võimalused

Selles peatükis uuritakse SSIS käivitamise võimalusi ning millistel juhtudel neid kasutada saab. Vaatluse all on SQL Server Agent ja Task Scheduler tööd.

### 3.5.1 SQL Server Agent

SQL Server Agent on SQL Server-iga kaasa tulev Windowsi teenus. Sellega saab SQL Serveri masinas luua töid, mida käivitatakse määratud intervalli tagant. Tööde lisamine ja muutmine käib SSMS rakenduse kaudu [16].

SQL Server Agent töö sisuks on ülesanded, mida täidetakse töö käivitamisel. Ülesandeks saab olla näiteks PowerShell-i skripti, SSIS paketi või Transact-SQL skripti käivitamine. Ühele tööle võib lisada mitmeid erinevaid ülesandeid ja määrata, mis järjekorras neid täidetakse. Ühe ülesande edukal lõpetamisel käivitatakse järgmine nimekirjas olev ülesanne. Ülesande ebaõnnestumise puhul katkestatakse terve töö ja raporteeritakse ebaõnnestumise põhjus [16].

Töö jaoks on vaja määrata tööde käivitamise ajakava. Ajakava seadistamiseks on neli erinevat võimalust:

1. Töö käivitatakse, kui SQL Server Agent käivitatakse. See juhtub siis kui SQL Serveri masin pannakse käima.
2. Töö käivitatakse, kui masin on jõudeolekus ja protsessori kasutus on piisavalt madal. Protsessori jõudeoleku piiri saab määrata SSMS kaudu.
3. Tööle määratakse käivitamiseks üks kuupäev koos kellaajaga. Seda tööd käivitatakse ühekordselt määratud ajal.
4. Tööle määratakse sagedus, kui tihti ja mis kellaajal käivitamine toimub. Sagedust määratakse kuude, nädalate ja päevade põhjal. Alamsagedusena saab määrata, mitu korda ühe päeva jooksul käivitamine toimub ja mis kellaaja vahemikus või mitme tunni tagant [17].

Tööle on võimalik määrata töötamise käigus tekkinud SQL Serveri poolt salvestatud Windowsi logi sündmustele reageerimine. Sündmusi saab tuvastada sündmuse kriitilisuse, sisu või veanumbri põhisel. Tingimustele sobiva sündmuse leidmisel käivitatakse seadistatud SQL Server Agent töö või teavitatakse valitud isikuid sündmuse tekkimisest [17].

SSIS paketi lõpptulemust on võimalik edastada erinevatesse kanalitesse. Edastada saab infot nii töö õnnestumise kui töö lõpetamise kohta. Mõlemal juhul on võimalik töö

seadistada iseennast kustutama töö tulemuse saavutamisel. Iga töö kohta saab vaadata käivitamiste ajalugu. Ajaloos on näha, mis ajal tööd käivitati, kui kaua töö aega võttis ja kas töö lõppes edukalt [17].

### **3.5.2 Task Scheduler**

Windows 10 operatsioonisüsteemil on sisse ehitatud tööriist Task Scheduler. Sellega saab luua automaatselt käivitatavaid tööülesandeid. Ülesannete käivitamiseks on kolm võimalust – automaatselt määratud intervalli tagant, arvuti käivitamisel või spetsiifilise logisündmuse tekkimisel. Tööülesande sisuks saab olla programmi või lisatud skripti käivitamine kindlas asukohas [18].



## 4 Analüüs

Selles osas võrreldakse erinevaid leitud lahendusviise ning kontrollitakse, kas need vastavad nõuetele. Võrdluste tulemuseks valitakse välja lahendusviisid, mida kasutatakse probleemide lahendamiseks ja tulemuste saavutamiseks.

### 4.1 Nõuded lahendusele

Synerall AS tarkvaraarhitektiga hetkeolukorra arutamise ja puuduste kokku kogumise käigus tulid esile nõudmised, millele migratsiooniprotsessi automatiseerimise lahendus peab vastama:

- Nõue 1: Uue migratsioonirakenduse halduse lahenduse kasutusele võtmine peab olema võimalikult lihtne ja kiire töötajate jaoks. Saab võtta kasutusele ilma pikema täiendkoolitusega.
- Nõue 2: Uus migratsioonirakenduse halduse lahendus ei tohi tuua suuri lisakulutusi Synerall AS firmale.
- Nõue 3: Migratsioonirakenduse uuendusi peab olema lihtne ja kiire klientide serveritesse paigaldada ja kasutusele võtta. Uuendusi ei paigaldata käsitsi serveritesse.
- Nõue 4: SSIS pakett peab automaatselt faile töötleva hakkama, ilma käsitsi käivitamiseta. Migratsiooni alustatakse siis kui klient on CSV failid kausta lisanud, ei ole vaja eraldi Syneralli tiimilt küsida migratsiooni alustamist.
- Nõue 5: SSIS projekti peab olema lihtsasti ja kiiresti uuendatav kõikide klientide serverites korraga. Serverite pakette ei pea eraldi ükshaaval uuendama.

### 4.2 Migratsioonirakenduse automaatse uuendamise lahendus

Syneralli põhirakenduse haldamiseks kasutatakse Synerallis Azure DevOps Server-it. Autor võrdleb selle vastavust nõuetele ja sama lahenduse kasutusele võtmise võimalust migratsioonirakenduse haldamiseks ja uuendamiseks

- Nõue 1: Syneralli rakenduse haldamine on juba eelnevalt selle peale üles ehitatud, mille tõttu on see töötajatele tuttav ja mugav tööriist. Ei ole vajalik õpetada uue lahenduse kasutamist.
- Nõue 2: Azure DevOps Server on Synerall AS-i poolt kasutuses juba ja selle kasutamise eest makstakse. Migratsioonirakenduse haldamise Azure DevOps Server-isse viimine ei too lisakulutusi juurde.
- Nõue 3: Azure DevOps Server täidab kogu CI/CD lahenduse kasutuselevõtu punktid. Rakenduse projekti saab hoida versioonihalduses. Kokku ehitamise konveierit kasutades luuakse pidevalt kasutusele võetavaid uusi versioone rakendusest. Paigaldamise konveieriga on võimalik uusi versioone paigaldada klientide serveritesse kasutamiseks.

CI/CD lahendus kiirendab migratsioonirakenduse muudatuste kasutusele võtmist. Rakenduse muudatuste paigaldamiseks piisab ainult konveieri käivitamisest. Iga uue serveri puhul piisab ainult konveierile uue keskkonna lisamisest ja selle seadistamisest. Lisaks kaob sellega ära vajadus paigaldada migratsioonirakendust käsitsi uue kliendi serverisse. Selle asemel on vaja lisada konveierile uus keskkond kuhu paigaldatakse ja see ära seadistada. Paigaldamise konveieril uue keskkonna jaoks uuendamise käivitamisel paigaldatakse rakendus serverisse.

### **4.3 SSIS projekti automaatse uuendamise lahenduse valik**

Selles peatükis võrreldakse SSIS projekti haldamiseks kasutatavaid vahendeid. Võrdlemisel võetakse arvesse uue keskkonna lisamiseks vajalikke tegevusi, seadistamise lihtsust ja uuenduste kasutatavasse keskkonda viimist (vt Tabel 1 lk 27).

Tabel 1. SSIS paigaldamise vahendite võrdlus.

|   | <b>SQL Server Data Tools (SSDT)</b>  | <b>SSIS DevOps Tools ja Azure DevOps Server-i konveierid</b>                          |
|---|--|---|
| <b>SSIS muudatuste paigaldamine</b>             | Visual Studios iga keskkonna jaoks eraldi paigaldamise viisardi läbimisega                       | Arendaja muudatuste versioonihaldusesse üles laadimine (muudatus käivitab konveierid) |
| <b>Seadistused</b>                              | Projekti jaoks project.params fail, ülejäänud käsitsi SSIS kataloogis                            | Kogu seadistused paigaldamise konveieris JSON kujul                                   |
| <b>Uue andmebaasi jaoks vajalikud tegevused</b> | Lisada uus SSIS kataloog.<br>Seadistada SSIS kataloog.<br>Käivitada paigaldamine Visual Studios. | Lisada uus SSIS kataloog.<br>Lisada konveierile uus keskkond.<br>Käivitada konveier.  |

Tabelis 1 olevate punktidest järeldub, et SSDT puhul on vaja rohkem tööd teha konfiguratsiooni lisamisega. SSIS DevOps Tools puhul on kogu konfiguratsioon JSON failis, seega on vaja ainult uue kataloogi loomine.

SSDT puhul peab iga kord paigaldamist käsitsi käivitama iga sihtkoha jaoks kui on vaja SSIS projekti uut versiooni paigaldada. Azure DevOps Server-i konveieril piisab ühekordsest seadistusest, mida saab vastavalt vajadusele hiljem muuta. Lisaks on Azure DevOps Server-is võimalik kontrollida, mis keskkondadesse milline versioon on paigaldatud.

Nõuet 5 täidab SSIS DevOps Tools ja Azure DevOps Server paremini, kuna paigaldamiseks on vaja muudatus üles laadida ja ülejäänud paigaldustega tegeleb konveier. SSDT ei vasta nõudele kuna iga keskkonna jaoks on vajalik eraldi paigaldamise viisard läbida ja sihtkoha andmed lisada. Seda peab iga muudatuse puhul tegema.

Eelneva võrdluse põhjal valib autor SSIS paketi haldamiseks Azure DevOps Server konveieri, mis on varasemalt valitud migratsioonirakenduse haldamiseks.

#### **4.4 SSIS paketi käivitamise lahenduse valik**

Järgnevalt võrreldakse millist lahendust saab kasutada SSIS paketi automaatseks käivitamiseks SQL Serveris.

Windows 10 peal on võimalik käivitada programme ja skripte Task Scheduler tööriistaga. SQL Server Agent on SQL Serveris algselt olemas ja sellega on võimalik luua spetsiaalselt SSIS paketi käivitamise tööülesanne.

SSIS projekt asub andmebaasi kataloogis, mitte failisüsteemis. See tähendab, et Task Scheduler-iga ei ole võimalik paketti otse käivitada, kuid paketi käivitamiseks on võimalik kasutada käivituskripti, näiteks PowerShell. SQL Server Agent-i puhul luuakse töö SSMS-i viisardi abil.

Võrdluse järgi otsustab autor, et SQL Server Agent on sobivam variant SSIS paketi käivitamiseks. Seda on lihtne kasutada – viisardis peab ainult töö andmeid täitma, skripti või mõnda muud lisakomponenti ei ole vaja luua. Lisaks käib SQL Server Agent tööde haldamine SSMS rakenduse kaudu, mida kasutatakse SSIS projektile ligi pääsemiseks.

Valitud SQL Server Agent töö ei kontrolli failide olemasolu. SSIS paketi korrektseks töötamiseks on vajalik lisada paketi algusesse failide olemasolu kontroll, mis käivitab ülejäänud paketi ainult failide olemasolul. Samuti tuleks lisada failide täielikkuse kontroll, mis välistab andmete liigutamise keset failide kausta lisamist.

Paketi käivitamine toimub SQL Server Agent töö poolt pidevalt, seega ainult failide kontrollist ei piisa. Kontrolli tulemuse kasutamiseks oleks vajalik failid ära kustutada või teise kausta ümber tõsta. Failide kustutamine ei sobi, kuna vigade tekkimisel on vaja uurida failide sisu. Sobilik variant on SSIS paketi käsuvoos lõpus andmefailide teisaldamine arhiivi kausta.

## **5 Projekti planeerimine**

Analüüsi tulemuste järgi koostatakse projekti plaan. Planeerimises määratakse komponentide uuendamise automatiseerimise ning protsessi ja komponentide käivitamise automatiseerimise viis ja tegevused. Määratakse ka migratsioonirakenduse ja SSIS paketi automatiseerimist toestavad muudatused.

### **5.1 SSIS konveierite loomine ja seadistamine**

Konveierite kasutamiseks viiakse olemasolev SSIS projekt versioonihaldusesse. Projekt lisatakse samasse versioonihaldusesse kui põhirakendus.

Kokku ehitamise ja paigaldamise konveier luuakse Azure DevOps Server-is. Esimese sammuna tuleb luua ja seadistada kokku ehitamise konveier, mille sisendiks on SSIS projekti uuenduse signaal versioonihaldusest.

Järgmiseks tuleb luua ja seadistada paigaldamise konveier, mille sisendiks on kokku ehitamise konveieri tulemus. Paigaldamise konveier viib rakenduse uue versiooni lisatud serveritesse. Konveierile lisatav esimene server jääb testkeskkonnaks, kuhu uus versioon paigaldatakse alati esimesena. Õnnestumise korral jätkatakse teistesse keskkondadesse paigaldamisega. Testkeskkonda paigaldamise ebaõnnestumisel ei paigaldata seda versiooni järgmistesse keskkondadesse.

### **5.2 SSIS paketi automaatne käivitamine ning andmefailide olemasolu kontrolli lisamine**

Enne SSIS paketi automaatset käivitamist tuleb lisada paketi töö algusesse andmefailide olemasolu kontroll. Selleks lisatakse paketi käsuvoos algusesse tööülesanne, mis kontrollib seadetes määratud kausta sisu. Kaustas failide olemasolul jätkatakse käsuvoos tööülesannete käivitamisega. Andmefailide olemasolu kontrolli lisatakse failide loetavuse kontroll ennetamaks failide kopeerimise ajal nende töötlemise.

Analüüsis paketi käivitamise lahenduse valimisel leiti et ainult käivitamise töö lisamine tekitab olukorra, kus samu faile töödeldakse mitu korda. Lahenduseks lisatakse SSIS paketi käsuvoos lõppu tööülesanne andmefailide teisaldamiseks teise kausta. Paketi

andmete liigutamise lõpuks peab olema andmefailide võtmise kaust tühi, et pakett järgmise käivitamise korral samu faile uuesti ei töötleks.

SSIS paketi automaatseks käivitamiseks luuakse andmebaasi serveris eelnevalt valitud SQL Server Agent töö. Töö ülesandeks on SSIS paketi käivitamine ja seda tehakse tööpäevadel seadistatava intervalli tagant. Töö käivitamise graafikul ei määrata lõpukuupäeva, töö desaktiveeritakse kui kliendi migratsioonide tegemine lõpetatakse.

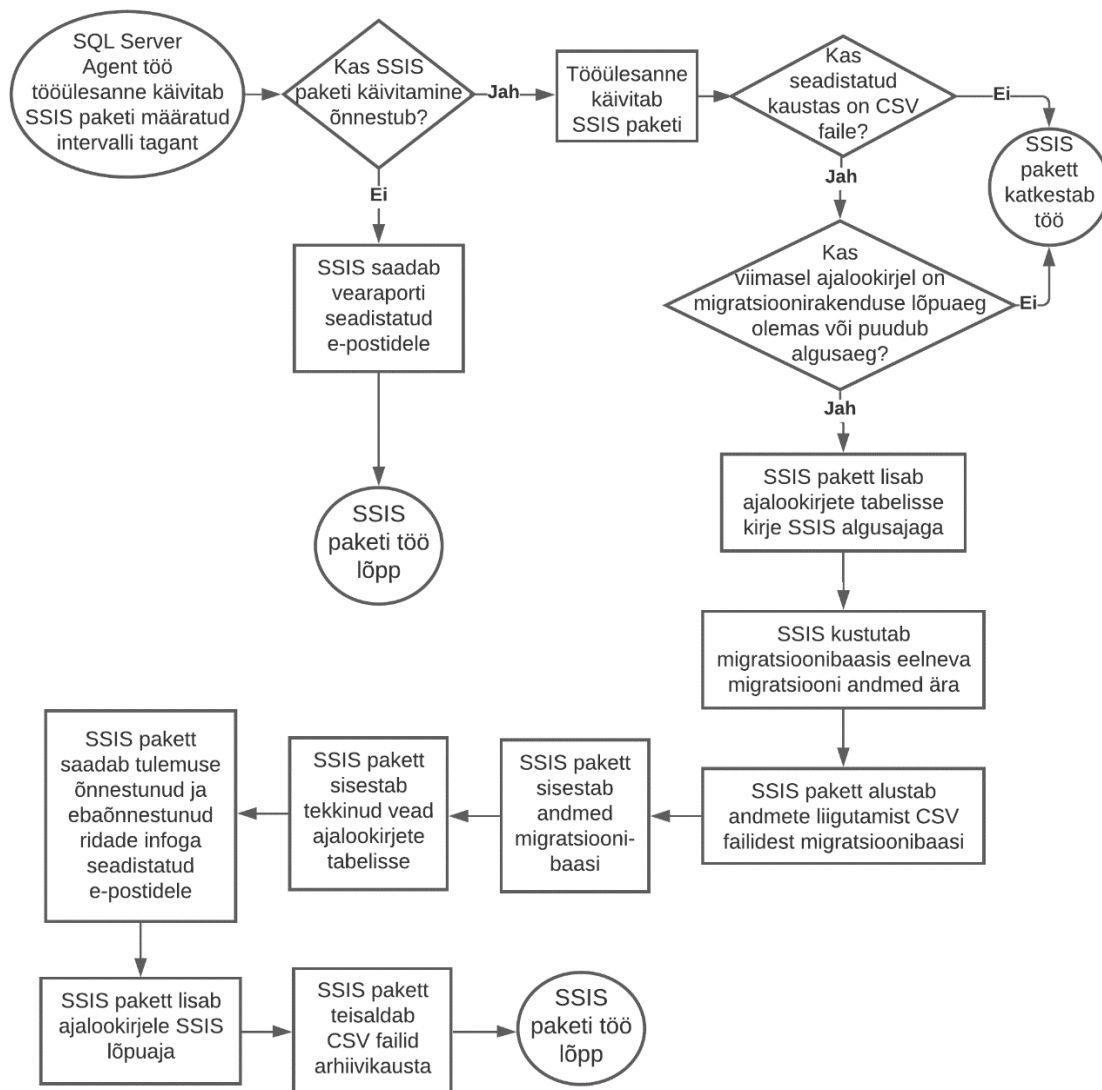
Migratsioonirakendusele on vajalik anda infot edasi kas migratsioonibaasis asuvad andmed vajavad põhirakenduse andmebaasi üle viimist. Selleks luuakse migratsioonibaasi uus tabel migratsiooni ajalookirjete jaoks.

Ajalookirjete tabel koosneb kuuest väärusest:

- ID – väärtustatakse automaatselt kasvavate väärtustega;
- Status – kas SSIS komponendi töö õnnestus;
- SSISStartTime – SSIS paketi käivitamise algusaeg;
- SSISEndTime – SSIS paketi töö lõpuaeg;
- MigrationStartTime – migratsioonirakenduse töö algusaeg;
- MigrationEndTime – migratsioonirakenduse töö lõpuaeg.

Tabelit täidab SSIS pakett peale andmete migratsioonibaasi sisestamist. Paketi käsuvoogu lisatakse SQL tööülesanded, mis sisestavad ajalookirjete tabelisse andmed SSIS alguse ja lõpu ning paketi töö õnnestumise kohta. Migratsioonirakenduse algus ja lõpp jääb SSIS paketi poolt tühjaks, need väärtustatakse migratsioonirakenduse töö käigus. SSIS paketi ei lisata ajalookirjete tabelit tühjendatavate tabelite hulka.

Eelnevate punktide teostus täidab nõudeid 4 ja 5. Nõude 4 kohaselt ei vaja SSIS pakett enam käsitsi käivitamist. Määratud intervalli tagant käivitatakse SSIS pakett ja paketi käsuvoogu alguses kontrollitakse, et andmete liigutamine käivitatakse ainult failide olemasolul (vt Joonis 4 lk 31). Nõue 5 lahendatakse konveierite loomisega, mis tegelevad SSIS projekti kokku ehitamise ja paigaldamisega.



Joonis 4. Uuendatud migratsiooni esimese etapi protsessi voodiagramm.

### 5.3 Migratsioonirakenduse uuenduste automatiseerimine ja automaatne käivitamine

Migratsioonirakendus juba asub samas versioonihalduses kus põhirakendus. Esimese sammuna tuleb luua ja seadistada kokku ehitamise konveier, mille sisendiks on migratsioonirakenduse uuenduse signaal versioonihaldusest.

Järgmiseks tuleb luua ja seadistada paigaldamise konveier, mille sisendiks on kokku ehitamise konveieri tulemus. Paigaldamise konveier viib uue rakenduse versiooni lisatud serveritesse. Konveierile lisatav esimene server jääb testkeskkonnaks, kuhu uus versioon paigaldatakse alati esimesena. Õnnestumise korral alustatakse teistesse keskkondadesse

paigaldamisega. Testkeskkonda paigaldamise ebaõnnestumisel ei paigaldata seda versiooni järgmistesse keskkondadesse.

Migratsioonirakenduse automaatseks käivitamiseks luuakse Task Scheduler-i töö mis käivitab migratsioonirakenduse igapäevaselt teatud intervalli tagant (vt Joonis 5 lk 33).

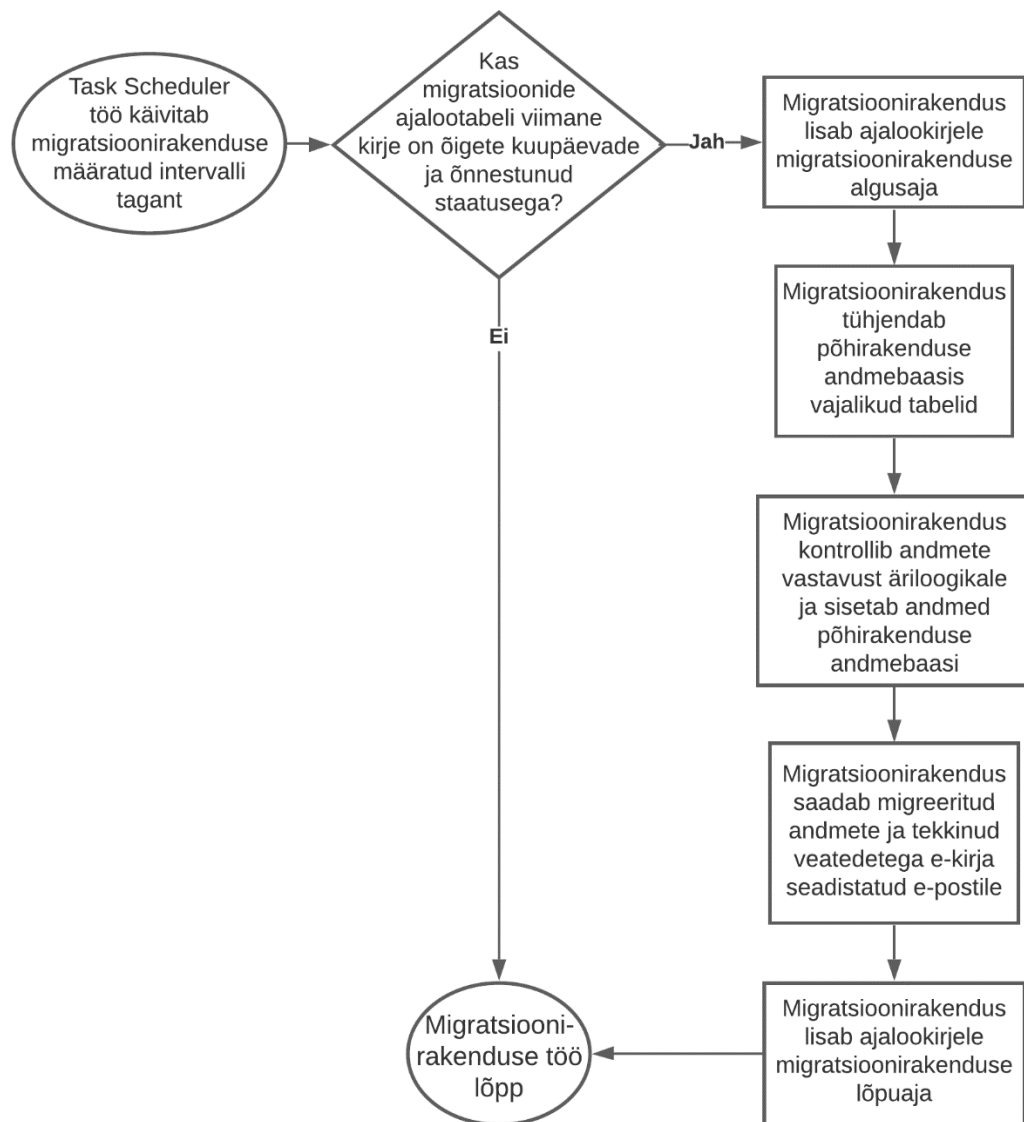
Migratsioonirakenduse koodi lisatakse migratsioonibaasi ajalookirjete tabeli hiliseima kirje kontroll, et saaks kindlaks teha kas migratsioonibaasi on lisatud uusi andmeid mida põhirakenduse baasi ei ole veel üle viidud ja mille töötlemine ei ole hetkel pooleli. Mitte põhirakenduse baasi migreeritud kirje olemasolul käivitatakse migratsioonirakendus, muul juhul rakendus ei jätku tööd.

Ajalookirjete kontrolli käivitatakse ainult Task Scheduler-i töö kaudu käivitamisel. Task Scheduler-i tööle seadistatakse argument mis antakse migratsioonirakendusele edasi. Rakenduse käivitamisel tehakse ajalookirjete kontroll ainult seadistatud argumenti olemasolul.

Ilma argumentita käivitamist teostatakse juhul kui klient ei soovi migratsiooni esimest etappi teha CSV failide variandis. Teise variandi puhul sisestab klient andmed ise migratsioonibaasi ja migratsioonirakendus käivitatakse käsitsi.

Peale andmete migreerimist lisab migratsioonirakendus migratsioonibaasi ajalookirjete tabelisse migreerimise lõpukellaaja. Rakenduse järgmise käivitamise jaoks on vajalik uus ajalookirje ilma lõpukuupäevata.





Joonis 5. Uuendatud migratsiooni teise etapi protsessi voodiagramm.

## 6 Teostus

Käesolevas osas kirjeldab autor töö praktilise osa teostust põhinedes planeeritud projektile. Teostati konveierite loomine, migratsiooni andmebaasi muutmine, komponentidele kontrollide ja täienduste lisamine ning komponentide käivitamise tööde loomine.

### 6.1 SSIS projekti viimine versioonihaldusesse ja konveierite loomine automaatseks uuendamiseks

SSIS projekt lisati Synerall põhirakendusest eraldi kausta ja viidi TFVC versioonihaldusesse. Järgmisena lisati Azure DevOps Server-ile laiend SSIS DevOps Tools, mis laeti alla Azure DevOps Marketplace lehelt. Loodi SSIS projekti jaoks kokku ehitamise konveier ning seejärel paigaldamise konveier.

Kokku ehitamise konveieril määrati projekti asukohaks TFVC versioonihaldus ja SSIS projekti absoluuttee versioonihalduse juurkataloogist. Konveierile lisati Build SSIS töö, millele seadistati projektifaili kataloogi absoluuttee ning töö tulemuse asukoht kasutades süsteemi muutujat.

Paigaldamise konveieril määrati sisendiks kokku ehitamise konveieri tulemus ning valiti paigaldatava projekt asukoht ja eelnevalt loodud kokku ehitamise konveier. Järjekorras esimeseks paigaldamise keskkonnaks lisati testserver, mille järele lisatakse edaspidi uued paralleelselt paigaldamise keskkonnad. Keskkondade järjekord on nähtav Lisa 2 pildil.

Paigaldamise konveierile lisati SSIS Deploy töö. Tööl seadistati sisendi asukoht kasutades süsteemi muutujat, sihtkohaks valiti SSISDB. Sihtkoha serveri nimeks ning kataloogi asukoha määramiseks kasutati parameetreid ja testkeskkonna jaoks määrati mõlemale sobiv väärtus. SSIS paketil lubati ülekirjutamine. Veaolukorra tekkimisel määrati paigaldamine katkestama antud keskkonnas.

## 6.2 SSIS failide olemasolu kontrolli ja automaatne käivitamine

SSIS pakatile lisati failide olemasolu kontroll. Kontrolli jaoks loodi SSIS paketi käsuvoog algusesse tööüks Script Task, kuhu lisati C# koodina failide olemasolu ning loetavuse kontroll. Failide puudumisel või lukustatud failide leidmisel käsuvoog katkestatakse.

Teiseks kontrolliks lisati viimase ajalookirje kontroll. Ajalookirjete jaoks lisati migratsioonibaasi loomise skripti MigrationLogs tabeli loomine. Käsuvoog määrati katkestama, kui viimasel ajalookirjel on olemas MigrationStartTime ja puudub MigrationEndTime. Selle kontrolliga välditi migratsioonirakenduse töö käigus migratsioonibaasi andmete muutmist.

SQL tööülesanne lisati SSIS paketi käsuvoog algusesse peale failide olemasolu kontrolli. See lisab SQL päringuga ajalookirjete tabelisse kirje koos Status ja SSISStartTime väärtusega.

Teine SQL tööülesanne lisati käsuvoog lõppu peale veateadete salvestamist andmebaasi. Lisatud tööülesanne annab SQL päringuga ajalookirjele SSISEndTime väärtuse.

SSIS paketi käsuvoog lõppu lisati CSV failide teisaldamine arhiivi, mida tehakse peale vigade e-kirja saatmist. Failide teisaldamiseks SSIS tööüks File System Task. Sihtkoha asukohta jaoks loodi uus paketi põhine muutuja ArchiveFolder, mille väärtuseks sisestati töödeldud failide kausta asukoht.

Automaatseks käivitamiseks loodi SSMS kaudu SQL Server Agent töö. Tööle lisati SSIS paketi käivitamise tööülesanne, nähtav Lisa 3 pildil. Tööülesande tüübiks valiti SQL Server Integration Services Package, mis määras tööülesandeks paketi käivitamise. Paketi asukohaks valiti SSIS kataloog, määrati server ja paketi täpne asukoht.

Tööle lisati käivitamise graafik. Tööde graafikuks määrati 15 minutiliste intervallide tagant iga tööpäev kella 7-st hommikul kuni kella 19-ni õhtul. Graafiku alguskuupäevaks määrati töö loomise kuupäev ning lõpukuupäeva ei määratud.

### **6.3 Migratsioonirakenduse konveierite loomine automaatseks uuendamiseks**

Kokku ehitamise konveieril määrati projekti asukohaks TFVC versioonihaldus ja lisati migratsioonirakenduse projekti absoluuttee versioonihalduse juurkataloogist. Konveierile lisati ning seadistati .NET rakenduse kokku ehitamise jaoks vajalikud tööd.

Paigaldamise konveieril määrati sisendiks kokku ehitamise konveieri tulemus. Lisati paigaldatava rakenduse asukoht ja eelnevalt loodud kokku ehitamise konveier. Sihtkoht serveris lisati seadistatava parameetriga.

Paigaldamise keskkonnad seadistati samamoodi nagu SSIS projekti paigaldamise konveieril. Esimeseks paigaldamise keskkonnaks määrati testkeskkond ja edaspidised keskkonnad lisatakse peale seda. Keskkondade järjekorra näide on nähtav Lisa 2 pildil.

### **6.4 Migratsioonirakenduse täiendamine ja automaatne käivitamine**

Esimese sammuna lisati migratsioonirakenduse Main meetodi algusesse migreeritavate andmete olemasolu kontroll, mida teostatakse ainult argumendi CheckMigrationLogs olemasolul. Migratsioonibaasist päritakse MigrationLogs tabelist kõige viimane kirje. Rakendus määrati jätkama tööd, kui viimasel kirjel on edukas staatus, SSISEndTime väljal on väärtus ning MigrationStartTime ja MigrationEndTime on ilma väärtusteta. See lisati vältimaks migratsioonirakenduse käivitamist, kui SSIS paketi töö on pooleli või kui eelnevalt käivitatud migratsioonirakendus ei ole tööd lõpetanud.

Migratsioonirakenduse töö algusesse peale kontrollide õnnestumist lisati ajalookirjele MigrationStartTime väärtuse lisamine. Migratsioonirakenduse töö lõppu lisati ajalookirje MigrationEndTime väärtustamine.

Migratsioonirakenduse käivitamiseks lisati serverisse Task Scheduler tööülesanne. Tööülesandele lisati argument CheckMigrationLogs, mis antakse käivitatavale rakendusele edasi. Tööülesande graafikuks määrati tööpäevad kella 7:15-st hommikul kuni kella 19:45-ni öhtul iga 15 minuti tagant.

## 7 Tulemus

Töö tulemusena automatiseeriti Syneralli andmemigratsiooni kaks põhilist probleemi vastavalt nõuetele. Nendeks olid komponentide käsitsi uuendamine ning andmemigratsiooni protsessi komponentide käsitsi käivitamine.

Komponentide uuendamiseks ei ole vajalik arendajal kõikidesse serveritesse ligipääsu, kuna uuendamine ei toimu käsitsi enam. Uuenduste tegemiseks on vajalik muudatused versioonihaldusesse üles laadida ja edasise uuendamisega tegelevad Azure DevOps Server-i konveierid. Uuendused paigaldatakse kõikidesse serveritesse korraga, kui testserveris uuendamine õnnestus. Sellega tagatakse kõikide klientide serverites sama versiooni olemasolu, milleks on kõige uuem versioon. Lisaks asub SSIS projekt ühises versioonihalduses, seega edaspidiste muudatuste ajalugu on jälgitav.

Paigaldamise konveier lihtsustab ja kindlustab ka komponentide paigaldamist uue kliendi serverisse. Enam ei pea faile käsitsi serveritesse lisama, piisab konveieril uue keskkonna lisamisest ning seadistamisest. Ainult SSIS paketi puhul peab andmebaasi lisama tühja kataloogi SSMS-i kaudu. Uues serveris peab andmemigratsiooni protsessi automaatseks käivitamiseks lisama komponentide käivitamise tööd.

Protsessi käivitamisel ei ole enam vajalik Syneralli töötaja sekkumine. Piisab kliendi CSV andmete lisamisest seadistatud failikataloogi. Andmete migreerimise komponentide käivitamisel läbitakse andmete liigutamise loogika ainult uute andmete olemasolul. Protsess ei jää pooleli peale esimese komponendi tööd. Peale esimese komponendi edukat lõppu salvestatakse andmed, mille põhjal käivitatakse järgmine komponent. Välistatud on olukord, kus andmete migratsioon alustati, aga protsess jäi inimvea tõttu pooleli.

Tehtud töö vähendab Syneralli töötajate andmemigratsiooni uuenduste ja protsessile kuluvat tööaega. Säästetud aja saavad töötajad suunata tootearendusse. Lisaks ei kulu enam töötajate ega kliendi tööaeg algselt tekkinud inimvigadest tulenevate olukordade uurimisele ja lahendamisele.

## 8 Tulevikuplaanid

Tulevikus saab Syneralli migratsiooniprotsessile juurde lisada migratsioonide statistika ja vaatamise lahenduse, et oleks võimalik vaadelda migratsioonide andmekvaliteeti läbi aja. Hetkel saadetakse migratsiooni lõpptulemus e-kirja teel seadistatud töötajatele. Tulemustes on loetletud kõik veateated arendajale arusaadaval kujul. Kliendi jaoks koostatakse tulemuste aruanne käsitsi migratsiooni tulemustest.

Tulevikus võiks olla kliendil võimalik vaadata migratsiooni andmekvaliteedi ajalugu ja tulemuste aruannet peale migratsiooniprotsessi lõppu. Tulemustes on vajalik näha õnnestunud ja ebaõnnestunud ridade ning tekkinud vigade kokkuvõtet.

Lahenduse jaoks on vaja uurida, mis andmeid on vajalik näidata, mis kujul ja kuidas. Statistika näitamiseks on palju erinevaid viise, millest tuleb valida kõige sobivam ja turvalisem. Turvalisuse puhul on kõige olulisem, et andmetele ei oleks kõrvalistel isikutel ligipääsu ning et andmed ei liiguks kliendi serverist välja.

Tekkinud vigade kokkuvõtte koostamiseks on vajalik luua kriteeriumid, mille põhjal on võimalik vigu kategoriseerida. Kategooriatele tuleb määrata kriitilisus, et kliendil oleks võimalik kriitilisemad vead eelisjärjekorras üle vaadata.

## 9 Kokkuvõte

Käesolevas töös lahendati Synerall AS-i toote andmemigratsiooniga seotud probleeme. Probleemiks oli ajakulukas, veaohklik ning manuaalne töö. Manuaalne töö koosnes andmemigratsiooni komponentide uuendamisest ning andmemigratsiooni protsessi läbimisest.

Komponentide uuendamiseks oli vajalik arendajatel kõikidesse kliendi serveritesse ligipääsu, et uuendusi käsitsi serveris paigaldada. See tekitas olukordi, kus uuendusi ei paigaldatud kõikide klientide serveritesse.

Andmemigratsiooni protsessi puhul oli vajalik mõlema komponendi õigeaegne käivitamine. Klient pidi ise teavitama Syneralli töötajaid andmemigratsiooni alustamiseks. Sarnaselt komponentide uuendamisel oli ka käivitamiseks vajalik töötaja ligipääs serverisse. Probleemseks kohaks oli veel inimveana teise komponendi käivitamise unustamine.

Töö eesmärgiks oli muuta komponentide uuendamine ning migratsiooni protsess automaatsemaks, et vähendada selle peale kuluvat töötajate tööaega ja vigade teket. Komponentide uuendamine pidi muutuma töötajate jaoks lihtsamaks, kiiremaks ja veakindlamaks. Andmemigratsiooni protsess pidi ilma töötaja sekkumiseta käivituma CSV failide olemasolul ning tegema migratsiooni algusest lõpuni.

Komponentide uuendamiseks kasutati Azure DevOps Server-i versioonihaldust ja konveiereid. Protsessi algusest lõpuni töötamiseks kasutati SSIS paketi käivitamiseks SQL Server Agent tööd. Migratsioonirakenduse käivitamiseks kasutati Task Scheduler tööd.

Mõlema komponendi sisu muudeti ning loodi uus migratsiooni ajalookirjete tabel. SSIS pakatile lisati failide olemasolu ja kompleksuse kontroll, uue ajalookirje lisamine ning töödeldud failide teisaldamine. Migratsioonirakendusele lisati ajalookirjete tabelist uute andmete olemasolu kontroll. Kontrolli teostatakse ainult Task Scheduler tööle lisatud parameetri olemasolul. Migratsioonirakenduse manuaalsel käivitamisel ilma parameetrita kontrolli ei teostada.

Töö tulemuseks on kiirem ja lihtsam komponentide muutmine ning andmemigratsiooni protsessi automaatne käivitamine andmefailide olemasolul. Arendajal on vaja komponentide muudatuste kasutusse võtmiseks muudatused versioonihaldusesse üles laadida. Komponentide serveritesse paigaldamisega tegelevad konveierid. See tagab kõikide klientide jaoks kõige uuema versiooni kasutamise. Andmemigratsiooni tegemine ei vaja enam töötaja sekkumist, see käivitub kliendi andmete olemasolul ning kogu töö tehakse automaatselt algusest lõpuni. Protsess ei jää inimvea tõttu poolikuks.

Andmemigratsiooni probleemide lahenduste tulemusena säästetud tööaja saab suunata tootearendusse. Samuti ei kulu lisa aega inimvigade tõttu tekkinud olukordade lahendamisele.

Migratsiooni protsessi autmatiseerimise ning komponentide halduse eesmärgid täideti käesoleva töö raames.

Migratsiooniprotsessi tulevaseks edasiarendusteks on migratsiooni tulemuste statistika lahenduse välja töötamine. Selle jaoks on vajalik tekkinud vigade kategoriseerimine ning statistika kliendile kuvamine.



## Kasutatud kirjandus

- [1] Synerall AS, „Synerall v 2.11. CIS and billing software features,“ Tallinn, 2020.
- [2] I. Ivanov, Interviewee, *Syneralli migratsiooniprotsess*. [Intervjuu]. 26 03 2021.
- [3] Synerall AS, „Synerall data migrations,“ Tallinn, 2018.
- [4] Synerall AS, „SYN-D-002 installation manual,“ Tallinn, 2010.
- [5] K. Rungi, Interviewee, *Migratsiooniprotsessi probleemid ja nõuded*. [Intervjuu]. 14 08 2020.
- [6] B. Laster, *Continuous Integration vs. Continuous Delivery vs. Continuous Deployment*, 2nd Edition, Sebastopol: O'Reilly Media, Inc., 2020.
- [7] C. O. Matthew Skelton, *Continuous Delivery with Windows and .NET*, Sebastopol: O'Reilly Media, Inc., 2016.
- [8] C. Chandrasekara, *Beginning Build and Release Management with TFS 2017 and VSTS: Leveraging Continuous Delivery for Your Business*, Colombo: Apress, 2017.
- [9] Microsoft, „Introducing Azure DevOps,“ Microsoft, 10 09 2018. [Võrgumaterjal]. Available: <https://azure.microsoft.com/en-us/blog/introducing-azure-devops/>. [Kasutatud 4 04 2021].
- [10] Microsoft, „Choosing the right version control for your project,“ Microsoft, 12 05 2017. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/devops/repos/tfvc/comparison-git-tfvc>. [Kasutatud 05 05 2021].
- [11] Microsoft, „What is Azure Pipelines?,“ Microsoft, 31 05 2019. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines>. [Kasutatud 11 04 2021].
- [12] Microsoft, „SQL Server Integration Services,“ Microsoft, 07 06 2018. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services>. [Kasutatud 04 04 2021].
- [13] Microsoft, „Integration Services (SSIS) Packages,“ Microsoft, 8 31 2016. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/integration-services/integration-services-ssis-packages>. [Kasutatud 17 04 2021].
- [14] T. LeBlanc, „Deploying Packages to SQL Server Integration Services Catalog (SSISDB),“ SQLShack, 16 01 2017. [Võrgumaterjal]. Available: <https://www.sqlshack.com/deploying-packages-to-sql-server-integration-services-catalog-ssisdb/>. [Kasutatud 23 4 2021].
- [15] Microsoft, „SQL Server Integration Services (SSIS) DevOps Tools Azure DevOps extension,“ Microsoft, 12 06 2019. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/integration-services/devops/ssis-devops-overview>. [Kasutatud 15 04 2021].
- [16] L. Burrows, „Understanding the Role of SQL Server Agent in SQL Server,“ SQLServerCentral, 06 06 2018. [Võrgumaterjal]. Available: <https://www.sqlservercentral.com/blogs/understanding-the-role-of-sql-server-agent-in-sql-server>. [Kasutatud 25 04 2021].

- [17] A. Das, „Introduction to the SQL Server Agent,“ SQL Shack, 15 12 2020. [Võrgumaterjal]. Available: <https://www.sqlshack.com/introduction-to-the-sql-server-agent/>. [Kasutatud 25 04 2021].
- [18] M. Huculak, „How to create an automated task using Task Scheduler on Windows 10,“ Windows Central, 30 01 2019. [Võrgumaterjal]. Available: <https://www.windowscentral.com/how-create-automated-task-using-task-scheduler-windows-10>. [Kasutatud 27 04 2021].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Arnika Rästa

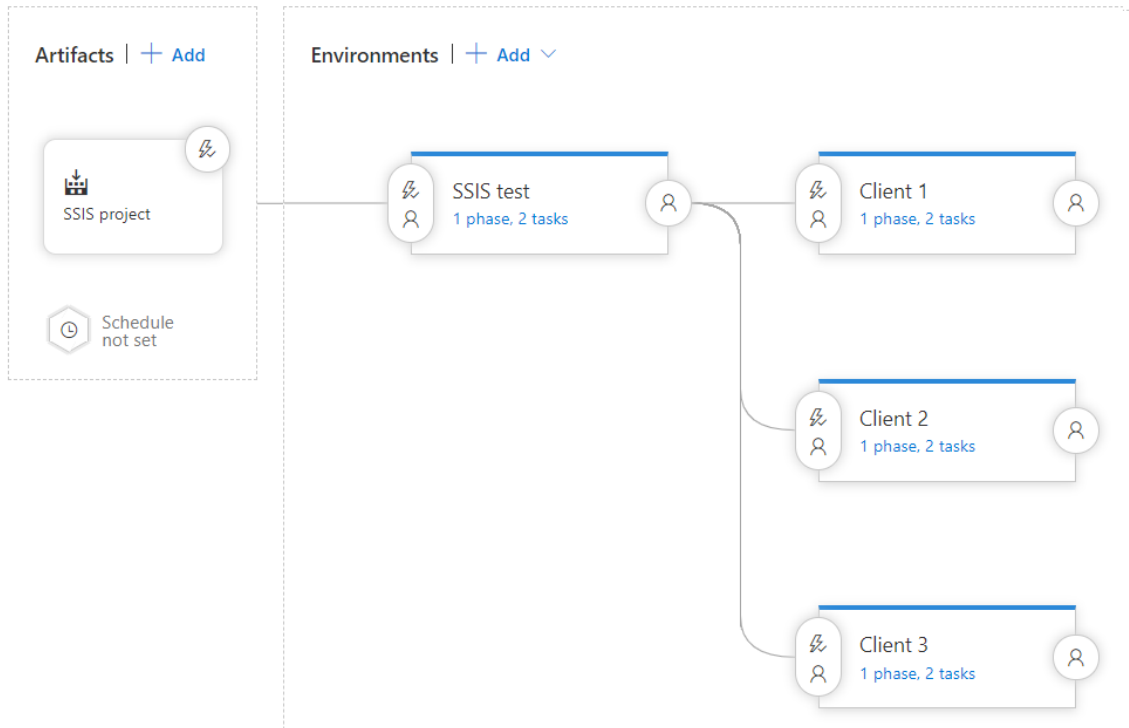
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Andmete migratsiooniprotsessi automatiseerimine AS-i Synerall näitel", mille juhendaja on Priit Rospel
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Paigaldamise konveieri keskkondade järjekorra seadistamine testkeskkonnaga



## Lisa 3 – SQL Server Agent töö loomisel SSIS paketi käivitamise tööülesande lisamine SSMS-i kaudu

