

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatika instituut

Rauno Pihlak 132421IABMM

**ANDMEIDA LAADIMISTE
PLANEERIMISE TÖÖRIISTA VALIK
FINANTSETTEVÕTTES**

Magistritöö

Juhendaja: Eduard Ševtšenko
Doktorikraad

Tallinn 2017

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: Rauno Pihlak

11.04.2017

Annotatsioon

Käesoleva diplomitöö eesmärgiks on valida finantsettevõttele uus andmeida laadimise planeerimise tööriist. Ettevõtte olemasolev tööriist ei võimalda andmete laadimisi planeerida efektiivselt ning töötab vananenud platvormil. Seetõttu tuleb kasutusele võtta uus vahend.

Diplomitöö esimene osa annab ülevaate andmeiteadest, olulisematest komponentidest ning andmelaadimise protsessist.

Töö käigus kirjeldatakse nõuded valitavale tööriistale. Detailsemalt analüüsitakse nelja erinevat tööriista ning tehakse valik.

Uue tööriista kasutusele võtmise eelduseks on olemasolevate andmete migreerimine vanast tööriistast. Diplomitöö sisaldab analüüsi võimalike migratsiooni meetodite kohta ning kirjeldab vajalikud sammud uue tööriista kasutusele võtmiseks.

Järelanalüüs koosneb valiku protsessi järelhindamisest ning annab soovitusi edasisteks tegevusteks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 47 leheküljel, 8 peatükki, 11 joonist, 7 tabelit.

Abstract

Data Warehouse Loading Planning Tool Selection in a Financial Services Company

The purpose of this thesis is to select a new data warehouse loading planning tool for a financial services company. The existing tool is not fulfilling the requirements of planning loadings efficiently and runs on an outdated platform.

The second chapter describes data warehouses, their components and the data loading process. The process of loading data from source systems to the warehouse main layer and data marts is the basis for this thesis.

Functional and non-functional requirements for selecting a new tool are described in the third chapter. Both requirements are predecessors for analysing the functionalities of new possible tools.

Four different tools are analysed in the fourth chapter and as a result, a new tool is selected. It is concluded that many different workflow automation tools are applicable as a data warehouse loading planning tool.

Data migration from the old system to the new system is a prerequisite to start using the selected tool. The sixth chapter describes different methods and ways to conduct the migration and gives an overview of the necessary steps.

The post-analysis chapter evaluates the selection process and brings out the recommendations for future activities. Two simulation models in the example of one loading flow show the benefit of complex dependencies configuration.

The thesis is in Estonian and contains 47 pages of text, 8 chapters, 11 figures, 7 tables.

Lühendite ja mõistete sõnastik

API	<i>Application programming interface</i> , programmiiliides
CMDB	<i>Configuration management database</i> , konfiguratsiooni halduse andmebaas
CRM	<i>Customer relationship management</i> , kliendisuhete juhtimine
ERP	<i>Enterprise resource planning</i> , ettevõtte ressursside planeerimine
ETL	<i>Extract, transform, load</i> , ekstraktimine, transformeerimine, laadimine
FIFO	<i>First in, first out</i> , esimesena sisse, esimesena välja
FTP	<i>File transfer protocol</i> , failiedastusprotokoll
ID	<i>Identifier</i> , identifikaator
RAW	Oracle andmetüüp
SFTP	<i>Secure file transfer protocol</i> , turvaline failiedastusprotokoll
SLA	<i>Service level agreement</i> , teenusetaseme leping
SQL	<i>Structured query language</i> , struktuurpäringukeel
SSO	<i>Single sign-on</i> , ühekordne sisselogimine
ZIP	Arhiveerimise faili formaat
XML	<i>Extensible markup language</i> , laiendatav märgistuskeel

Sisukord

1 Sissejuhatus	10
1.1 Taust ja probleem	10
1.2 Ülesande püstitus	10
1.3 Metoodika.....	10
1.4 Töö struktuur	11
2 Andmeait ja andmelaadimiste planeerimine.....	12
2.1 Andmeait	12
2.2 Andmelaadimiste planeerimine	13
2.3 Laadimiste ressursi jaotamine	17
3 Nõuded tööriistale	18
3.1 Funktsionaalsed nõuded	19
3.2 Mittefunktsionaalsed nõuded.....	23
4 Andmeaida laadimiste planeerimiseks sobivad tööriistad.....	25
4.1 BMC Control-M	25
4.2 IBM Workload Scheduler.....	26
4.3 VisualCron.....	28
4.4 Stonebranch Universal Controller	29
4.5 Infrastruktuuri tugi.....	31
5 Tööriista valimine.....	33
5.1 Vastavus funktsionaalsetele nõuetele	33
5.2 Vastavus mittefunktsionaalsetele nõuetele.....	34
5.3 Tööriista valik.....	35
6 Migratsiooniplaan	36
6.1 Import moodul	37
6.2 Datamigrate	40
6.3 Andmebaasi põhine migratsioon	40
6.4 Migratsiooni tegevused.....	41
7 Järelanalüüs	43
7.1 Valiku protsessi hinnang	43

7.1.1 Projekti meeskonna loomine ja tegevuskava.....	43
7.1.2 Informatsiooniallikate kaardistamine	44
7.1.3 Olemasoleva protsessi ülevaatamine	44
7.1.4 Äri vajaduste prioritseerimine	45
7.1.5 Valiku eelpiiramine	45
7.1.6 Tarkvara pakkujate hindamine	45
7.1.7 Võimalike pakkujate hindamine	46
7.1.8 Otsuse tegemine.....	46
7.1.9 Implementatsiooniplaani loomine	46
7.1.10 Hinna ja SLA tingimuste seadmine	47
7.2 Simulatsioon protsessi ahela näitel.....	47
7.3 Võimalikud edasised tegevused.....	49
8 Kokkuvõte	51
Kasutatud kirjandus	53
Lisa 1 – Extract faili struktuur ja sisu.....	55

Jooniste loetelu

Joonis 1. Andmeaida struktuur	13
Joonis 2. Näide: Protsesside grupp nimega Protsess_grupp_1 koondab endas mitut erinevat protsessi	14
Joonis 3. Näide: Sõltuvused kolme protsesside grupi vahel.....	15
Joonis 4. Näide: Sõltuvused kolme protsesside grupi vahel loogiliste gruppidega.....	16
Joonis 5. Control-M arhitektuuri joonis	26
Joonis 6. IBM Workload Scheduler arhitektuuri joonis.....	28
Joonis 7. Universal Controller arhitektuuri joonis.....	31
Joonis 8. Näide: Protsessi Protsess_transformeerimine_2_1 definitsioon	38
Joonis 9. Näide: Protsesside grupi Protsess_grupp_2 definitsioon	39
Joonis 10. Simulatsioon nelja protsesside grupi vahel	48
Joonis 11. Sõltuvustega simulatsioon nelja protsesside grupi vahel	49

Tabelite loetelu

Tabel 1. Nõuetele määratud prioriteedid	23
Tabel 2. VisualCron ning tugiteenuse hinnad	29
Tabel 3. Infrastruktuuri tugi	32
Tabel 4. Kõrgeima prioriteetiga funktsionaalsetele nõuetele vastavus	33
Tabel 5. Funktsionaalsetele nõuetele vastavus. Prioriteedid 2-3.....	34
Tabel 6. Mittefunktsionaalsete nõuete vastavus	34
Tabel 7. Praktikad ning vastavad tegevused.....	36

1 Sissejuhatus

1.1 Taust ja probleem

Finantsettevõttes hetkel kasutusel olev andmeaida laadimiste planeerimise vahend töötab vananenud Microsofti platvormil. Sellega kaasneb ettevõttele risk ja kulu, mille maandamiseks on oluline valida ja implementeerida uus tööriist. Teine oluline aspekt - pikalt kasutusel olnud tööriist ei täida tänapäevaseid nõudeid, millega hallata suurt hulka andmete laadimiste protsesse efektiivsel viisil. Ettevõtte andmeaita lisandub pidevalt uusi arendusi ning süsteemi keerukus kasvab, mistõttu on tõhus haldamine äärmiselt oluline. Diplomitöö kirjutatakse ühe ettevõtte näitel, kuid nõuete ja valiku tegemise osa on kasutatav ka teistes ettevõtetes, mis tegelevad andmeaitadega.

1.2 Ülesande püstitus

Diplomitöö eesmärk on kirjeldada nõuded ning valida välja uus andmeaida laadimiste planeerimise tööriist. 2017 kevadel peab ettevõtte plaanide kohaselt tööriista valik olema tehtud ning valitud vahend paigaldatud arendus- ja testkeskkondades. Valitava vahendiga loodav andmelaadimiste planeerimine peab olema efektiivne, arusaadav ja vastama seatud nõuetele.

Lisaks on käesoleva diplomitöö skoobis kirjeldada migratsiooniplaani uue vahendi kasutusele võtmiseks, mis koosneb olemasoleva tööriista andmete osalisest või täielikust teisendamisest sisendiks uuele planeerimise tööriistale. Kui valitav vahend ei võimalda täielikult automatiseeritud migratsiooni, annab autor hinnangu ning kirjeldab plaani vajalike etappidega uue tööriista kasutusele võtmiseks. See on oluline eeldus sujuvaks üleminekuks ja erinevate keskkondade konfigureerimiseks.

1.3 Metoodika

Eesmärgi saavutamiseks kirjeldatakse nõuded valitavale tarkvarale. Nõuete kirjeldamisel kaasatakse ettevõtte haldus ning arendusmeeskonna liikmeid, et selgitada

välja hetkel kasutusel oleva tööriista head omadused ning suuremad puudused. Kuna ettevõtte teistes üksustes on olnud projekte tööde automatiseerimise osas, on võimalik osaliselt koostatud analüüside tulemusi kasutada ka antud töös. Lisaks lähtutakse nõuete kirjeldamisel parimatest praktikatest, mida erinevad uurimuste autorid on välja toonud.

MoSCoW meetodit kasutades määratakse nõuetele prioriteetid lihtsustamaks hilisemat valiku tegemist. Olemasolevate tööriistade valimi piiramiseks kasutatakse analüüsi ning hinnanguid pakutava tarkvara osas. Üheks oluliseks allikaks on Gartneri maagiline ruut töövoogude automatiseerimise osas. Valimi tulemit analüüsitakse detailsemalt ja hinnatakse vastavust seatud nõuetele.

1.4 Töö struktuur

Töö jaguneb kuude sisulisse peatükki, lisaks sissejuhatus ning kokkuvõte. Teises peatükis kirjeldatakse andmeaitasid ning andmelaadimiste planeerimist üldises võtmes. Välja tuuakse olulisemad mõisted, komponendid ja seosed.

Kolmandas peatükis kirjeldatakse funktsionaalsed ja mittefunktsionaalsed nõuded valitavale vahendile. Igale funktsionaalsele nõudele määratakse prioriteet kasutades sobivat metoodikat.

Neljanda peatüki subjektiks on erinevad tööriistad, mida saab andmelaadimiste planeerimiseks kasutada. Gartneri maagiline ruut on põhiliseks sisendiks valiku tegemisel.

Viiendas peatükis valitakse sobiv vahend. Valiku aluseks on kolmandas peatükis kirjeldatud nõuded ning neljandas peatükis vaadeldud tööriistad.

Kuuenda peatüki sisuks on migratsiooniplaan ja vajaliku migratsiooni etappide välja töötamine. Plaan peab tagama sujuva ülemineku uue tööriista kasutusele võtmiseks.

Seitsmendas peatükis keskendutakse järelanalüüsile tehtud töö osas ja antakse soovitusi edasisteks tegevusteks.

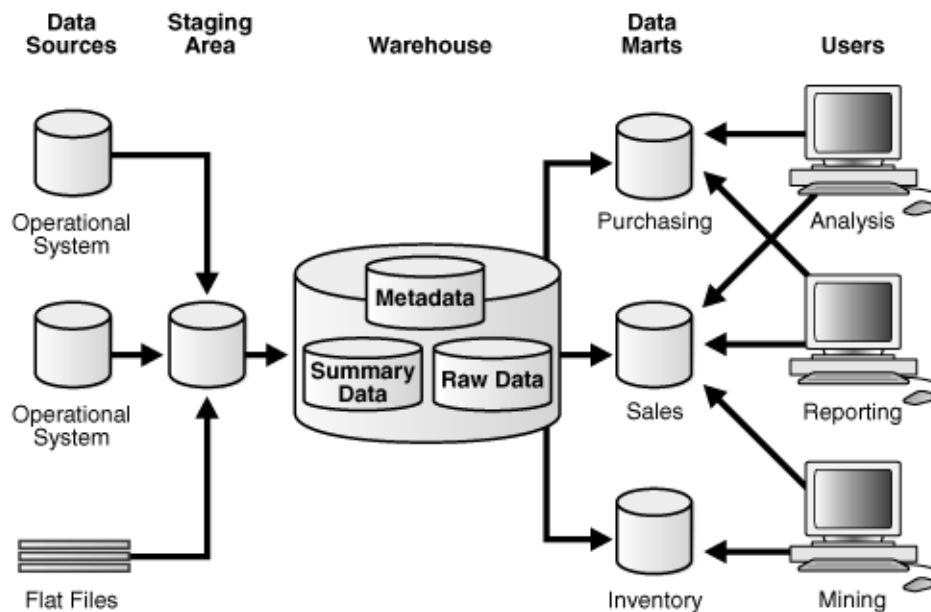
2 Andmeait ja andmelaadimiste planeerimine

Järgnevate alampeatükkide eesmärk on anda ülevaade andmeaitadest ning andmelaadimiste (*Data loading*) planeerimisest üldiselt, kuid ka diplomitöö aluseks oleva finantsettevõtte näitel. Välja tuuakse põhilised mõisted ning andmeaida komponendid. Viimases alampeatükis kirjeldatakse ressursside jaotamist ning selle olulisust andmelaadimiste protsessis.

2.1 Andmeait

Andmeait koondab endas andmeid erinevatest allikasüsteemidest. Nendeks võivad olla ettevõtte operatiivandmebaasid, erinevat tüüpi andmefailid, logifailid, ERP (*Enterprise resource planning*) või CRM (*Customer relationship management*) süsteemidest pärinevad andmed. Lisaks ettevõtte välised andmed ehk näiteks avalik info internetist või koostööpartnerite vahel jagatavad andmestik. Gartneri definitsiooni kohaselt on andmeait andmete hoiustamise süsteem, kuhu talletatakse andmed operatiivsüsteemidest ja välistest andmeallikatest [1].

Laialdaselt kasutatava struktuuri (Joonis 1) korral koosneb andmeait viiest põhikihist. Teise kihti laetakse andmed esimesest allikasüsteemide kihist muutmata või vähesel määral teisendatud kujul. Kolmandasse kihti agregeeritakse andmed sõltuvalt nõuetest ning neljas kiht võib olla andmevakk, vaadete kiht (*Presentation layer*) või mõni muu andmete esitamiseks sobiv vorm. Neljandat kihti kasutatakse raportite ja analüüside koostamiseks. Samas ei ole selline viiekihiline struktuur range raamistik ning ettevõteted kohandavad andmeaida arhitektuuri vastavalt vajadustele [2].



Joonis 1. Andmeaida struktuur [3] jaguneb viide kihti. Esimeses on allikaandmed (*Data Sources*), teises *Staging Area* kihis on allikaandmed laetud muutmata kujul. *Warehouse* kihis on andmed transformeeritud kujul ning raporteerimiseks ja aruandluseks kasutatakse andmevakkasid (*Data Marts*). Viies kiht koosneb tööriistadest, mille abil analüüse ja raporteid koostatakse.

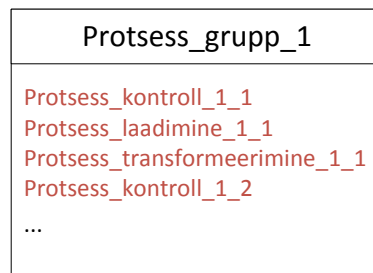
Andmeait koondab endas tihti erinevaid tehnoloogiaid ja tarkvara komponente. Üks suur komponent on andmelao platvorm ehk andmebaasisüsteem, mida kasutatakse andmete talletamiseks. Teine suur komponent on ETL (*Extract, transform, load*) tööriist, mille abil andmed laetakse ning transformeeritakse sobivale kujule. Lisaks erinevad raporteerimise ja andmeanalüüsi tööriistad. Nimetatud kolme suure komponendi kõrval on samuti oluline laadimiste planeerimise tööriist. Väiksemate ja lihtsamate andmeaitade korral ei pruugi see suurt tähtsust omada. Lihtsamat ajastamist (*Scheduling*) pakuvad tihti ETL vahendid või operatsioonisüsteemiga kaasasolevad ajastamise lahendused. Keerulisema andmeaida korral ei pruugi sellised tööriistad olla piisavalt efektiivsed ning tuleb kasutada mõnda spetsiaalset ajastamise ja planeerimise vahendit.

2.2 Andmelaadimiste planeerimine

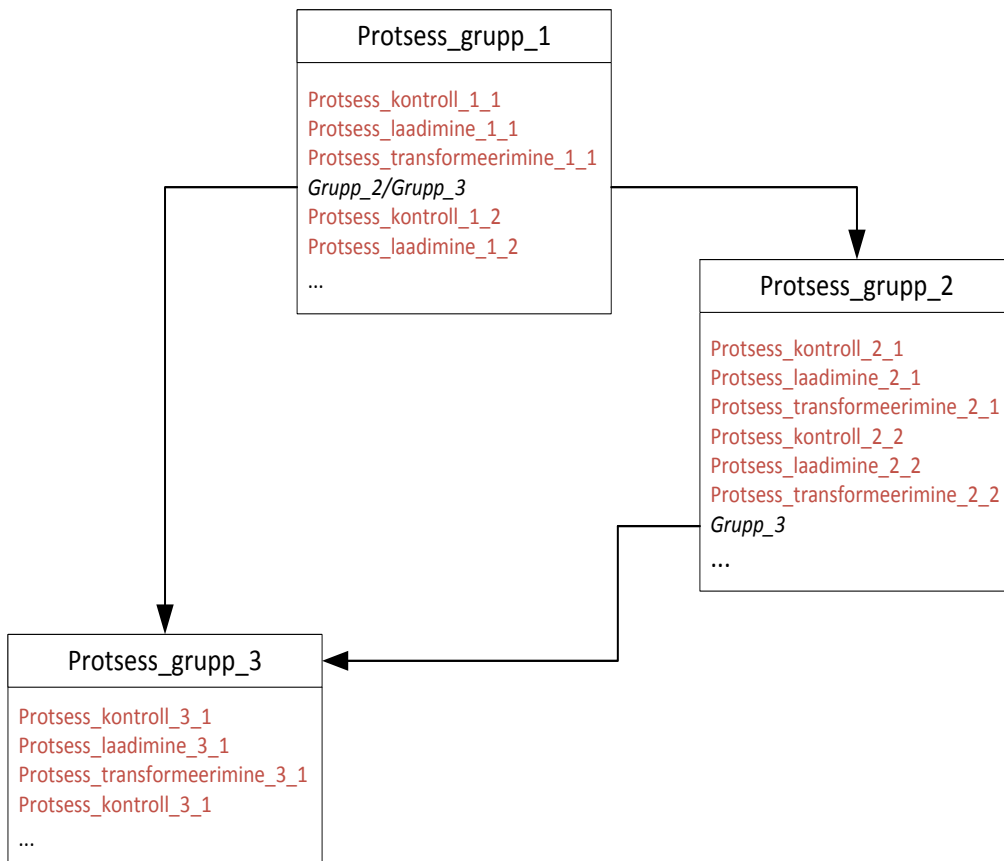
Andmeaida andmetega laadimine toimub enamasti määratud intervalli alusel. Diplomitöös käsitletavas ettevõttes on levinuim igapäevane andmelaadimine, mis on ajastatud väljaspool tööaega ning kuine andmelaadimine, mis võib sõltuda näiteks puhkepäevadest või pühadest. Lisaks eraldi kvartaalsed, aastased ning käsupõhiseid (*On-demand*) andmelaadimised. Kuna ettevõtte tegutseb mitmes riigis, toob see kaasa

erinevad pühad riikide lõikes ning see lisab komplitseeritust laadimiste planeerimisel. Samas on oluline, et laadimised toimuksid õigeaegselt, sest finantsettevõttele seatavad ajalised piirangud raportite osas on ranged ning mitte õigeaegne esitamine toob kaasa trahvid ja piirangud.

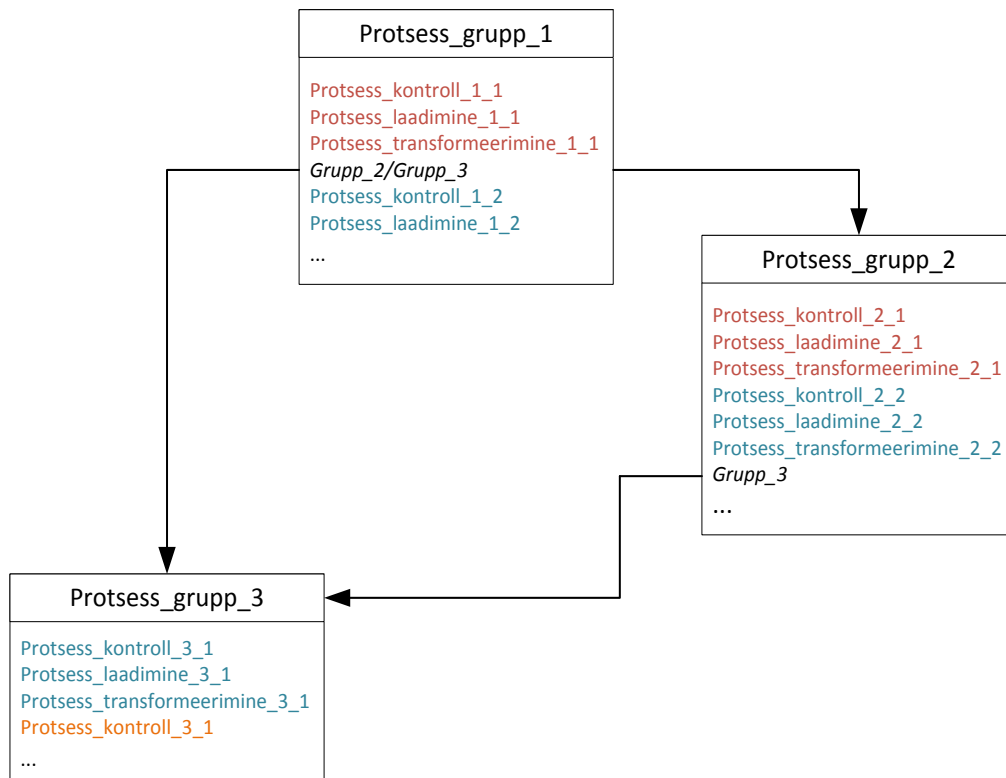
Diplomitöös käsitletavas ettevõttes on laadimiste protsessid grupeeritud. Diplomitöös kasutatakse illustreerivaid protsesside ja protsessi gruppide nimetusi. Üks protsesside grupp võib sisaldada ühte kuni mitut protsessi (Joonis 2). Samas võivad protsesside grupid omada omavahelisi sõltuvusi (Joonis 3). Samuti võivad protsessid jaguneda omakorda mõnda loogilisse gruppi (Joonis 4), mis hetkel kasutusel ei ole, kuid vajadus sellise grupeerimise järele eksisteerib. Loogilised grupid võivad olla laadimised ühest allikasüsteemist või ühe liigilised andmed ehk kliendiandmed või lepingud. Andmelaadimiste planeerimise puhul on väga oluline järjekord, mille alusel protsesse ning nende grupe käivitatakse. Vale järjekorra kasutamisel või halva planeerimise korral on andmete hilisem parandamine andmeaidas mahukas ja aeganõudev töö.



Joonis 2. Näide: Protsesside grupp nimega Protsess_grupp_1 koondab endas mitut erinevat protsessi. Punase värviga on tähistatud protsesside kuulumine ühte loogilisse gruppi.



Joonis 3. Näide: Sõltuvused kolme protsesside grupi vahel. Punase värviga on tähistatud protsesside kuulumine ühte loogilisse gruppi. Protsess_grupp_2 sõltub Protsess_grupp_1 protsessi Protsess_transformeerimine_1_1 edukast käivitamisest. Protsess_grupp_3 sõltub Protsess_grupp_1 protsessi Protsess_transformeerimine_1_1 edukast käivitamisest ning Protsess_grupp_2 protsessi Protsess_transformeerimine_2_2 edukast käivitamisest.



Joonis 4. Näide: Sõltuvused kolme protsesside grupi vahel. Punase värviga on tähistatud protsesside kuulumine ühte loogilisse gruppi, sinise värviga protsesside kuulumine teise loogilisse gruppi ning oranžiga kolmandasse loogilisse gruppi. Protsess_grupp_2 sõltub Protsess_grupp_1 protsessi Protsess_transformeerimine_1_1 edukast käivitamisest. Protsess_grupp_3 sõltub Protsess_grupp_1 protsessi Protsess_transformeerimine_1_1 edukast käivitamisest ning Protsess_grupp_2 protsessi Protsess_transformeerimine_2_2 edukast käivitamisest.

Andmelaadimiste planeerimisel võib kasutada mitut lähenemist. Nendest esimene ja lihtsam on protsesside käivitamine kindlal ajahetkel. Näiteks on allikasüsteemiga SLA (*Service level agreement*) alusel kokku lepitud, et andmed on saadaval kindlaks kellaajaks ning andmeaida protsessid võivad käivituda. Sellise lähenemise plussiks on lihtsus, kuid juhul kui allikasüsteemis andmed mingil põhjusel saadaval ei ole, tekib konflikt ning andmelaadimiste üks osa seiskub. Sellise olukorra vältimiseks on võimalik kasutada teist lähenemist, kus andmeaida protsess käivitub mõne märguande peale, mis teavitab, et andmed allikasüsteemis on saadaval laadimiseks. Märguande saatmine sõltub juba osapoolte vahelisest kokkuleppes. Näiteks ettevõtte väline allikasüsteem võib saata faili, mida süsteem ootab ning kui fail jõuab määratud kausta, alustab koheselt vastav andmelaadimiste protsess. Ettevõtte sees võib asi toimuda sarnaselt või ka võimalusel kasutada ühtset ettevõtte sisest automatiseerimise ja planeerimise tööriista, kus on võimalik luua seoseid protsesside vahel. Eelpool nimetatud kaks

lähenemist on enim kasutatavad ning õige lähenemise valimine sõltub juba ettevõtte vajadustest. Kindlasti kasutatakse ka nende kahe lähenemise kombinatsiooni, mis võimaldab paindlikult seada ülesse kogu andmelaadimiste loogika ja arhitektuur [4].

2.3 Laadimiste ressursi jaotamine

Andmelaadimiste korral jaguneb ressursid peamiselt kahe suure komponendi vahel. Suure koormuse saavad andmelao aluseks olev andmebaasisüsteem ja ETL tööriist, kuna selles tehakse suur osa andmete agregeerimise loogikast. Seetõttu on oluline, et andmelaadimiste protsess oleks ühtlaselt jaotatud. Diplomitöös käsitlevas ettevõttes põhiline osa laadimistest peab toimuma igapäevaselt alates keskööst kuni kella kaheksani hommikul. Mõningad protsessid algavad juba tööpäeva lõpus ning tööpäeva lõpp kuni kesköö on ajalises plaanis sobiv vahemik, kuhu saab planeerida päevavahetusest sõltumatuid protsesse. Tööpäeva alguseks on oluline, et andmelaadimiste protsessid oleksid lõpetanud töö, sest siis alustavad kasutajad andmete kasutamist. Vastasel juhul pärsiks suur päringute arv nii kasutajate kui andmelaadimiste kiirust.

Ressursi paremaks jaotamiseks on oluline analüüsida andmebaasi ning ETL vahendi koormust. Monitooring peaks olema järjepidev ning võimaldama parendada kogu andmelaadimiste protsessi. Ettevõttes hetkel kasutusel olev tööriist ei võimalda mõistlikul viisil piirata töös olevate protsesside hulka. Näiteks saab tuua, et vahemikus kell 01:00-01:30 käivitatakse mitmed protsesside grupid. Samas ei piirata seda, et korraga võib töös olla piiratud arv protsesse. Selle lahendamiseks on võimalik muuta aega, millal protsessid alustavad, kuid see ei ole piisavalt efektiivne. Üks võimalik parem lahendus võiks olla näiteks seadistatav piirang. Kui piirang oleks viis korraga töötavat andmete laadimise protsessi, siis kuues läheks järjekorda ning käivituks kohe, kui üks koht vabaneb. Järjekord peaks olema *FiFo* meetodil ehk esimesena järjekorda lisatud protsess väljub esimesena ning seda kohe, kui üks töös olnud viiest andmelaadimise protsessist lõpetab. Sarnaselt saaks lahendada ka probleemi, kus ühte allikasüsteemi võib sattuda korraga liiga palju päringuid.

Järgnevates peatükkides keskendutakse just andmelaadimiste planeerimise tööriistale. Esmapiilgul andmeida arhitektuuri kirjeldades ei pruugita sellele suurt tähelepanu pöörata, kuid suuremates andmeaitades on tegemist olulise komponendiga.

3 Nõuded tööriistale

Uue tööriista valimise eelduseks on nõuete kirjeldamine, mis on üks osa kogu projektist, mille käigus valitakse finantsettevõttele uut andmelaadimiste planeerimise tööriista. Olemasolevale tööriistale seatud nõuded ei ole tänapäevased ning seetõttu tuleb need uuesti kirjeldada. Üldised nõuded uuele tööriistale on järgnevad:

- Võimaldab laadimiste planeerimist
- Monitooringu mooduli olemasolu
- Sõltuvuste haldamise võimekus
- Ressursi haldamise võimekus
- Eelistatult veebipõhine kasutajaliides

Täpsemad funktsionaalsed ning mittefunktsionaalsed nõuded on kirjeldatud järgnevates alampeatükkides. Alampeatükkides vaadeldakse nõudeid, mis on andmelaadimiste spetsiifilised. Välja on jäetud üldised ettevõtte süsteemidele seatavad nõuded, sest tegemist on konfidentsiaalse infoga ning diplomitöös ei kajastada.

Dai Clegg poolt kirjeldatud MoSCoW meetodiga [5], mida kasutatakse prioriteetide määramiseks, märgitakse igale nõudele prioriteet. Meetodi järgi kasutatakse järgnevat skaalat:

- Peab olema (1)
- Peaks olema (2)
- Võiks olla (3)
- Pole oluline (4)

3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded uuele tööriistale on kokku pandud koostöös andmelao halduse ja arenduse meeskonnaga. Lisaks on arvesse võetud arendusperspektiivi ning planeerimise ja automatiseerimise parimaid praktikaid [6]. Funktsionaalsed nõuded kirjeldavad, millist funktsionaalsust peab süsteem kasutajale pakkuma [7]. Igale nõudele on lisatud identifikaator hilisemaks viitamiseks. Detailsema ülevaate saamiseks on nõuetel alamkirjeldused, mis peavad olema kõik täidetud, et funktsionaalne nõue vastaks soovitule. Funktsionaalsed nõuded on järgmised:

- Sõltuvuste haldamine (**FN_01**) - Protsesside ning protsessi gruppide vahel peab olema võimalik luua omavahelisi seoseid. Seosed võivad olla erinevad ehk protsess võib sõltuda mõnest teisest protsessist või terveist protsesside grupist
 - **FN_01_1** – Sõltuvuste lisamine
 - **FN_01_2** – Sõltuvuste muutmine
 - **FN_01_3** – Sõltuvuste kustutamine
- Sõltuvuste jälgimine (**FN_02**) - Sõltuvusi peab olema võimalik kasutajale arusaadaval kujul kuvada ning seda kogu ahela ulatuses. Näiteks kui protsesside grupi C eelduseks on grupp B ning protsesside grupi B eelduseks on grupp A, siis peab olema võimalik näha tervet sõltuvuste ahelat. Seda kas graafilisel või mõnel muul esitataval viisil
 - **FN_02_1** – Eelduste kuvamine
 - **FN_02_2** – Järgmiste kuvamine
 - **FN_02_3** – Kogu sõltuvuste ahela kuvamine
- SQL skriptide käivitamine (**FN_03**) - Võimalus lisada ajutisi protsesse, mis käivitavad kokkulepitud meetodil SQL (*Structured query language*) skripte. Käivitamise vajadus võib olla nii ühe- kui mitmekordne või määratud intervalli alusel
 - **FN_03_1** – SQL skriptide kohene käivitamine

- **FN_03_2** – SQL skriptide planeeritud käivitamine
- Taaskäivitamine (**FN_04**) - Vea tõttu katkenud protsessi peab olema võimalik taaskäivitada. Lisaks peab olema võimalik taaskäivitada algusest protsesside gruppi, milles olev protsess vea tõttu katkes
 - **FN_04_1** – Protsessi taaskäivitamine
 - **FN_04_2** – Protsesside grupi taaskäivitamine
- Protsesside haldamine (**FN_05**) – Kasutajaliidese olemasolu, mille kaudu toimub kogu protsesside haldus
 - **FN_05_1** – Protsesside lisamine
 - **FN_05_2** – Protsesside muutmine
 - **FN_05_3** – Protsesside kustutamine
- Erinevates keskkondades protsesside käivitamine (**FN_06**) – Käsupõhiste ning keskkonnapõhiste programmide käivitamine Linux ja Windows keskkondades
 - **FN_06_1** – Käivitamine Linux keskkondades
 - **FN_06_2** – Käivitamine Windows keskkondades
- Protsesside seiskamine (**FN_07**) - Protsesse peab olema võimalik kasutaja sekkumisel katkestada. Katkestamise korral järgmised ahelas olevad protsessid käivituda ei tohi
 - **FN_07_1** – Käivitatud protsessi katkestamine
- Planeeritud protsesside piiramine (**FN_08**) - Juba planeeritud protsesside käivitumist peab olema võimalik kasutajal piirata. Näiteks teatud kellaajal planeeritud protsessi alustamise peatamine ning piirangu maha võtmine sobival hetkel
 - **FN_08_1** – Käivituspiirangu lisamine protsessile
 - **FN_08_2** – Käivituspiirangu eemaldamine protsessilt

- **Monitooring läbi kasutajaliidese (FN_09) – Kasutajaliidese olemasolu monitooringuks. Võimalus vaadata käivituste staatusi, ajalugu ja logisid**
 - **FN_09_1 – Käivituste ajaloo kuvamine**
 - **FN_09_2 – Käivituste logi kuvamine**
 - **FN_09_3 – Aktiivsete protsesside kuvamine**
 - **FN_09_4 – Planeeritud protsesside kuvamine**

- **Konfigureeritav kalender (FN_10) – Võimalus määrata tööpäevaid, pühasid ja puhkepäevaid. Sellest tulenevalt on võimalik planeerida protsesside käivitamise intervalli**
 - **FN_10_1 – Tööpäevade määramine**
 - **FN_10_2 – Puhkepäevade määramine**
 - **FN_10_3 – Pühade määramine**
 - **FN_10_4 – Riigipõhiste kalendrite seadistamine**
 - **FN_10_5 – Kalendri muutmine**
 - **FN_10_6 – Kalendri kustutamine**

- **Mitme kalendri kasutamine (FN_11) – Võimalus kasutada riigipõhiseid kalendreid planeerimisel**
 - **FN_11_1 – Planeerimisel sobiva konfigureeritud kalendri valimine**

- **Konfigureeritav teavituste süsteem (FN_12) – Teavituste süsteemi seadistamise võimalus. Võimalus edastada ettevõtte kesksetele SMS ja e-mail teenustele vajalik info teavituste edastamiseks**
 - **FN_12_1 – Teavituste lisamine**
 - **FN_12_2 – Teavituste muutmine**
 - **FN_12_3 – Teavituste kustutamine**

- Importimise/eksportimise moodul (**FN_13**) – Moodul, mille abil on võimalik defineeritud kujul protsesse ja konfiguratsioone eksportida ning importida. Vajalik migratsiooni tegemiseks ning erinevate keskkondade üles seadmiseks ning nende automatiseeritud haldamiseks
 - **FN_13_1** – Ekspordi moodul seadistuste eksportimiseks
 - **FN_13_2** – Impordi moodul seadistuste importimiseks
 - **FN_13_3** – Keskkondade haldamine läbi vastava mooduli

- Sündmuste põhine protsesside käivitamine (**FN_14**) – Teatud sündmuse põhjal protsessi käivitamine. Selleks võib olla näiteks faili tekkimine defineeritud kausta
 - **FN_14_1** – Sündmuse põhine käivituse lisamine
 - **FN_14_2** – Sündmuse põhine käivituse muutmine
 - **FN_14_3** – Sündmuse põhine käivituse kustutamine

- Ressursside piiramine (**FN_15**) - Võimalus piirata korraga töös olevate protsesside hulka, vastavalt kas arvu või loogilise liigituse järgi
 - **FN_15_1** – Ressursi lisamine
 - **FN_15_2** – Ressursi muutmine
 - **FN_15_3** – Ressursi kustutamine
 - **FN_15_4** – Ressursi koguste defineerimine
 - **FN_15_5** – Ressursi kasutamine

Tabelis 1 on lisatud igale nõudele prioriteet. Kokku on 5 funktsionaalset nõuet, mida tööriist peab kindlasti täitma. Kui prioriteetiga 1 nõuded on täidetud, saab vaadelda iga tööriista kohta madalama prioriteediga nõudeid. Nii on võimalik esialgset tööriistade valimit mõistlikul viisil vähendada. Ükski nõue ei saanud prioriteeti 4.

Tabel 1. Nõuetele määratud prioriteedid

Peab olema (1)	Peaks olema (2)	Võiks olla (3)
FN_01	FN_03	FN_08
FN_02	FN_05	FN_09
FN_04	FN_06	FN_11
FN_07	FN_10	FN_13
FN_14	FN_12	
	FN_15	

3.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete osas on välja toodud ülesandepüstituse spetsiifilised nõuded. Mittefunktsionaalsed nõuded kirjeldavad kuidas, kui hästi või millisele standardile vastates funktsionaalsus peab töötama [7]. Mittefunktsionaalsetele nõuetele ei määrata prioriteeti erinevalt funktsionaalsetest nõuetest, sest kõik kirjeldatud mittefunktsionaalsed nõuded peavad olema täidetud. Mittefunktsionaalsed nõuded on järgmised:

- Kasutajatoe olemasolu (**MFN_1**) – Tootjapoolse kasutajatoe soetamise võimalus
- Hind (**MFN_2**) – Võrdsete või sarnaste tööriistade puhul on valiku tegemisel oluline soodsam hind
- Avatud andmebaas (**MFN_03**) – Avatud andmebaas auditite ning statistika tegemiseks
- Platvorm (**MFN_04**) – Tugi serverplatvormidele Windows Server 2012, Linux Redhat või zOS
- Dokumentatsioon (**MFN_05**) – Detailne dokumentatsioon, mis kirjeldab tööriista kasutamist ja selle võimalusi
- Põhifunktsionaalsus lihtsalt kasutatav (**MFN_06**) – Programmeerimiskeelte oskus ei ole vajalik põhifunktsionaalsuse kasutamiseks

- Arusaadavus (**MFN_07**) – Kasutajaliidese loogiline ning arusaadav ülesehitus
- Referentsid (**MFN_08**) – Tööriista kasutavad suurettevõtted ning kasutajaskond ei ole marginaalne

Kokku on mittefunktsionaalseid nõudeid kaheksa.

4 Andmeida laadimiste planeerimiseks sobivad tööriistad

Eraldiseisvaid andmeaitade laadimiste planeerimiseks mõeldud tööriistu erinevad tarkvara tootjad ei paku. Automatiseerimine ning sõltuvuste haldus hõlmab üha rohkem tervet ettevõtet ning üldisem mõiste on *Workload Automation*. Nimetatud käsitluse juurde kuuluvad mõisted, mida erinevates tööriistades kasutatakse:

- 1) Töö (*Job*) – Käesolevas töös kasutatakse mõistet protsess
- 2) Sõltuvused (*Dependencies*)
- 3) Töövood (*Job streams*) – Käesolevas töös kasutatakse mõistet protsesside grupp
- 4) Kasutajad (*Users*)

Kuna puudub eraldi kategooria andmeaitade jaoks, lisab see keerukust valiku tegemisel, sest teatud nõuetest tulenevast spetsiifikast ei pruugi kõik pakutavad vahendid sobida seatud eesmärki täitma. Autor valib detailsemaks analüüsiks neli erinevat vahendit. Esmase valiku aluseks on Gartneri maagilises ruudus välja toodud tootjate lahendused [8]. Kuigi raport pärineb aastast 2012, ei paista autori hinnangul käesoleval hetkel väga suuri muudatusi selles osas olevat. Lisaks on üks tööriist, VisualCron, võetud väljaspool 2012 raportit. Tegemist on väiksema tootega, kuid võib konkurentsi pakkuda suurte tootjate tööriistadele.

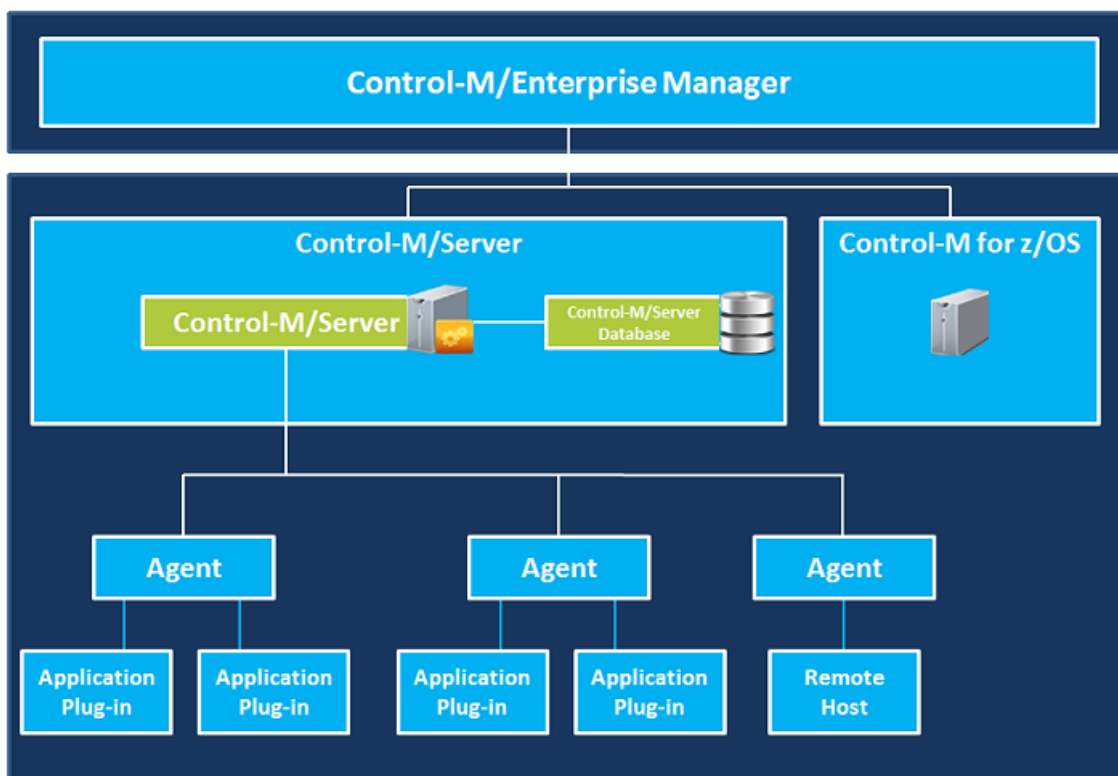
4.1 BMC Control-M

BMC poolt pakutav automatiseerimise tööriist. Gartneri 2012 raporti hinnangul liider antud valdkonnas IBM ees [8]. Tootja poolt välja toodud eelised [9]:

- Võimaldab automatiseerida ja planeerida protsesse kogu ettevõtte IT süsteemides.
- Haldamine ning monitooring toimub läbi keskse kasutajaliidese.
- Lihtne kasutada ning ei nõua spetsiifilisi teadmisi.

- Lai valik mooduleid ja lisaülesannete lahendamiseks.
- Globaalne kalendrite süsteem erinevate kalendrite seadistamiseks.
- Automaatne agentide ja klientsüsteemi paigaldus lihtsustab uuenduste paigaldamist.

Control-M tööriista lihtsustatud arhitektuur on kujutletud Joonisel 5.



Joonis 5. Control-M arhitektuuri joonis [10]. Läbi klientrakenduse suheldakse serveriga ning agendid võivad paikneda erinevates serverites, vastavalt sellele, kuskohas protsesse käivitada soovitakse.

4.2 IBM Workload Scheduler

IBM poolt pakutav tööriist, mis jaguneb kaheks alamtooteks ehk IBM Workload Scheduler ning IBM Workload Scheduler for zOS. Neist esimene omab Windowsi ja Linuxi erinevate versioonide tuge ning teine on mõeldud zOS keskkonnas kasutamiseks.

Tootja on klientidele välja toonud järgnevad tugevused [11]:

Lihtne administreerimine:

- Keskne keskkondade haldus.
- Puudub vajadus paigaldada klientrakendusi kasutaja arvutisse. Erinevad kasutajad saavad tööriista kasutada üheaegselt läbi veebilehitseja.
- Keskne autoriseerimise haldus kasutades SSO (*Single sign-on*) meetodit ja rollipõhine õiguste haldus gruppidele ning kasutajatele.

Tõhus haldus:

- Seadistatavad vaated võimaldavad proaktiivselt monitoorida ja hallata kriitilisi protsesse.
- Võimalus luua ja käivitada raporteid, et koguda statistikat käivituste kohta.
- IT infrastruktuuri optimaalne kasutamine. Võimalus määrata, milliseid infrastruktuuri komponente kasutatakse.

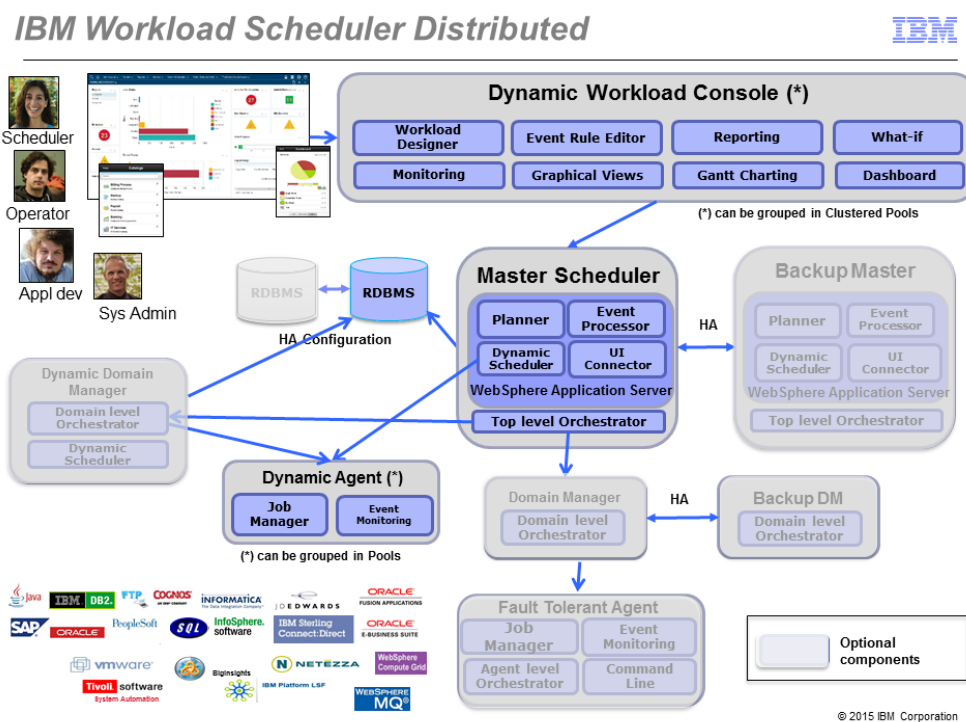
Graafiline liides:

- Graafilised vaated loodud süsteemidest ja ahelatest.
- Võimalus simuleerida plaane ning ahelate jada.
- Mõjuanalüüsi vaade. Läbi loodud ahelate võimalus jälgida mõju ahela ulatuses.
- Graafilise liidese vaateid on võimalik kasutada otse muudatuste tegemiseks.

Turvalisus:

- Turvalise keskkonna seadistamine. Kogu kontroll läbi IBM Workload Scheduler IT Administraatori, mistõttu ei ole vajadust kasutaja arvutis eraldi midagi konfigureerida.
- Kasutajatele ja kasutajagruppidele eraldi õiguste määramine.
- Objektid on kaitstud granulaarsel tasandil, et ligipääs oleks tagatud ainult ettenähtud kasutajatele.

Tööriista arhitektuuri kirjeldab Joonis 6. zOS versioonil on mõningad erinevused, kuid enamused komponendid on samad.



Joonis 6. IBM Workload Scheduler arhitektuuri joonis [12]. Veebipõhine kasutajaliides kasutab Dynamic Workload Console. Master Scheduler kaudu käivitatakse planeeritud protsesse. Agendid protsesside käivitamiseks võivad paikneda erinevates serverites.

4.3 VisualCron

VisualCron on automatiseerimise, integreerimise ja ajastamise klient-server tööriist ainult Windows platvormile. Ettevõtte, kes pakub nimetatud tööriista, toob põhilisteks eelisteks välja järgneva [13]:

- Programmeerimise taust ei ole tööriista kasutamiseks oluline. Läbi kasutajaliidese on võimalik endale sobiv keskkond ülesse seada.
- Objektipõhiline kasutajaliides on terviklik ning lihtsasti õpitav.
- Võimalus integreerida üle 100 erineva tehnoloogia.

- Arenduse aluseks on klientide soovid, mida on võimalik ettevõttele esitada.
- Erinevad logid aitavad kaasa probleemihaldusele ja auditeerimisele.
- Keskne vigade logimine, kuvamine ja parandamine.
- API (*Application programming interface*) – võimalus integreerida teiste süsteemidega.
- Mõistlik hind.

Erinevalt paljudest teistest tootjatest on VisualCroni hinnapoliitika toodud selgelt välja ettevõtte kodulehel. Tabel 2 toob välja hinnad vastavalt litsentsi tüübile [14]. Eraldi on võimalik juurde osta ka tugi ja haldusteenus. Selline avalik hindade välja toomine on valitud tööriistade hulgas ainulaadne, sest teiste puhul lepitakse hind eraldi pakkuja ning tellija vahel kokku.

Tabel 2. VisualCron ning tugiteenuse hinnad

Toode	Hind	Tugi ja haldus
1 serveri litsents	\$449	\$169 iga serveri kohta
5 serveri litsents	\$1699	
<i>Site license</i>	\$2999	\$879
<i>Country license</i>	\$5999	\$1649
<i>World license</i>	\$9999	\$2749

4.4 Stonebranch Universal Controller

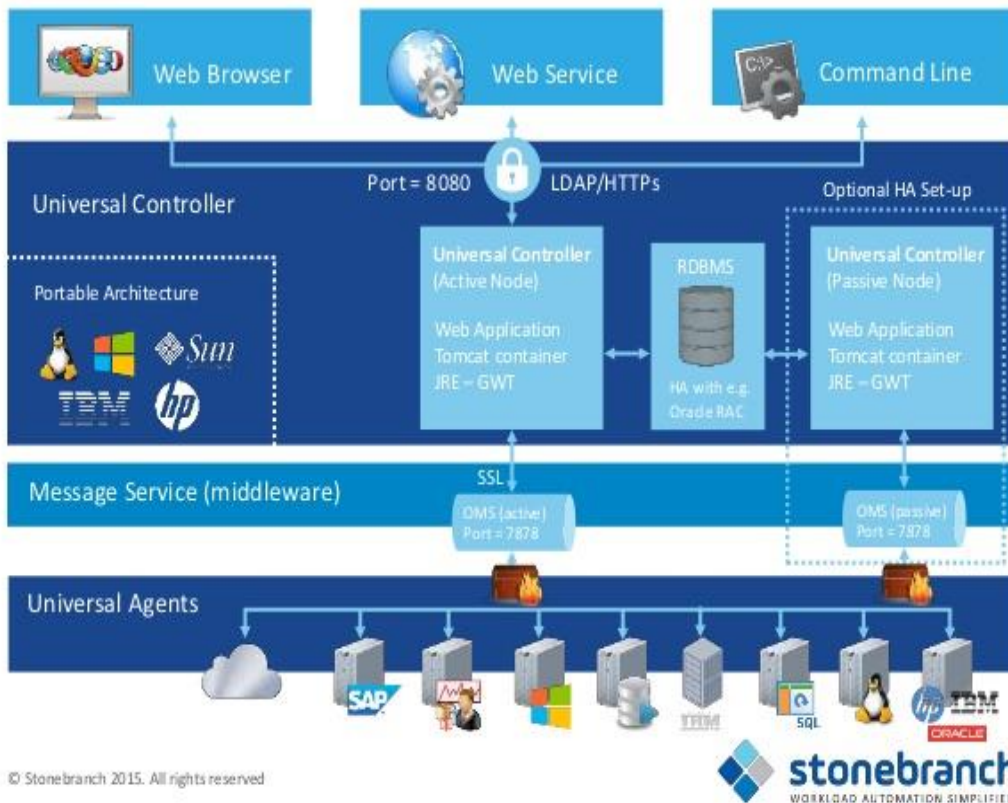
Ettevõtte Stonebrach poolt pakutav tööriist, millel on tugi Windowsi, Linuxi ja Unixi serveritele. Lisaks on olemas ka tugi zOS keskkonnas kasutamiseks. Väljatoodud tugevused, millega ettevõtte oma toodet tutvustab [15]:

- Innovaatiline lahendus võimaldab automatiseerida ettevõtte protsesse ühes kohas.

- Tööriista kasutajaliides võimaldab igal kasutajal seadistada endale sobiv ja vajalik vaade, mille abil jälgida protsesse reaajas. Raportite jaoks on liides, mis võimaldab kasutajal luua talle vajalike raporteid või kasutada olemasolevaid, kohandades neid vastavalt soovile.
- CMDB (*configuration management database*) on Universal Controlleri üks osa. Ärikasutajad saavad defineerida äriteenuseid ning lisada vajalike ettevõtte ressursse. Läbi selle on võimalik ärikasutajatel hiljem jälgida teenuste toimimist ning kasutatavaid ressursse.
- Klasterdamine veakindluse ning kättesaadavuse suurendamiseks. Kõrge teenuse kättesaadavuse tagamiseks saab luua klastrid, et suurendatakse veakindlust ning balanseerida koormust.
- Windows, Linux, Unix või zOS keskkonnad sobivad tööriista paigaldamiseks vastavalt ettevõtte vajadusele. Kõigil platvormidel pakutakse reaajas failide monitooringut ning failide liigutamist kasutades FTP (*File transfer protocol*) või SFTP (*Secure file transfer protocol*) protokolle.
- Süsteemi turvalisus määratakse toote siseselt. Süsteemiadministraator loob kasutajad ja kasutajagrupid, millele määratakse ligipääsuõigused. Seda saab teha väga granulaarsel tasandil, kuni spetsiifilise tegumini välja.

Universal Controlleri arhitektuur koosneb neljast põhikihist ning need on kujutatud Joonisel 7.

Universal Controller



Joonis 7. Universal Controller arhitektuuri joonis [16]. Läbi kasutajaliidese, veebiteenuse või käsurea pöörduakse Universal Controlleri serveri poole. Agendid võivad paikneda erinevates serverites, vastavalt sellele, kuskohast protsesse käivitatakse.

4.5 Infrastruktuuri tugi

Nõuded infrastruktuurile ja toetatavad serveri platvormid ning versioonid on olulised ettevõtte infrastruktuuri osakonnale. Tabelis 3 on välja toodud uusimate tööriista versioonide sobivus serverite ning andmebaasi platvormide osas. Lõplik valitava tööriista kulu koosneb kahest põhilisest osast. Esimene on tööriista hind ja sellele kulumis tugevus. Teise osa moodustab infrastruktuuri kulu ehk serveri ja andmebaasi platvormi kasutamise eest makstav tasu. Tabel 3 ei kajasta kõiki serverite versioone, mida tööriistad toetavad, vaid olulised on uusimad versioonid toetatavast infrastruktuurist.

Tabel 3. Infrastruktuuri tugi.

	BMC Control-M	IBM Workload Scheduler	VisualCron	Stonebranch Universal Controller
Windows Server 2008 R2	+	+	+	+
Windows Server 2012 R2	+	+	+	+
Red Hat Enterprise Linux (RHEL) Server 7	+	+	-	+
SUSE Linux Enterprise Server (SLES) 11	+	+	-	+
z/OS	-	+	-	-
PostgreSQL 9.2.8	+	-	-	-
Oracle 11g	+	+	-	+
Oracle 12c	+	+	-	+
MSSQL 2014	+	-	+	+
MSSQL 2012	+	-	+	+

5 Tööriista valimine

Sobivaima tööriista valimiseks vaadeldakse nõuetele vastavust kõigi nelja tööriista osas. Testimiseks ja funktsionaalsuse proovimiseks pakuvad kõik neli tootjat tasuta 30-45 päeva pikkust võimalust tarkvara kasutada. Lisaks on kõigil kättesaadav põhjalik dokumentatsioon, mis sisaldab detailset infot tööriista ja selle funktsionaalsuse kohta.

5.1 Vastavus funktsionaalsetele nõuetele

Kolmandas peatükis kirjeldatud nõuded on aluseks järgneval valiku tegemisel. Tabelis 4 on välja toodud kõige kõrgema prioriteediga nõuded. Need nõuded peavad olema kõik täidetud ning puuduste korral muude nõudmiste täitmist ei kontrollita.

Tabel 4. Kõrgeima prioriteediga funktsionaalsetele nõuetele vastavus

	BMC Control-M	IBM Workload Scheduler	VisualCron	Stonebranch Universal Controller
FN_01	+	+	+	+
FN_02	+	+	+	+
FN_04	+	+	+	+
FN_07	+	+	+	+
FN_14	+	+	+	+

Kõik neli tööriista täidavad kõrgeima prioriteediga nõuded. VisualCroni loogika erineb mõningal määral kolmest ülejäänud tööriistast.

Tegemist on kogu ettevõtte protsesside automatiseerimiseks mõeldud vahenditega, mille üks oluline osa on sõltuvuste haldus. Seetõttu sobivad need vahendid hästi ka andmelaadimiste planeerimiseks. Järgnevalt on Tabelis 5 välja toodud ülejäänud funktsionaalsed nõuded ning nende vastavus.

Tabel 5. Funktsionaalsetele nõuetele vastavus. Prioriteetidid 2-3.

	BMC Control-M	IBM Workload Scheduler	VisualCron	Stonebranch Universal Controller
FN_3	+	+	+	+
FN_5	+	+	?	+
FN_6	+	+	-	+
FN_8	+	+	+	+
FN_9	+	+	?	+
FN_10	+	+	-	+
FN_11	+	+	-	+
FN_12	+	+	+	+
FN_13	+	+	+	+
FN_15	+	+	-	+

Tabelis 5 välja toodu põhjal ei täida kõiki funktsionaalseid nõudeid VisualCron. Küsimärgiga on tähistatud kasutajaliidesega seotud nõuded. Kuna VisualCron vahendil puudub veebiliides, tähendaks see vajadust iga kasutaja arvutisse kliendi poolse tarkvara komponendi paigaldamist ning uuenduste haldamist. Selles osas on veebipõhine kasutajaliides eelistatum.

Funktsionaalsed nõuded täidavad kolm tööriista, mis on ka Gartneri maagilises ruudus liidritena välja toodud.

5.2 Vastavus mittefunktsionaalsetele nõuetele

Mittefunktsionaalsetele nõuetele vastavus on toodud välja Tabelis 6.

Tabel 6. Mittefunktsionaalsete nõuete vastavus

	BMC Control-M	IBM Workload Scheduler	VisualCron	Stonebranch Universal Controller
MFN_1	+	+	+	+
MFN_2	-	+	-	-
MFN_3	+	+	+	+

MFN_4	+	+	+	+
MFN_5	+	+	+	+
MFN_6	+	+	+	+
MFN_7	+	+	+	+
MFN_8	+	+	+	+

Mittefunktsionaalsed nõuded on täidetud kõigi valimis olevate tööriistade poolt. MFN_2 puhul on tabelis pluss märk ainult ühel, IBM Workload Scheduleril, mille hind on konkurentidest soodsam. Konfidentsiaalsuse tõttu ei ole võimalik hindasid avalikult välja tuua. VisualCroni puhul on hinnapoliitika varasemalt detailselt kirjeldatud. Teiste toodete puhul on pakkumised ettevõtetele personaalsed sõltuvalt spetsiifilistest nõudmistest.

5.3 Tööriista valik

Diplomitöö aluseks olevas finantsettevõttes on kasutusel ettevõtete BMC, IBM ja Stonebranch muud tooted. Antud juhul on samade tingimuste korral IBM Workload Scheduler sobivaim valik. Nimetatud tööriist vastab kõigile seatud funktsionaalsetele ja mittefunktsionaalsetele nõuetele ning on hinna poolest soodsam kui konkurendid.

Kõik kolm sobivad täitma seatud spetsiifilist ülesannet ning neid tööriistu saaks kasutada ettevõttes palju laialdasemalt. Automatiseerides protsesse ühtselt ühes keskkonnas, tekib võimalus seotud süsteemidel töötada automatiseeritult aja- ja kuluefektiivselt. Koostatud analüüsi tulemusena on tekkinud ideed, kuidas kasutada valitud tööriista hiljem näiteks raportite automaatseks genereerimiseks koheselt, kui andmebaasi objektides on andmed laetud. Lisaks laadida ettevõtte väliselt saabuvad failid kohe, kui koostööpartnerilt on failid saabunud. Need on vaid mõned üksikud näited, kuidas rakendada valitud tööriista efektiivsuse suurendamiseks.

Järgmises peatükis lähtutakse migratsiooniplaani tegemisel valitud tööriistast.

6 Migratsiooniplaan

Migratsiooniplaani eesmärk on tagada sujuv üleminek uue tööriista kasutusele võtmiseks. Mõne tuhande protsessi käsitsi sisestamine on ajamahukas ja riskantne töö. Selle käigus peaks sisestama kõik protsessid, protsesside grupid ja sõltuvused. Need omakorda sisaldavad detailset infot käivitusaegade ja tegumite (*Task*) kohta. Sellise mahukuse juures on väga lihtne vigu teha. Migratsioon peaks olema võimalikult automatiseeritud, kuid eesmärk ei ole 100% automatiseerimine, kui selleks tuleb arendada eraldi mahukas programm. Alan Raddingi artiklis [17] välja toodud viis parimat praktikat andmete migreerimisel on hea sisend, millele antud migratsiooni raames tähelepanu pöörata. Tabelis 7 on toodud käesoleva migratsiooni tegevused, mis on vastandatud parimate praktikatega.

Tabel 7. Praktikad ning vastavad tegevused

Praktika	Tegevus
Analüüsida ja aru saada olemasolevatest andmetest. Andmete asukoht, millisel kujul need on salvestatud ja millisele kujule tuleb teisendada.	Analüüsida, millised andmed on olemas ja mida saab kasutada. Luua selge kaardistus vana ja uue süsteemi andmete vahele koos vajalike formaadi muudatustega.
Töötle ning kasuta võimalust puhastada olemasolevaid andmeid.	Analüüsida, kas on dubleerivaid ning mitte kasutusel olevaid andmeid. Uude tööriista sellised andmeid mitte importida.
Migreeri andmed süstemaatilisel viisil ning parimal võimalikul ajal.	Luua süsteem andmete migreerimiseks. Kuna tegemist on uue süsteemi kasutusel võtmisega, siis ranget ajalist piirangut ei ole.
Testi ja valideeri migreeritud andmeid, et tagada soovitud väljund. Ilma testimata ei ole võimalik andmete terviklikkuses	Koostada testiplaan tulemi testimiseks.

kindel olla.	
Auditeeri ning dokumenteeri protsessi, et tagada selge arusaam, kuidas ja mida täpselt tehti.	Dokumenteerida ja säilitada dokumentatsioon ettevõttes ette nähtud korras.

Tabeli 7 tegevuste mahukus on suur ning autori hinnangul moodustab see ligi poole uue tööriista valiku projektist. Järgmises kolmes alampeatükis analüüsitakse kolme moodust andmete migreerimiseks. Nendest esimene on import võimaluse kasutamine IBM Workload Scheduler tööriistas ning teine *Datamigrate* funktsionaalsuse põhine import. Kolmas võimalus on otsene kahe andmebaasi vaheline andmete migreerimine.

6.1 Import moodul

IBM Workload Scheduler sisaldab import moodulit, mis võimaldab keskkonda ülesse seada ilma ükshaaval vajalike protsesse sisestamata. Moodulit saab samal viisil kasutada ka erinevate keskkondade ülesse seadmiseks. Importimiseks kasutatav *ZIP* fail peab sisaldama kolme XML faili:

- *name_Definitions.UTF8.xml* – Sisaldab kogu keskkonna andmestikku. Tegemist on põhifailiga, kus on protsesside, protsessi gruppide, sõltuvuste, käivitusaegade, tüüpide, käskude jne info
- *name_Mapping.UTF8.properties* – Võimaldab objektide nimedele luua seoseid, mis importimisel uude keskkonda asendatakse. Nii ei pea *name_Definitions.UTF8.xml* failis hakkama muudatusi tegema, sest selles faili struktuur on keerulisem ning lihtne on jätta midagi märkamata
- *name_SourceEnv_reference.txt* – Sisaldab infot keskkonna kohta, mida soovitakse kasutada. Näiteks võib samas serveris olla mitu keskkonda

Joonisel 8 on näide ühe protsessi kohta. See koosneb kolmest elemendist *JobDefinition*, *task* ja *recovery*. *JobDefinition* elemendis on atribuudid:

- *name* – Protsessi nimi

- *successCondition* – Millisel tagastatava koodiga oli käivitus edukas
- *workstation* – Keskkond

Task elemendis on olulised atribuudid tegumi kohta:

- *cmd* – Kas käivitatakse käsk otse
- *exec* – Milline käsklus käivitatakse
- *type* – Näitab millist tüüpi tegumiga on tegemist

Recovery elemendis on atribuut *action*, millega määratakse tegevus *successCondition* korral. Antud juhul, kui selle väärtus ei ole 0, jääb protsess seisma.

```
<model:JobDefinition name="$JOB_PROTSESS_TRANSFORMEERIMINE_2_1$"
successCondition="RC=0" workstation="$DEV$"
  <model:task cmd="false" exec="&lt;?xml version='1.0'
encoding='UTF-8'>?&#xA;&lt;jsdl:jobDefinition
xmlns:jsdl='http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl'
xmlns:jsdle='http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle'
name='&quot;executable&quot;'>&#xA;  &lt;jsdl:application
name='&quot;executable&quot;'>&#xA;    &lt;jsdle:executable
interactive='&quot;false&quot;'
workingDirectory='&quot;/home/dev/execute&quot;'>&#xA;
&lt;jsdle:script suffix='&quot;&quot;'>execute.bat
Protsess_transformeeringine_2_1&lt;/jsdle:script>&#xA;
&lt;/jsdle:executable>&#xA;
&lt;/jsdl:application>&#xA;&lt;/jsdl:jobDefinition>" interactive="false"
jsdlTask="true" type="executable"/>
  <model:recovery action="stop"/>
</model:JobDefinition>
```

Joonis 8. Näide: Protsessi Protsess_transformeeringine_2_1 definitsioon koos oluliste atribuutidega.

Joonisel 9 on näide ühe protsesside grupi kohta, milles on kolm protsessi siseste sõltuvuste ning ühe välise sõltuvusega Protsess_grupp_1 protsessile Protsess_transformeeringine_1_1. XML koosneb järgmistest elementidest:

- *runcycles* – Sisaldab alamelemente ja atribuute, millega määratakse ära protsesside grupi käivitamise intervall ja kasutusel olev kalender
- *predecessor* – Sisaldab atribuute sõltuvuste kohta
- *jobs* – Iga alamelemendi *job* sees on üks protsessi grupiga seotud protsess, mida kirjeldavad erinevad atribuudid. Lisaks on iga *job* elemendi sees sõltuvuste info.

```

<model:JobStream carryforward="false" draft="false" iskey="false" limit="-1" name=
"$JOBSTREAM_PROTSESS_GRUPP_2$" onrequest="false" priority="100" workstation="$DEV$" >
  <model:runcycles>
    <model:rule except="false" icalendar="FREQ=DAILY;INTERVAL=1;BYWORKDAY" name=
"DAILY">
      <model:restrictions>
        <model:start time="200 0" timeDependent="false"/>
      </model:restrictions>
      <model:subsetOperator>OR</model:subsetOperator>
    </model:rule>
  </model:runcycles>
  <model:matching>
    <model:previous/>
  </model:matching>
  <model:restrictions/>
  <model:dependencies>
    <model:predecessor target=
"$DEV#$JOBSTREAM_PROTSESS_GRUPP_1$.JOB_PROTSESS_TRANSFORMEERIMINE_1_1$" >
      <model:matching/>
    </model:predecessor>
  </model:dependencies>
  <model:jobs>
    <model:job confirmed="false" defintion="$DEV#$JOB_PROTSESS_KONTROLL_2_1$"
isCritical="false" iskey="false" isNoOperation="false" name=
"$JOB_PROTSESS_KONTROLL_2_1$" priority="10">
      <model:restrictions/>
      <model:dependencies/>
    </model:job>
    <model:job confirmed="false" defintion="$DEV#$JOB_PROTSESS_LAADIMINE_2_1$"
isCritical="false" iskey="false" isNoOperation="false" name=
"$JOB_PROTSESS_LAADIMINE_2_1$" priority="10">
      <model:restrictions/>
      <model:dependencies>
        <model:predecessor target="$JOB_PROTSESS_KONTROLL_2_1$"/>
      </model:dependencies>
    </model:job>
    <model:job confirmed="false" defintion=
"$DEV#$JOB_PROTSESS_TRANSFORMEERIMINE_2_1$" isCritical="false" iskey="false"
isNoOperation="false" name="$JOB_PROTSESS_TRANSFORMEERIMINE_2_1$" priority="10">
      <model:restrictions/>
      <model:dependencies>
        <model:predecessor target="$JOB_PROTSESS_LAADIMINE_2_1$"/>
      </model:dependencies>
    </model:job>
  </model:jobs>
</model:JobStream>

```

Joonis 9. Näide: Protsside grupi Protss_grupp_2 definitsioon koos kolme protsessi ja sõltuvustega.

Protssesse on kokku paartuhat ning protsessigruppe ligikaudu sada koos erinevate sõltuvustega. Loodavad XML failid tuleksid mahukad, kuid oleksid migratsiooni raames korduvalt kasutatavad. Joonistel 8 ja 9 ei ole toodud välja kõik võimalikud elemendid ja atribuudid, mida migratsiooni jaoks on vaja kasutada. Need sisaldavad põhikomponente, mis on olulised vähemalt osaliseks migratsiooniks.

6.2 Datamigrate

Datamigrate võimaldab keskkonda ülesse seada kasutades kindla struktuuriga teksti faile. Erinevaid teksti faile on kokku üheksa ning need sisaldavad infot topoloogia, käskude, kalendrite, parameetrite, ressursside, kasutajate, töötajate, tööde ning plaanide kohta. Olemasolevast IBM Workload Scheduler keskkonnast failide genereerimiseks kasutatakse *composer extract* käsuri *extract* käsuga ning failide importimiseks *datamigrate* käsku erinevate parameetritega. Faili struktuuridest parema ülevaate saamiseks on autor arenduskeskkonna IBM Workload Scheduleris loonud kolm protsesside gruppi sõltuvustega, mida on kujutatud eelnevalt Joonisel 3. Käsuga „*extract jobstream dev#@:full*“ luuakse üks fail protsessi gruppide, protsesside ja sõltuvustega. Faili struktuur ja sisu on välja toodud Lisas 1.

Iga protsessi grupi juures on parameetrid käivitus tsükli kohta ning sõltuvuse olemasolul *FOLLOWS* parameeter. Protsesside juures on käivitus käsklus parameeter *TASK*, eduka käivituse tunnuseks olev tagastuskood *RCCONDSUCC* ning tegevus protsessi katkemisel parameetriga *RECOVERY*. Faili struktuur on oluliselt lihtsam kui alampeatükis 6.1 välja toodud XML faili struktuur.

6.3 Andmebaasi põhine migratsioon

Andmebaasi põhise migratsiooni tegemiseks on vajalik aru saada lähteandmete ehk olemasoleva süsteemi andmete struktuurist ning uue vahendi andmete struktuurist andmebaasis. Olemasoleva süsteemi andmed on MS SQL server andmebaasis väga lihtsal kujul. Iga protsessi juures on kirjeldatud protsessi nimi, kuuluvus gruppi, järjekord grupis, käivituskäsklus, käivitusintervall ning mõningad lisaatribuudid. IBM Workload Scheduler kasutab Oracle andmebaasi ning see koosneb ligi üheksakümnest tabelist ja neljakümnest vaatest (*View*). Tabelid on läbi ID (*Identifier*) väärtuste relatsioonilised ning ID on *RAW* formaadis.

Migratsiooni tegemiseks, mis põhineb andmebaasi andmete siirdamistest erinevate andmebaasi platvormide vahel, tuleks kasutada mõnda tööriista. Ettevõttes kasutusel olevad ETL tööriistad ilmselt võimaldaksid seda, kuid sellise programmi programmeerimine oleks liiga ajamahukas ettevõtmine. Oracle andmebaasi andmete

õigetesse objektidesse laadimine ning nendevahelist relatsioonide tekitamine nõuaks kogu struktuuri täpset selgeks tegemist.

Andmebaasi põhine migratsioon võiks olla mõeldav, kui relatsioonide arv oleks väiksem ning seosed oleks väärtuste, mitte ID põhised. Kuna migratsioon on ühekordne, ei omaks andmete migreerimiseks kirjutatav programm hilisemat väärtust. *Datamigrate* funktsionaalsuse või XML import/eksport kasutamine oleks olulised hilisemate keskkondade ülesse seadmisel ning keskkondade varundamisel.

6.4 Migratsiooni tegevused

Andmete migreerimiseks uue vahendi kasutusele võtmiseks on mitu erinevat meetodit. Analüüsi põhjal peaks sobivaim olema *Datamigrate* funktsionaalsuse kasutamine ning failide loomine etteantud struktuuri põhjal. Migratsiooniplaan võiks jaguneda seitsmesse etappi:

- 1) Olemasoleva MS SQL serveri andmetabelite üks-ühele viimine IMB Workload Scheduler andmebaasi. Selle tulemusena oleks võimalik vana tööriista loogikat hiljem vajadusel otse Oracle andmebaasist vaadelda ning analüüsida. Üks sobiv viis oleks MS SQL serverist tabelite sisu importida Excelisse ning järgmise sammuna import Oracle andmebaasi. Tulemuseks peab olema muutmata kujul andmete migratsioon. Dokumenteerida tuleb kogu tegevus.
- 2) Andmefaili genereerimine Lisas 1 toodud kujul või alampeatükis 6.1 välja toodud XML failidena. Soovitav oleks kasutada siiski andmefaili loomist märksa lihtsama struktuuri tõttu. Loodavatest failidest peaks välja jääma kõik mitteaktiivsed kirjed vanast süsteemist.
- 3) Import IMB Workload Scheduler tööriista. XML korral tuleb kasutada import moodulit, andmefaili korral *Datamigrate* funktsionaalsust. IBM hinnangul ei ole mõne tuhande protsessi korraga importimine probleem ning faile ei pea poolitama hakkama.
- 4) Testida käivituste tsükleid paralleelselt uues ja vanas tööriistas, et veenduda migratsiooni protsessi õigsuses. Esialgu peaks välistama muudatused olemasolevas loogikas.

- 5) Optimeerida sõltuvusi ning lisada ressursid protsessidele. Eelduseks on analüüs, millised ressursid on vajalik tekitada ning millised sõltuvused tagavad ajaliselt optimaalsemad käivitused.
- 6) Migreerida testkeskkond toodangusse. XML korral kasutada eksport/import moodulit, faili korral *Datamigrate* funktsionaalsust.
- 7) Seisata vana toodangu keskkond ning alustada uue tööriista kasutamist. Valideerida õigsust ning vajadusel teha muudatusi.

Eelpool välja toodud seitse tegevuskava punkti on diplomitöös üldistavad ning täpne migratsiooni plaan kirjeldatakse ettevõttes detailse dokumendina koos varuplaaniga.

7 Järelanalüüs

Järelanalüüsi eesmärk on anda hinnang valiku tegemise protsessile ning hinnata saavutatud tulemust. Tarkvara valimise protsessi kirjeldavad kümme sammu [18] ning antud diplomitöö tulemust on võimalik nende järgi tagantjärele hinnata. Sarnaste tarkvara valiku projektide korral saab tuleviks lähtuda tehtud tööst ning vajadusel parendada valiku protsessi.

Ülesande püstituse üks olulisemaid punkte oli, et olemasolev tööriistaga ei ole andmete laadimine piisavalt efektiivne. Uue tööriista kasutusele võtmisest tulenevat kasu saab hinnata mitmes aspektis. Andmelaadimiste korral on ajaline võit äärmiselt oluline, sest suurem osa protsessist peab toimuma kindla ajaperioodi vältel. Ajalise kokkuhoiu eelduseks on protsesside käivitamine koheselt, kui allikasüsteemides on andmed valmis ning õiged seosed protsesside vahel. Uue tööriista kasutusele võtmisega saavutatud ajalise kokkuhoiu simuleerimiseks ühe ahela näitel on autor valinud Bizagi Process Modeler tarkvara. See võimaldab simuleerida protsesside tööd [19] kasutades vana tööriista loogikat ning uue tööriista võimalusi.

7.1 Valiku protsessi hinnang

Tarkvara valiku tegemise protsessi kümme etappi [18] annavad ette plaani, mille jälgimine peaks kaasa aitama soovitud tulemuse saavutamisel. Tehtud tööle hinnangu andmiseks vaadeldakse kõiki kümme etappi järgnevates alampeatükkides ning iga etapi juures antakse hinnang selle täitmisele.

7.1.1 Projekti meeskonna loomine ja tegevuskava

Projekti meeskond ja tegevuskava on aluseks kogu projektile. Seetõttu on oluline teha kohe alguses võimalikult hea valik projekti liikmete osas ning luua ühtne arusaam oodatavast tulemusest. Tähtis on kaasata erinevaid osapooli, vältimaks olukorda, kus projekti lõppedes ei leia valitud uus tarkvara kasutust. Oluline on leida inimene, kes projekti veab ja paneb paika tegevuskava. Kaasata tuleb IT poole spetsialistid ja väga tähtis on kaasata inimesed, kes tarkvara kasutama hakkavad. Väga suurte ja kõrge

maksumusega projektide puhul võib olla mõistlik kaasata ettevõtte väliseid konsultante, kes aitavad projekti edule kaasa. Samas peab meeskond koosnema ainult inimestest, kes loovad sinna väärtust.

Andmeaida laadimiste planeerimise tööriista valiku projekti puhul koosnes põhimeeskond kolmest inimesest: projektijuht, diplomitöö autor ning osakonna arhitekt. Lisaks kaasati projekti vältel inimesi, kes uut tööriista kasutama hakkavad ja arendusmeeskonna inimesi, kes aitasid kaasa nõuete kirjeldamisel. Projekti meeskonna valiku ning tegevuskavaga võib rahule jääda, sest suuremaid takistusi ei esinenud. Kõik vajalikud osapooled olid kaasatud.

7.1.2 Informatsiooniallikate kaardistamine

Enne uute lahenduste hindamist tuleb kokku koguda olemasolev informatsioon. Dokumentatsioon, koosolekute protokollid, kirjavahetused ja muu sarnane on abistavaks sisendiks uue tööriista valimisel. See on oluline järgmiseks sammuks - üle vaadata kasutusel olevad protsessid.

Andmete kogumist andmelaadimiste planeerimise tööriista valiku projektis saab hinnata rahuldavaks. Osaliselt jäi koosolekutel kokku lepitu dokumenteerimata ning see tekitas projekti käigus mõningat segadust. Ka olemasolev dokumentatsioon vana süsteemi kohta oli puudulik.

7.1.3 Olemasoleva protsessi ülevaatamine

Uue tööriista valiku eelduseks on arusaamine olemasolevast protsessist ja kitsaskohtadest, mida on võimalik parendada. Läbi tuleks vaadata kõik protsessi osad ning aru saada, kuidas kasutusel olev vahend protsessile vastab ning millised on puudused. Protsesside ülevaatamise tulemus on heaks sisendiks uue tööriista nõuete kirjeldamiseks.

Protsesside analüüsi ning vajaduste kirjeldamine andmelaadimiste planeerimise tööriista valiku projektis võib hinnata heaks. Olemasolev protsess vaadati põhjalikult läbi ning kaardistati uued vajadused, mis hiljem olid aluseks nõuete kirjeldamisel.

7.1.4 Äri vajaduste prioritseerimine

Äri vajaduste prioritseerimise osa aitab selgust saada, millist funktsionaalsust tarkvara peab pakkuma. Nõutav funktsionaalsus tuleks kirjeldada ja anda prioriteedid sobival skaalal. Tihti läheb nõutava funktsionaalsuse nimekiri pikaks. Sellisel juhul aitavad määratud prioriteedid keskenduda esialgu ainult kriitilisele osale.

Alampeatükkides 7.1.3 ja 7.1.4 välja toodud tegevused koondusid andmelaadimiste planeerimise tööriista valiku projektis üheks tegevuseks. Otseselt ei oma valitav tööriist äri pakutavat funktsionaalsust. Tööriist on osaks IT protsessile, mille käigus äri kasutajatele andmed luuakse. Siiski olid kaasatud kasutajad, kes hakkavad valitava tööriistaga igapäevaselt kokku puutuma ning nemad aitasid prioritseerimisel kaasa.

7.1.5 Valiku eelpiiramine

Sõltuvalt projektist võib pakutav tööriistade hulk, mis eeldatavasti peaks täitma seatud nõudeid, olla väga lai. Kõigi läbi analüüsimine ei ole otstarbekas ja muudab projekti pikaajaliseks. Valiku piiramiseks saab kasutada olemasolevaid uurimistöid ning hinnanguid, mida pakuvad erinevad veebilehed.

Gartneri maagiline ruut töövoogude automatiseerimise osas oli üheks aluseks käesolevas projektis. Sealne analüüs toob välja valdkonna liidrid, mis võeti ka valiku tegemisel analüüsimiseks. Automatiseerimise funktsionaalsust pakuvad mitmed tööriistad, kuid andmeaitade spetsiifilisi tööriistu eraldi kusagil välja ei tooda. Valimisse võeti ka üks väiksem tööriist, et võrrelda seda liidritega ning saada selgusele, kas kõik automatiseerimise vahendid sobivad seatud ülesannet täitma.

7.1.6 Tarkvara pakkujate hindamine

Tarkvara tootjad ja pakkujad omavad tähtsat rolli uue tarkvara kasutusele võtmisel ja kasutajatoe pakkumisel. Kui nõuded on paigas, siis järgmine samm oleks pakkujatega koos üle vaadata pakutav tarkvara ja hinnata selle sobivust. Kohtumiste ja ühiste koosolekute tulemusena peab olema kindel, et pakutav tarkvara sobib täitma seatud nõudeid. Enamasti on pakkujad huvitatud kohtumistest, sest see võimaldab neil oma toodet kliendile täpsemalt tutvustada ning veenda klienti just nende toote sobivuses.

Andmelaadimiste planeerimise tööriista valiku projektis ei kohtunud kõigi pakkujatega. Osa küsimusi sai lahendatud e-maili teel ning kuna kõigi tööriistade dokumentatsiooni võib pidada väga heaks, puudus selleks otsene vajadus. Samas leiab autor, et sellist lähenemist võiks järgmiste projektide raames kasutada.

7.1.7 Võimalike pakkujate hindamine

Pakkujate hindamiseks võib küsida teistelt toote kasutajatelt tagasisidet. Üks võimalus on küsida tarkvara pakkujalt ettevõtteid, kes juba tarkvara kasutavad ja siis pöörduda küsimustega nende poole. Samas ei väljasta tarkvara pakkuja ilmselt neid firmasid, kellel esineb tootega probleeme ning rahulolu on madal. Seetõttu võib ise proovida leida kasutajaid tagasiside saamiseks.

Võimalike pakkujate hindamist käesoleva projekti raames ei kasutatud. Kuna BMC, IBM ja StoneBranch teised tooted on ettevõttes kasutuses, siis ei peetud seda vajalikuks. VisualCron ei täitnud kõiki seatud nõudeid ning seetõttu puudus vajadus pakkujast hinnata. Märkusena võib lisada, et VisualCron meeskonna kasutajatugi oli väga kiire vastama küsimustele ja pakkus igakülgset abi.

7.1.8 Otsuse tegemine

Iga eelnevalt kirjeldatud samm peaks aitama otsuse tegemisele kaasa. Uue tööriista valimine peaks olema mitme etapiline loogiline protsess. Võimalik on valida uus tööriist ka ühe koosolekuga paari ettevõtte töötaja vahele, kuid see ei pruugi tagada soovitud tulemust.

Otsuse tegemine IBM Workload Scheduleri kasuks on kirjeldatud peatükis 5.3. Vaadeldi nõuete täitmist ja peale projekti sisest valiku tegemist tegi IBM demo, et kinnitada nõuetele vastavust. Hinnal oli otsuse tegemisel küllaltki suur osakaal, sest sarnast funktsionaalsust pakkusid teisedki tööriistad.

7.1.9 Implementatsiooniplaani loomine

Uue tarkvara kasutusele võtmisel peab olema plaan. Olemasoleva tarkvara välja vahetamise ning täiesti uue tarkvara kasutusele võtmise plaanid võivad olla erinevad. Koostöö ja ühtne arusaam tarkvara pakkujaga on aluseks heale implementatsiooniplaanile. Mõlemad osapooled peavad ühtselt mõistma saavutatavat eesmärki.

Uue andmelaadimiste planeerimise tööriista valiku projekti sisendiks oli olemasoleva tööriista välja vahetamine ning migratsiooni tegemine uue tööriista kasutusele võtmiseks. IBM pakkus tuge ja vajalikku abi migratsiooni tegemiseks ning tööriista implementeerimiseks.

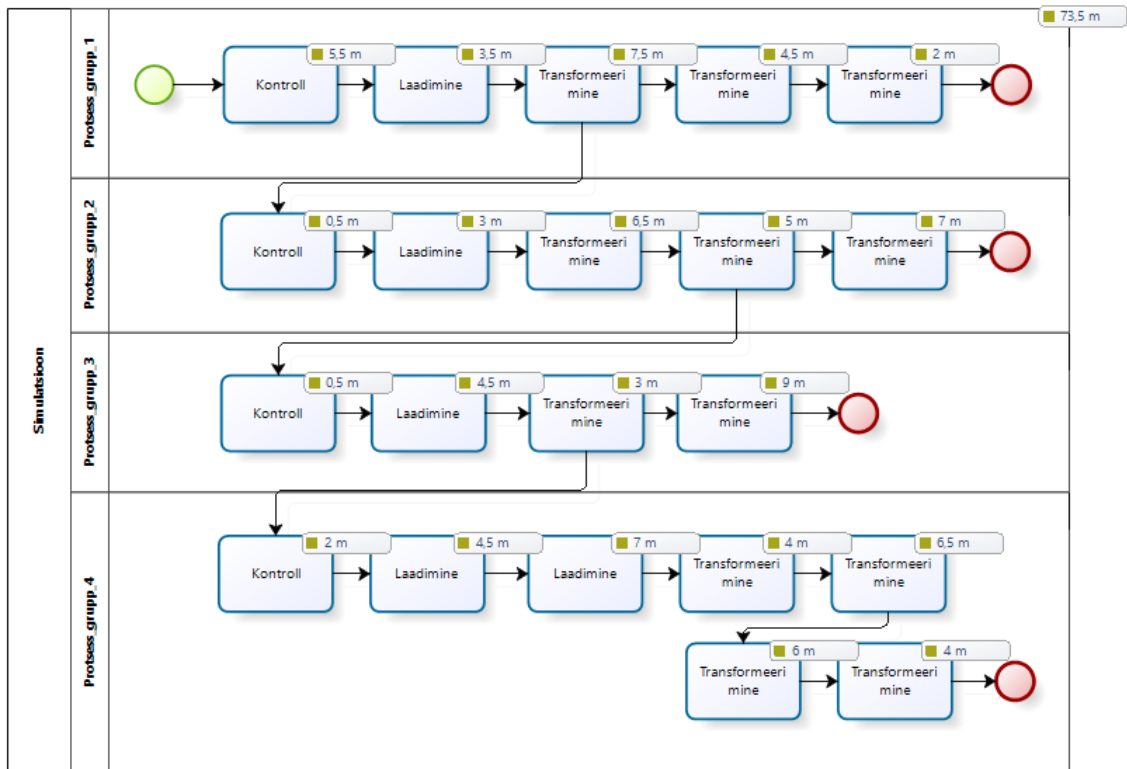
7.1.10 Hinna ja SLA tingimuste seadmine

Viimane samm on hinna ja SLA tingimustes kokku leppimine. Tarkvara valija eesmärk ei tohiks olla võimalikult madala hinna välja kauplemine, vaid mõlemaid osapooli rahuldava kokkuleppe saavutamine. Tihti kaasneb tarkvaraga tootja poolne tugi. Mõistlik kokkulepe kahe osapooli vahel aitavad kaasa edasistes sammudes kasutajatoe ja uuenduste pakkumise osas. Hinna ning SLA tingimuste osa on antud diplomitöös konfidentsiaalne, kuid kokkuleppe saavutamisel lähtuti mõistlikkusest ja edasisest koostööst.

7.2 Simulatsioon protsessi ahela näitel

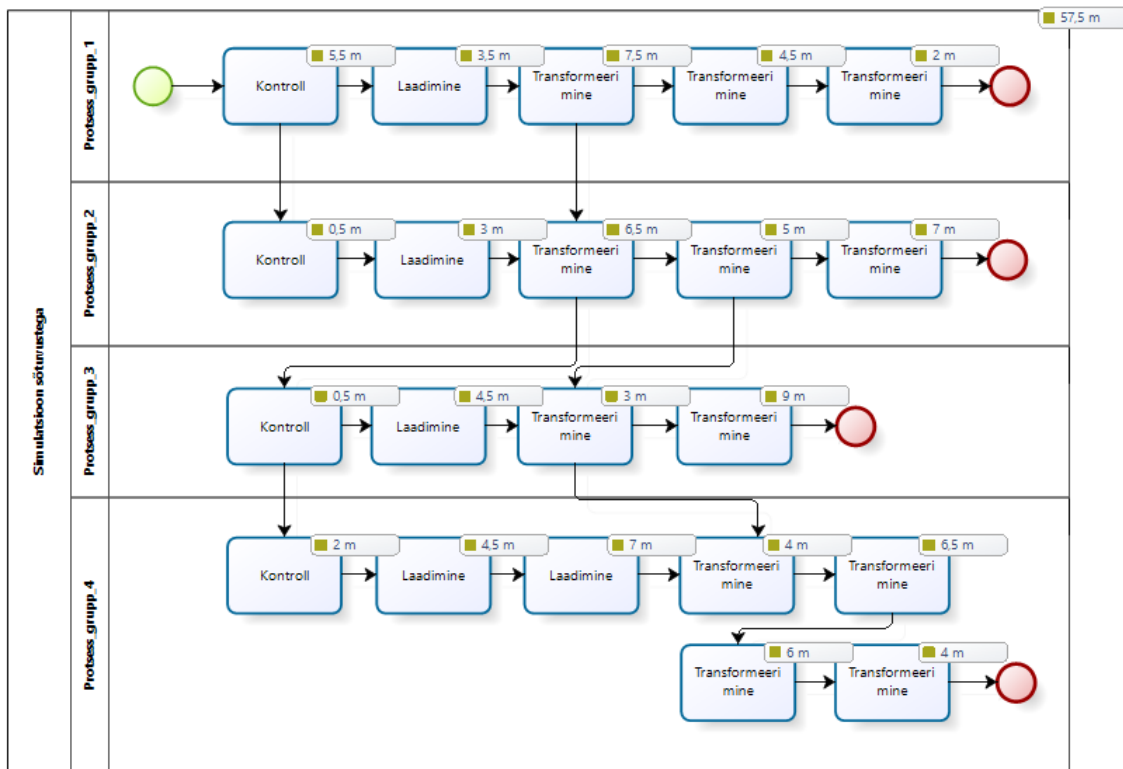
Üks võimalus hinnata ajalist võitu ühe laadimiste ahela osas on simuleerida protsesside tööd. Bizagi Process Modeleri kasutades on diplomitöö autor loonud mudelid, mis simuleerivad nelja protsesside grupi tööd ja mõõdavad kuluvat aega. Mudelite loomisel on protsessidele määratud töötamise ajad ideaalolukorras. See tähendab, et andmebaasi süsteem ei ole koormatud ning protsessid ei ole väga mahukad. Sellised mudelid ei simuleeri saja protsendiliselt toodangu süsteemi tööd, kuid annavad võimaluse hinnata ligikaudselt protsesside ahela ajalist kestvust.

Esimene simulatsioon (Joonis 10) kujutab vana laadimiste planeerimise tööriista võimalusi luua seosed protsessi gruppide vahel. Võimalik on käivitada uus protsesside grupp ainult algusest. Seda kas eelmise protsesside grupi lõpust või sobiva protsessi juurest. Joonis on kuvatõmmis Bizagi Process Modeleri simulatsiooni vaatest ning iga protsessi juures on töötamise aeg minutites. Paremal nurgas on kogu mudeli töötamise aeg summeeritult ehk aeg, mis kulub esimese protsessi käivitamisest viimase protsessi lõpetamiseks. Vaadeldava mudeli korral on lõpetamiseks kuluv aeg 73.5 minutit.



Joonis 10. Simulatsioon nelja protsesside grupi vahel. Iga protsesside grupp saab viidata järgmise grupi esimesele protsessile.

Teine simulatsioon (Joonis 11) kujutab uue laadimiste planeerimise tööriista võimalusi luua seosed protsesside ja gruppide vahel. Võimalik on luua mitmesuguseid seoseid nii protsessi gruppide kui protsesside vahel. See omakorda tähendab, et samaaegselt on töös korraka rohkem protsesse. Ajaliselt lõpetab mudel 57.5 minutiga. Optimaalsemate sõltuvuste korral on eelmise mudeliga võrreldes ajaline võit umbes 22%. Selline ajaline kasu tuleb olukorras, kus terve ahel töötab tõrgeteta algusest lõpuni. Ajaline võit on suurem näiteks juhul, kui mudelitel olev Protsess_grupp_1 esimene transformeerimise protsess vea tõttu katkeks. Taaskäivitamisel esimese mudeli puhul käivitatakse Protsess_grupp_2 esimese protsess Kontroll. Teise mudeli korral oleks Protsess_grupp_2 esimesed protsessid Kontroll ja Laadimine juba lõpetanud ning ajaline võit oleks 3.5 minutit.



Joonis 11. Sõltuvustega simulatsioon nelja protsesside grupi vahel. Keerulisemad sõltuvused gruppide ja protsesside vahel kasutades uue tööriista pakutatavat funktsionaalsust.

Kogu uue andmelaadimiste planeerimise tööriista kasutusele võtmisest tulenevat kasu erinevate mõõdikute järgi saab hinnata alles siis, kui uus toodangu keskkond ja sõltuvused on ülesse seatud võimalikult optimaalselt. Diplomitöö lõpetamise hetkel ei ole sõltuvused täielikult paigas ja töö optimeerimine selles osas veel käib.

7.3 Võimalikud edasised tegevused

Käesolevas alampeatükis kirjeldatakse lühidalt mõned võimalikud tegevused kogu andmelaadimiste protsessi parendamiseks peale uue tööriista kasutusele võtmist. Valitud vahend IBM Workload Scheduler võimaldab planeerida, ajastada ja automatiseerida ühes tööriistas erinevad ettevõtte protsessid. Sellest tulenevalt võiksid autori hinnangul olla kolm edasist tegevust järgmised:

- 1) Analüüsida raporteerimise süsteemide andmete kasutamist ning ajastada raportite genereerimine kohe pärast andmete laadimist andmeaita.
- 2) Luua standard allikasüsteemide ja andmeaida laadimiste käivitamise efektiivsuse tõstmiseks. Selle tulemusena peaks suurem osa andmete laadimisi

olema sündmuse põhised. Ühtne standard ja juhend, kuidas sündmuse põhise ajastamist ülesse seada, oleks aluseks kõigi uute andmelaadimiste arendamisel ning olemasolevate ümber tegemisel.

- 3) Koostöös andmeaita kasutavate osapooltega analüüsida olemasolevaid andmete kasutusmustrid ning vähendada käsitsi tehtavat tööd läbi automatiseerimise. Samuti tutvustada uut tööriista osapooltele ning leida laialdasemat kasutust.

Välja toodud kolm tegevust aitaksid kaasa ettevõtte sisese töö efektiivsuse tõstmisel ja rakendada valitud tööriista mitmekülgselt. Samas peab andmelaadimiste planeerimise osa tööriistas jääma isoleerituks ning esmajärgus täitma seatud eesmärki laadida andmed efektiivsel viisil.

8 Kokkuvõte

Uue andmeaida laadimiste planeerimise tööriista valiku protsess finantsettevõttes koosnes mitmest etapist. Vajadus olemasolev tööriist välja vahetada tulenes kolmest olulisest aspektist – olemasolev tööriist töötab vananenud platvormil, andmete andmeaita laadimine ei ole piisavalt efektiivne ning kulu ja risk vana süsteemi üleval hoidmiseks ületavad mõistlikkuse piiri.

Uued funktsionaalsed ja mittefunktsionaalsed nõuded kirjeldati koostöös ettevõtte erinevate tiimide ja kasutajatega ning seatud nõuded olid aluseks uue tööriista valimisel. Kokku pandi kirja 15 funktsionaalset nõuet, mida täpsustasid 43 alampunkti. Mittefunktsionaalseid nõuded kirjeldati kokku 8. Nõuete kirjeldamisel oli oluline sisend olemasoleva tööriista puudused, sest üks vajadus uue tööriista valimiseks oli tõsta laadimiste protsessi efektiivsust.

Erinevaid automatiseerimise ja planeerimise tööriistu on palju ning valiku kitsendamise aluseks võeti Gartneri 2012 maagiline ruut töövoogude automatiseerimise osas. Sealt valiti välja kolm valdkonna liidrit ning lisaks üks väiksem tööriist väljastpoolt maagilist ruutu. Kõigi nelja tööriista osas vaadeldi kirjeldatud nõuete täitmist kahes etapis. Esimeses etapis vaadeldi, kas tööriistad täidavad viite kriitilist nõuet. Etapi eesmärk oli välja filtreerida tööriistad, mis kriitilist funktsionaalsust ei täida. Kuna kõik neli tööriista täitsid kriitilised nõuded, vaadeldi teises etapis kõiki ülejäänud funktsionaalseid ja mittefunktsionaalseid nõudeid.

Valituks osutus IBM Workload Scheduler, mis täitis kõik seatud nõuded ja osutus hinna osas konkurentidest soodsamaks. Funktsionaalsuse osas oleks sobinud valikusse nii Stonebranch Universal Controller kui BMC Control-M, kuid mittefunktsionaalse nõue MFN_2 sai otsustavaks. Konfidentsiaalsuse tõttu ei ole hindasid võimalik antud diplomitöös välja tuua.

Analüüs ja migratsioon uue tööriista kasutusele võtmiseks koosnes kolme erineva meetodi analüüsist. Ühekordse tegevusena tehtava migratsiooni jaoks ei ole mõistlik luua programmi, mille arendus on aja- ja kulumahukas. IBM Workload Scheduler

pakub mitut erinevat meetodit vanast süsteemist andmete migreerimiseks uude süsteemi ning kõige mõistlikumaks pidas autor *Datamigrate* funktsionaalsust. Tööriista dokumentatsioonis on hästi kirjeldatud, kuidas luua sobiv fail IBM Workload Scheduleris keskkonna ülesse seadistamiseks. Migratsiooni plaanis lähtuti eemärgist olemasolev andmestik migreerida ilma suuremaid loogikamuudatusi tegemata, kuid tähelepanu pöörati Alan Raddingi artiklis kirjeldatud punktidele.

Viimases järelanalüüsi peatükis hinnati valiku tegemise protsessi, mille aluseks tarkvara valimise protsessi kirjeldavad kümme sammu. Koostati kaks simulatsioonimudelit, et võrrelda sõltuvuste halduse võimekust uues ja vanas tööriistas. Sõltuvuste lisamine protsesside ja gruppide vahel oli üheks oluliseks nõudeks uue tööriista valikul ning simulatsiooni mudel näitas ajalist võitu keerulisemate sõltuvuste tekitamise läbi. Täieliku hinnangu efektiivsuse tõstmise osas tänu uue tööriista kasutusele võtmisega saab anda siis, kui toodangu keskkond on optimaalsete sõltuvuste ning ressursside seadistamisega ülesse seatud. Järelanalüüs peatükis anti lisaks soovitus edasisteks tegevusteks, tõstmaks ettevõtte siseste protsesside efektiivsust.

Autori hinnangul õnnestus uue tööriista valik ning täideti seatud eesmärk. Kuigi esialgu tundus, et andmeida laadimiste planeerimise seab teatud spetsiifilised nõuded valitavale tööriistale, selgus töö käigus, et suuremad protsesside automatiseerimise tööriistad pakuvad sobivat ja kohaldatavat funktsionaalsust.

Kasutatud kirjandus

- [1] Gartner IT Glossary. [WWW] <http://www.gartner.com/it-glossary/data-warehouse/> (02.02.2017)
- [2] The Data Warehouse Lifecycle Toolkit. Kimball R., Reeves L., Ross M., Thornthwaite W. : John Wiley & Sons, Inc. 1998
- [3] Oracle Database Data Warehousing Guide. [WWW] <https://docs.oracle.com/database/121/DWHSG/concept.htm#DWHSG-GUID-452FBA23-6976-4590-AA41-1369647AD14D> (07.02.2017)
- [4] The Data Warehouse ETL Toolkit. Kimball R., Caserta J. : Wiley Publishing Inc. 2004
- [5] Case Method Fast-Track: A RAD Approach. Clegg, D., Barker, R. : Addison Wesley Longman, 1994.
- [6] Georgakopoulos, D., Hornick, M., Sheth, A. Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, 1995. [WWW] <http://www.workflowpatterns.com/documentation/documents/workflow95.pdf> (11.02.2017)
- [7] Functional and Non-Functional Requirements. [WWW] https://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm (04.03.2017)
- [8] Govekar, M., Mahapatra, B. Magic Quadrant for Workload Automation 2012. [WWW] <https://www.gartner.com/doc/1934915/magic-quadrant-workload-automation> (06.03.2017)
- [9] Control-M Datasheet. [WWW] <http://documents.bmc.com/products/documents/82/71/468271/468271.pdf> (09.03.2017)
- [10] Control-M Architecture. [WWW] http://documents.bmc.com/supportu/ctrlm9/help/Main_help/en-US/index.htm#4533.htm (09.03.2017)
- [11] Workload Automation Product Summary. [WWW] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Workload%20Scheduler/page/Product%20Summary> (11.03.2017)
- [12] Architectural view of Workload Automation. [WWW] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Workload%20Scheduler/page/Product%20Architecture> (11.03.2017)
- [13] Visualcron Features. [WWW] <http://www.visualcron.com/features.aspx> (14.03.2017)
- [14] Visualcron Order. [WWW] <http://www.visualcron.com/order.aspx> (14.03.2017)
- [15] Universal Controller Data Sheet. [WWW] <http://www.stonebranch.com/common/pdf/technicalbriefs/universal-controller-solution-brief.pdf> (15.03.2017)
- [16] Universal Controller Architecture. [WWW] <https://image.slidesharecdn.com/agentsitsiloswebinar-151105203249-lva1-app6892/95/silos-are-for-farmers-not-it-24-638.jpg?cb=1446755986> (15.03.2017)

- [17] Radding, A. Avoid data migration project failure: Five best practices. [WWW] <http://searchstorage.techtarget.com/tip/Avoid-data-migration-project-failure-Five-best-practices> (17.03.2017)
- [18] Pendergraft, L. Blakely, A. [WWW] <http://content.arma.org/IMM/IMMJanuary-February2010/IMM0110businessmatterstensteps.aspx> (19.03.2017)
- [19] Simulation in Bizagi. [WWW] http://help.bizagi.com/process-modeler/en/index.html?simulation_in_bizagi.htm (29.03.2017)

Lisa 1 – Extract faili struktuur ja sisu

```
SCHEDULE DEV#PROTSESS_GRUPP_1
ON RUNCYCLE DAILY "FREQ=DAILY;INTERVAL=1;BYWORKDAY"
( AT 0200 )
AT 0200
MATCHING PREVIOUS
PRIORITY HI
:
DEV#PROTSESS_KONTROLL_1_1
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_kontroll_1_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP

DEV#PROTSESS_KONTROLL_1_2
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_kontroll_1_2</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSESS_TRANSFORMEERIMINE_1_1

DEV#PROTSESS_LAADIMINE_1_1
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_laadimine_1_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
```

RECOVERY STOP
FOLLOWS PROTSSESS_KONTROLL_1_1

DEV#PROTSSESS_LAADIMINE_1_2

TASK

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_laadimine_1_2</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSSESS_KONTROLL_1_2
```

DEV#PROTSSESS_TRANSFORMEERIMINE_1_1

TASK

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat
Protsess_transformeerimine_1_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSSESS_LAADIMINE_1_1
END
```

SCHEDULE DEV#PROTSSESS_GRUPP_2

ON RUNCYCLE DAILY "FREQ=DAILY;INTERVAL=1;BYWORKDAY"

(SCHEDTIME 0200)

MATCHING PREVIOUS

FOLLOWS DEV#PROTSSESS_GRUPP_1.PROTSSESS_TRANSFORMEERIMINE_1_1

PRIORITY HI

:

DEV#PROTSSESS_KONTROLL_2_1

TASK

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_kontroll_2_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
```



```

DEV#PROTSESS_LAADIMINE_2_1
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_laadimine_2_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSESS_KONTROLL_2_1

```

```

DEV#PROTSESS_TRANSFORMEERIMINE_2_1
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat
Protsess_transformeerimine_2_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSESS_LAADIMINE_2_1
END

```

```

SCHEDULE DEV#PROTSESS_GRUPP_3
ON RUNCYCLE DAILY "FREQ=DAILY;INTERVAL=1;BYWORKDAY"
( SCHEDTIME 0200 )
MATCHING PREVIOUS
FOLLOWS DEV#PROTSESS_GRUPP_2.@
FOLLOWS DEV#PROTSESS_GRUPP_1.PROTSESS_TRANSFORMEERIMINE_1_1
PRIORITY HI
:

```

```

DEV#PROTSESS_LAADIMINE_3_1
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat Protsess_laadimine_3_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP

```

```

DEV#PROTSESS_TRANSFORMEERIMINE_3_1
TASK

```

```
<?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="/home/dev/execute">
      <jsdle:script suffix="">execute.bat
Protsess_transformeerimine_3_1</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
RCCONDSUCC "RC=0"
RECOVERY STOP
FOLLOWS PROTSCESS_LAADIMINE_3_1
END
```