TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut
Tarkvaratehnika õppetool

Testimise parendamise meetodid andmeaidas

Bakalaureusetöö

Üliõpilane: Helen Vaupere
Üliõpilaskood: 123592IABB
Juhendajad: Jekaterina Ivask
Silver Saar

Tallinn
2015

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.

(*kuupäev*) (*allkiri*)

Annotatsioon

Käesoleva töö eesmärgiks on pakkuda välja meetodid andmeaida testijate tiimidele millega võimaldatakse efektiivsem töökorraldus ning tulemus. Analüüsitakse maailma parimaid praktikaid testimisel, et pakkuda välja lahendusi enamlevinud kitsaskohtadele milleks on liigne manuaalne töö, vähene regressioonitestimine ning testijate ressursi leidmine. Lisaks kirjeldab autor vahendeid mis on kasutusele võetud finantsasutuse andmeidas ning esitatakse nende mõju testimisele, tuginedes kogutud statistikale.

Tulemustena esitatakse ettepanekud mida võiksid rakendada aidameida testijate tiimid, et muuta oma tööd efektiivsemaks. Samuti pakutakse välja juhtnöörid kuidas aidata uutel testijatel kiiremini oma töösse sisse elada ning kuidas saavad kogenumad testijad selleks oma panuse anda.

Lõputöö on kirjutatud eesti keeles ning sisaldab 54 lehekülge teksti, 4 peatükki, 8 joonist, 4 tabelit.

Abstract

The objective of this bachelor theses is to devise more efficient methods for warehouse database testers to achieve better work arrangement and outcome. World's best testing practices are examined in order to draw attention to shortcomings and work towards solutions. The weak points are the extensive manual work, insufficient regression testing and limited human resources. In addition, the author describes techniques which are used in the warehouse database of the financial institution and their impact is analysed based on statistics.

As the result of the study several suggestions are offered in order to enhance efficiency and facilitate the work of data warehouse testers teams. As well, practical guidelines are suggested which help new testers familiarize with their work and give some advice to experienced testers so that they can also contribute to the adjusting process.

The thesis is in Estonian and contains 54 pages of text, 4 chapters, 8 figures, 4 tables.

Jooniste nimekiri

Tabelite nimekiri

Lühendite ja mõistete sõnastik

Mapping document	Vastavusdokument sisaldab endas allika ning sihtkoha ärireeglite informatsiooni. Dokument leiab suurimat rakendust ETL protsessi arendajalt kes disainib ning arendab ETL protsesse. The mapping document contains the source,target and business rules information's, this document will be the most important document for the ETL developer to design and develop the ETL jobs.
QA (quality assurance)	Viis vigade või defektide vältimiseks valmistatud toodetes ning probleemide vältimine lahenduse või teenuse pakkumisel klientidele. A way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers.
Batch file	Pakkfail on tekstifail, mis sisaldab arvuti operatsioonisüsteemile saadetavat käsujada. A batch file is a text file that contains a sequence of commands for a computer operating system.
CRQ (change request)	Formaalne ettepanek toote või süsteemi muudatusele. Formal proposal for an alteration to some product or system.
BI (business intelligence)	Äriteave. Tehnikate ning vahendite kogum millega muuta toorandmed tähendusrikkaks ning arusaadavaks informatsiooniks ärianalüüsi otstarbeks Business intelligence. The set of techniques and tools for the transformation of raw data into meaningful and useful information for business analysis purposes

Sisukord

1 Sissejuhatus

Selle töö eesmärk on analüüsida ning pakkuda välja sobivad ning kohased meetodid parandamaks andmeaida testimise protsessi ning muuta andmeaida testijate töökorraldust efektiivsemaks.

1.1 Taust ja probleem

Enamus ettevõtetel eksisteerib andmebaas. Need, kellel tarvis rohkem andmeid hoiustada, omavad andmeaita. Kuna pelgalt kokku korjatud toorandmed ei too ettevõttele tulu, on andmeid vaja andmeaidas sobival kujul korrapäraselt hoiustada. Andmeaida testimisel kerkivad esile probleemid, kuidas saaks testimist võimalikult efektiivselt läbi viia ka suurte andmehulkadega, sealjuures kulutada nii raha kui aega mõistlikkuse piires. Võtmesõnaks on testide automatiseerimine. Andmemahud on liialt suured, et rakendada manuaalset testimist. Kahjuks peab tõdema, et isegi tänapäeval rakendatakse manuaalset testimist liiga palju.

Kvalifitseeritud andmeaida testijad on hinnas, kuna korralik testimine nõuab testija poolt eelnevaid kogemusi antud valdkonnas, et toime tulla ka keerukamates olukordades. Kahjuks on selliste testijate leidmisega probleeme, kuna testija ametit peetakse enamasti lühikest aega ning suundutakse seejärel üle teistele ametipositsioonidele.

1.2 Ülesande püstitus

Töö eesmärk on pakkuda lahendused üldlevinud küsimustele seoses andmeaida ning ETL testimisega. Käsitletakse kolme olulist punkti – testide automatiseerimine, regressioonitestimine ning andmeaida testijate kompetents. Autor püüab leida vastused küsimustele, kas automatiseerida, millal automatiseerida ning mida automatiseerida. Kui põhjalikult teha regressioonitestimist ning millised peavad olema testijate oskused edukaks andmeaida testimiseks.

1.3 Metoodika

Antud töö raames on uuritud maailma parimaid praktikaid andmeaida testimisel kasutades selleks avaldatud teadusartikleid ning raamatuid. Eesmärk oli leida parimad lahendused

millega muuta efektiivsemaks andmeaida testimist. Kuna käesoleva töö autor töötab ühes Eesti suurimatest finantsasutustest andmeaida testijana, tuuakse autori töökohast näiteid vahenditest mis on arendatud eesmärgiga muuta testimist efektiivsemaks. Toetudes kogutud statistikale, hinnatakse selliste vahendite tegelikku mõju ning esitatakse asjakohased ettepanekud edasiseks töökorralduseks. Lisaks külastas autor finantsasutuse laenude osakonda, kus intervjueeriti andmebaaside testijuhti, et uurida millised meetodid on neil võetud kasutusele eesmärgiga parendada oma töökorraldust ning milliste kitsaskohadega seistakse silmitsi. Saadud informatsiooni põhjal esitab autor ettepanekud parimatest meetoditest mida soovitab rakendada erinevatel andmeaida testijate tiimidel.

1.4 Ülevaade tööst

Esimeses osas antakse lugejale ülevaade baasmõistetest. Käsitletakse mõisteid nagu andmeait, äriteave, testimine ning andmeaida testimine. Ühtlasi selgitatakse, miks pole võimalik võtta neid väljendeid üheselt. Teine osa toob välja probleemid ning kitsaskohad, mis puudutavad andmeaida testimist, tuginedes töö autori enda kogemustele andmeaida testimisel.

Kolmas osa analüüsib, millised meetmed aitavad lahendada teises peatükis välja toodud kitsaskohti. Konkreetse näitena kasutatakse töös finantsasutuse aidmeaita, kus testimise efektiivsuse tõstmiseks on kasutusele võetud mitmeid erinevad vahendeid. Analüüsides statistikat, tehakse järeldused, kas kirjeldatud meetmed on täitnud oma eesmärki ning antakse soovitused, mida võiks edasi arendada.

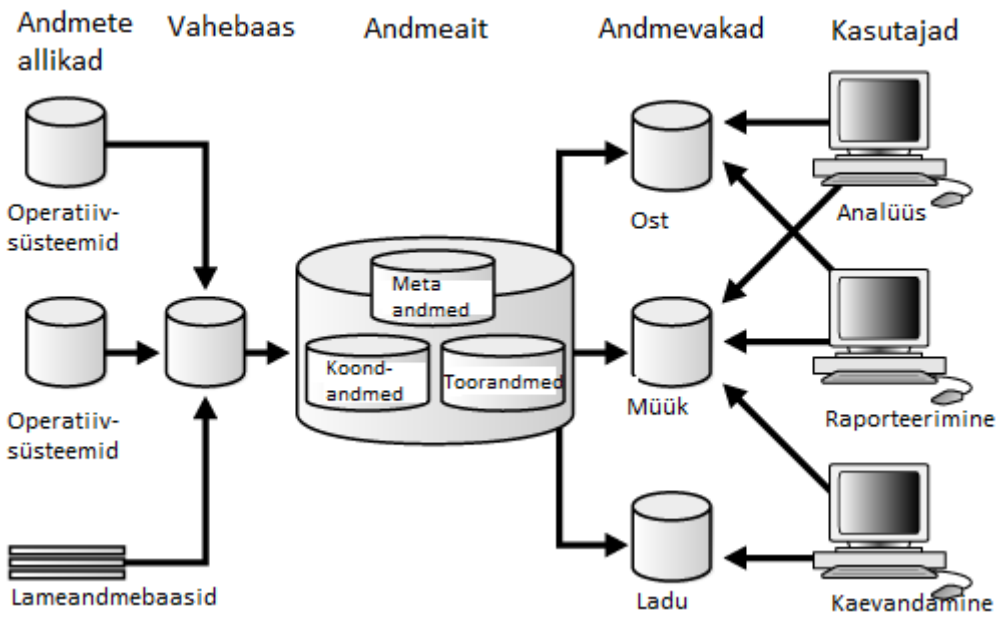
2 ÜLEVAADE ANDMEAIDAST

Finantsasutuse kui kliendipõhise organisatsiooni edukus sõltub suuresti sellest, kui hästi tunneb organisatsioon oma klienti. Selle põhjal suudab pank pakkuda just neid teenuseid ja väärtusi, mida kliendid tarbivad ja hindavad. Et kliente tundma õppida, kogub pank infot nende tegevusest - näiteks kontorivisiidid, laenutaotlused, internetipanga külastused ja muu selline informatsioon. Taoline info omab suurt väärtust, olles aluseks trendide avastamisel ning abiks juhtimisotsuste vastuvõtmisel.

Selline suur hulk infot talletatakse andmeaidas. Andmeait (ingl. k *data warehouse*) on defineeritud kui koopia transaktsiooni andmetest, mis on struktureeritud spetsiaalselt päringute ning analüüsi tarbeks (Kimball 1996, 310). Samas kui W.H. Inmon, kes on tuntud ka andmeaida vaimse isana, defineerib andmeaida järgmiselt: “Andmeait on subjekt-orienteeritud, integreeritud, püsiv, ajast sõltuv kollektsioon andmetest mis toetavad juhtimisotsuseid.” (Inmon 1995) Andmeait kogub andmeid erinevatest operatiivsetest süsteemidest ning neid andmeid ei kustutata ega muudeta. Kui varasemalt oli andmeait üksnes andmete säilitamiseks, siis tänapäeval kannab andmeait märksa rohkem rolle. Andmeait on äriteabe (ingl. k *business intelligence*) keskmeks. Andmed erinevatest allikatest läbivad ETL (Extract, Transformation, Load) protsessi enne kui nad andmeaita salvestatakse.

2.1 Andmeaida arhitektuur

Kuigi detailne andmeaida struktuur on ettevõtetel erinev, saame laias laastus andmeaida arhitektuuri esitada järgneva joonise põhjal.



Joonis 1. Andmeaida arhitektuur

Andmeaida keskkond

- Allikad, mis varustavad andmeaita või andmevakka andmetega. Joonisel on näidatud operatiivsüsteemid ja lameandmebaasid (ingl. k *flat-file database*)
- ETL protseduurid, mis kannavad andmeid ühest asukohast teise
- Erinevad tööriistad ning rakendused andmete kasutajatele
- Metaandmed

Andmed, mida soovitakse säilitada andmeaidas, laetakse erinevatest operatiivandmebaasidest (allikast). Näiteks finantsasutuses võivad need olla laenud, müük ja lepingud. Enne kui andmed lõplikult andmeaita salvestatakse, võivad, kuid ei pea. nad läbima vahebaasi, kus andmeid töödeltakse vastavalt vajadusele.

Andmevakad (ingl. k *data mart*) on osad andmeaidast, mis rahuldavad erinevate osakondade infovajadusi. Andmevakkade loomisele on erinevaid lähenemisviise. Järgnevalt käsitletakse kolme neist.

1. Esimesena luuakse andmevakk ning alles seejärel andmeait. Ettevõte valib välja osakonna või rühma, mis kõige enam võidab andmebaasi kasutamisest ja selle rühma andmete põhjal luuakse andmevakk (Vallaste 2015).

2. Eksisteerivale andmeaidale luuakse andmevakad, mis sisaldavad hoitava informatsiooni alamhulka.
3. Andmeait ning andmevakk on üksteisest sõltumatud ning andmevakkade andmed võetakse erinevatest pärandsüsteemidest.

Seisvad andmed on kasutud. Andmete eesmärk andmeaidas on toetada firmat informatsiooniga. Selleks võivad lõppkasutajad kasutada raporteid või online-analüüsi.

2.2 Äriteave

Äriteave on tehnoloogiapõhine protsess andmete analüüsimiseks ning esitamiseks eesmärgiga aidata ärijuhte ning teisi lõppkasutajaid juhtimisotsuste läbiviimisel (Rouse, 2009). Meid ümbritseb rohkem andmeid kui iial varem. Andmete hulk on viimastel aastatel drastiliselt suurenenud, kuid oskused sellest informatsiooni välja lugeda, on maha jäänud. Inimesed janunevad informatsiooni järele ning seda mitmel põhjusel. Suurem kogus informatsiooni annab turul eelise ning oma kliendi tundmine loob väärtuslikuma positsiooni.

Äriteave andmed tulenevad tavaliselt andmeaidast või väiksematest andmevakkadest, mis hoiavad alamosa firma informatsioonist. Äriteave ülesandeks on muuta andmeaidast tulenevad andmed informatsiooniks, mida kaustatakse paremate otsuste läbiviimiseks. Enne kui andmeid kasutatakse äriteave rakendustes, peavad allikabaasidest pärit toorandmed (ingl. k *raw data*) läbima transformatsiooni, mille käigus andmed korrastatakse ning veendutakse nende kvaliteedis.

2.3 ETL protsessid

ETL (Extract, Transformation, Load) süsteem on kõik see, mis jääb operatiivandmebaaside ja andmeaida esituse vahele. ETL protsess on andmeaida võtmekomponent, kuna ebakorrektsed või eksitavad andmed põhjustavad valesid äriotsuseid. ETL protsesside arendus võtab kuni 80% andmeaida projektide arendusest. (Liu *et al* 2011, 543-570)

ETL protsessi võib jagada kolmeks: eraldamine (ingl. k *extract*), transformatsioon (ingl. k *transformation*) ning sihtkohta laadimine (ingl. k *load*).

Eraldamine

Enamasti vajavad andmeaidad andmeid erinevatest allikatest (ingl. k *source*), milleks võivad olla relatsiooniline andmebaas, XML dokument ning isegi Exceli failid. Eraldamise osa ülesandeks jääb andmetest arusaamine ning selliste andmete kopeerimine, mida ETL protsess vajab edasiseks manipulatsiooniks.

Transformatsioon

Kui andmed on eraldatud allikast, on tarvis neid andmeid korrastada.

- parandada andmevead, lahendada domeenikonflikte, tegeleda puuduvate elementidega või paigutada andmed sobivasse formaati.
- kombineerida info, mis on tulnud mitmest erinevast allikast
- eemaldada duplikaadid.

Kõige sellega tegeleb vastav transformatsiooni osa. (Kimball 2004, 18)

Sihtkohta laadimine

Viimase ETL protsessi samm on andmed struktureerida ning füüsiliselt sihtkohta paigutada. Sihtkohaks võivad olla andmeait või andmevakk.

2.4 Testimine

Käesoleva töö autor vastaks küsimusele mis on testimine järgnevalt: tarkvara testimine on protsess mille käigus käivitatakse rakendus sooviga leida programmiviga (What is Software…, 2015). Sama definitsiooni võib laiendada mistahes testimisele, sealhulgas ka andmeaida testimisele.

Testimine on kui verifitseerimise ning valideerimise protsess, mille käigus tuvastatakse, et rakendus või produkt:

- vastab ärilistele ning tehnilistele nõudmistele
- töötab nii nagu temalt oodatakse

(What is Software…, 2015)

Testimine on tarkvaraarenduse elementaarne osa. Arendusosakonnad ei saa lasta välja tarkvara ilma, et kontrollitaks potentsiaalseid puudujääke. Testimise eesmärk on kontrollida, et tarkvara teostab funktsionaalsust, mida sellelt oodatakse. (Tanuška *et al* 2009, 7-11)

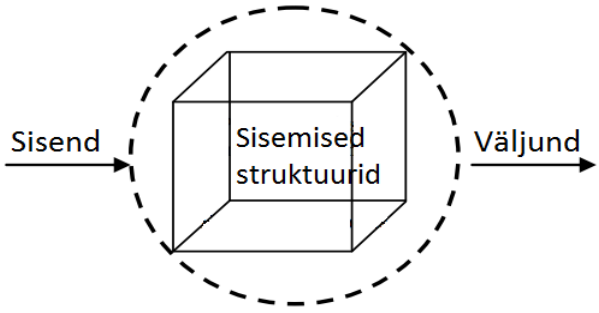
2.4.1 Testimise meetodid

Testimise meetodid jagatakse laias laastus kaheks: musta kasti meetod (ingl. k *black-box testing*) ning valge kasti meetod (ingl k. *white-box testing*).

Musta kasti meetodi puhul puudub testijal ligipääs sisemistele andmestruktuuridele. Nagu nimetus ütleb, on süsteem testija jaoks nagu must kast, kuhu sisse ta ei näe. Musta kasti testimine viiakse läbi nõnda, et antakse süsteemile sisendeid ning saadakse oodatud väljund. Fookuses on süsteemi funktsionaalsus.

Alternatiiviks musta kasti testimisele, on valge kasti meetod, kus testijal on ligipääs programmikoodile. Valge kasti testimine on äärmiselt efektiivne, kuna võimaldab leida programmiviga enne, kui see on jõudnud probleeme tekitada. Reeglina viib valge kasti testimist läbi arendaja, kuna tal on suurepärased teadmised, kuidas programmi komponendid omavahel suhtlevad. (Jovanovi 2008)

Igal testimismeetodil on omad plussid ning miinused. Mõningaid programmivigu ei leia ainult musta kasti testimisega ning sama kehtib ka valge kasti meetodi puhul. Halli kasti testimine (ingl. k *grey-box testing*) võimaldab kombineerida musta ning valge kasti meetodite tugevad küljed. Halli kasti testimine tõstab testide katvust (ingl k. *test coverage*), kuna lubab meil keskenduda süsteemi kõikedele kihtidele, kombineerides valge ning musta kasti meetodeid. Halli kasti testijal peab olema arusaam sisemistest andmestruktuuridest ning algoritmidest, et luua testjuhtumeid (ingl k *test case*). (Khan *et al* 2015)



Joonis 2. Halli kasti testimine andmeaida testimise puhul

Võib öelda, et halli kasti meetodi puhul toimub ka andmeaida testimine, kuna testijatel on ligipääs sisemistele andmestruktuuridele ning nad peavad omama laiemat arusaama süsteemist kui musta kasti testimisel.

Järgnevas peatükis antakse ülevaade andmeaida testimisest ning tuuakse välja selle erinevused võrreldes tarkvara testimisega. Samuti käsitletakse, milliseid erinevaid teste tuleks läbi viia andmeaidas ning milleks on need vajalikud.

2.4.2 Andmeaida testimine

Rääkides ükskõik millisest tarkvaraprojektist, on testimine oluline osa arendusest. Nii on see ka andmeaida arenduse puhul. Kujutage vaid ette, millised tagajärjed on defektsetel andmetel andmeaidas. Organisatsiooni juhtivad otsused sõltuvad suuresti andmetest, mida hoiustatakse andmeaidas. Sellised andmed peavad olema kõrgeima kvaliteediga. Keerulised ärireeglid ning transformatsiooni loogika, mis on ehitatud ETL protsessidesse, nõuavad põhjalikku testimist. ETL protsessi testimine on äärmiselt oluline andmeaida arenduses. See on üks komplekssemaid ning nõudlikumaid osasid andmeaida ehitamisel. Andmeaida testimine on võrdlemisi uus valdkond, seetõttu puudub antud tegevusel küpsus ning teadmiste jagamise ajalugu. (Deshpande 2014, 60-71)

Pole tähtis, kui süvitsi on süsteemi testitud, on peaaegu kindel, et varem või hiljem ilmneb ootamatu viga, mida ETL protsess ei saa korralikult käsitleda. Seda silmas pidades tuleb hoida meeles, et testimine peab ühel päeval lõppema, kuid andmete kvaliteedi verifitseerimine on lõpmatu protsess. Piir testimise ning andmete valideerimise vahel on selgelt sõltuv sellest, kui täpselt on nõuded fikseeritud projekti reguleerivas lepingus. (Khan *et al* 2012)

Andmeaida testimist ning andmebaasi testimist kiputakse samastama, kuid tegelikult erinevad need kaks testimist üksteisest mitmel moel. Järgnevalt tuuakse välja põhilised erinevused tarkvaratestimise ning andmeaida testimise vahel.

- Tarkvaratestimine on põhiliselt fokusseeritud programmikoodile, kuid andmeaida testimine on suunatud andmetele ja informatsioonile
- Erinevalt üldistest tarkvarasüsteemidest, hõlmab andmeaida testimine suuri andmehulki mis omavad märgatavat mõju tootlikkusele ja jõudlusele.
- Andmeaida testimisel on laiem skoop kui tarkvaratestimisel, kuna fookuses on kasutajatele toimetatavate andmete õigsus.

- Kuigi üldisel tarkvarasüsteemil on suur hulk erinevad stsenaariume, siis valiidsete stsenaariumite arv on piiratud. Teisest küljest on andmeaida eesmärk pakkuda mistahes andmevaateid. Seetõttu on kõikvõimalikud kombinatsioonid virtuaalselt limiteerimata ning ei saa olla täielikult testitud.
- Enamik üldisete tarkvarasüsteemide testimist tehakse ära enne süsteemi kasutuselevõttu. Kuid andmeaida testimise tegevused leiavad aset peale süsteemi redaksiooni.

(Golfarelli *et al* 2009)

Andmeaida testimise vajadus võib tuleneda erinevatest asjaoludest. Seetõttu võib andmeaida testimist vaadelda neljast vaatepunktist:

- Uue andmeaida testimine – uus andmeait on üles ehitatud. Andmete sisendid on vastavalt kliendi nõutele ning erinevad andmeallikad ning uus andmeait on loodud ETL vahendite abiga.
- Migratsioonitestimine – kliendil on olemasolev andmeait ning ETL protsessid ning soovitakse rakendada uut meedet või töövahendit, eesmärgiga parandada efektiivsust.
- Muudatuste testimine (ingl. k *change request*) – sellise projekti puhul laetakse uued andmed olemasolevasse andmeaita. Samuti võib muutus tuleneda ümber kujunenud ärireeglitest.
- Raporti testimine – raportid on andmeaida lõpptulem ning eesmärk, milleks andmeait on algselt loodud. Raporteid tuleb testida, valideerides asetust, andmeid ning arvutusi.

(Chhaperia 2015)

Dr. S.L. Gupta, Dr. Payal Pahwa ning Ms. Sonali Mathur esitavad järgmised testimise tüübid, mis on rakendatavad ka aindmeaidale.

- Funktsionaalne testimine – veendutakse, et andmed on laetud andmeaita ärinõuete kohaselt
- Kasutatavuse testimine – kasutaja suhtleb süsteemiga, et veenduda kasutajamugavuses ning arusaadavuses.
- Jõudluse testimine – testitakse süsteemi efektiivsust keskonnas, kus on kindel töökoormus. Jõudlustestide eesmärk on leida kitsaskohad ETL protsessi disainis, näiteks topeltlugemine dokumendist või ebavajalike vahetabelite loomine (Theobald 2007).

- Stressitest – hinnatakse süsteemi jõudlust suure töökoormuse all.
- Turvalisusetest – veendutakse, et süsteemis paiknevad andmed on kaitstud väliste osapoolte eest.
- Taastuvustest – hinnatakse, kui kiiresti taastub süsteem pärast kokkujooksmist.
- Regressioonitest – veendutakse terve süsteemi toimimises peale muudatuse jõustumist.

(Gupta *et al* 2012)

Testimine pole ühe persooni tegevus. Meeskond tuleks kokku panna juba projekti planeerimise faasis. (Khan *et al* 2012). Seetõttu on hea tava kaasata andmeaida testimisse järgnevad ressursid:

- Ärianalüütikud - loovad ärinõuded.
- Andmeaida testijad – arendavad testplaani ning käivitavad testjuhtumid. Täpse testiplaani arendamine projekti varajastes faasides on üks edu võtmetest. Kui viga on leitud hilisemas testimise etapis, võib see projektile tähendada kõrget finantskahjut. Seetõttu on testijate varane osalus nõuete ning disaini tegevustes tugevalt propageeritud. (Deshpande *et al* 2011, 543-570)
- Arhitektid – seavad üles testkeskkonna
- Arendajad – teostavad ühikteste
- Andmebaasi administraatorid – viivad läbi stressi- ja jõudlusteste
- Ärikasutajad – viivad läbi funktsionaalseid lõppkasutaja teste. Põhiline eesmärk andmeaida rakendustel on teha andmed kättesaadavaks ärikasutajatele. Kasutajad tunnevad andmeid kõige paremini ning nende osalemine testimisprotsessis on edukas andmeaida arenduses võtmekomponendiks. (Theobald, 2007).

Kokkuvõtvalt võib rolle ning nende testimisega seotuid isikuid kujutada järgnevas tabelis.

	Arendaja	**Andmeaida testija**	**Andmebaasi administraator**	**Ärikasutaja**
Ühiktestid	X			
Funktsionaalsus		X		
Kasutatavus				X
Jõudlus			X	
Stress			X	
Taastuvus			X	
Regressioon		X		
Turvalisus			X	

Tabel 1. Testimisega seotud rollid ning ülesanded

Käesolevas töös on vaatluse all andmeaida testijad, kelle ülesandeks on veenduda andmeaida funktsionaalsuses ning kontrollida, et muudatuste järel säiliks ka olemasolev funktsionaalsus.

Wayne Yaddow toob välja mitmed põhjuseid, miks on oluline andmeaida testimine:

- Allika andmed on tihti suuremahulised ning pärinevad erinevat tüüpi andmehoidlatest.
- Allika andmete kvaliteedis ei saa olla kindel, enamasti vajavad need andmed puhastamist ning profileerimist.
- Arvukad allikaandmete kirjed võidakse tagasi lükata.
- Allika välja väärtus võib olla kadunud kohas, kus see peaks alati eksisteerima.
- ETL protsess peab läbima mitmed faasid enne kui andmed laetakse andmeaita. Et tagada andmete korrektsus, peavad ETL komponendid olema üksikasjalikult testitud

(Yaddow *et al* 2012, 75)

Eraldi soovib autor antud töös pöörata tähelepanu andmeaida regressioonitestimisele, kuna regressioonivead võivad olla esmapilgul silmatorkamatud, kuid nende vigade maksumus seevastu suur. Järgnev peatükk tutvustab lähemalt regressioonitestimist andmeaidas.

2.4.3 Regressioonitestimine

Regressioonitestimise eesmärk on veenduda, et peale soovitud muudatuste ei tekiks tahtmatuid muudatusi süsteemis. Ärinõuded muutuvad pidevalt, mis omakorda on sisendiks muudatuste arendamisele. Regressioonitestimisel on eriti suur roll just andmeaidal kui pidevalt täieneval süsteemil. Regressioonitestimine automatiseerimine on vajalik, et liigselt aega kulutamata veenduda andmete korrektsuses peale muudatuste jõustumist.

Kaasaegsed arendusmeetodid nagu Extreme Programming (XP) toetavad regressiooniteste. See tähendab, et peale iga muudatust käivitatakse korduvad testid ning tegelikke tulemusi võrreldakse oodatud tulemustega. Väga sobilik ühik-testimise töövahend regressiooni testide läbiviimiseks on JUnit. Üldiselt viib ühiktestimist läbi arendaja ning määrab (enamasti) klassitasemel väited. Kui väited ei anna positiivset tulemust, on test ebaõnnestunud. Sellised raamistikud nagu JUnit ei kohandu ETL protsessidele, testimaks regressiooni. Üldjuhul on JUnit ning ning muud sarnased töövahendid head kasutamiseks väikeste, selgelt määratletud osade või funktsioonide puhul koodis. Vastupidiselt sellele on ETL protsesside puhul oluline testida kogu protsessi sündmuste käiku, kaasa arvatud kõrvalefekte. (Tjoa *et al* 2006).

Regressiooni võib põhjustada:

- Väär veaparandus
- Puudulik veaparandus
- Mittearvestamine erinevate versioonidega

Isegi kui ollakse mõistnud regressioonitestide vajalikkust, on on taoliste testide läbiviimine manuaalselt enneolematult suur aja- ning ressursikulu. On igati loogiline kasutada testide automatiseerimist ning seda mitte ainult regressioonitestide juures. Alljärgnev peatükk tutvustab lähemalt automatiseerimist andmeaidas. Tuuakse välja automatiseerimise eesmärgid ning kasutegurid.

2.4.4 Testimise automatiseerimine

Testide automatiseerimine on protsess, kus testid käivitatakse selleks ettenähtud takvara poolt. Lisaks testide käivitamisele, võib toimuda ka automaatne tulemuste võrdlus ning raportite genereerimine.

Andmeaida arendusele on iseloomulikud pidevad redaktsioonid (ingl k. *release*). Peale igat iteratsiooni lõppu oodatakse, et andmeait oleks valmis järgmisteks ETL arendusteks ning andmed tabelites oleksid piisavalt kvaliteetsed. Selline lähenemine nõuab automatiseerimise kasutuselevõttu.

Üksnes manuaalne testimine pole praktiline. Esile kerkivad kaks põhilist probleemi:

- Manuaalne testimine on ajakulukas ning aeglustab töötava tarkvara väljalaskeid.

- Manuaalne testimine pole otstarbekas regressioonitestide puhul. Lisaks uute väljalasete testimisele on sama oluline veenduda, et olemasolev funktsionaalsus säiliks.

Testimise automatiseerimine nõuab alguses lisapingutust ning võtab sissetöötamiseks parajalt aega, kuid pikemas perspektiivis suurendab efektiivsust. Automatiseerimine tõstatab aga hulga keerulisi küsimusi. Enamus hetkel saadaval olevatest automatiseerimise töövahenditest on disainitud tarkvaraarenduse jaoks ning pole hõlpsalt kohandatavad andmebaaside arenduseks. Lisaks on andmeaitade ning andmebaaside puhul tegemist suurte andmehulkadega, mis teevad olukorra veelgi keerulisemaks.

Testimise teeb komplitseerituks ka andmeaitade keerukas arhitektuur, mis hõlmad mitmeid andmebaase – vahebaasid, esitlusbaasid, ETL protsessid andmete eraldamiseks allikast, transformeerimiseks ning laadimiseks andmeaita. Samuti raporteerimismootorid ning rakendused. Testimise tööriistad on üldjuhul kallid. Levinud komme on kombineerida automatiseerimist manuaalne testimisega.

Wayne Yaddow toob oma raamatus välja testide automatiseerimise eesmärgid:

- automatiseerida andmeaida testimise tsüklist nii palju kui võimalik
- verifitseerida andmeid ETL protsessi igas punktis
- verifitseerida kogu andmeid, mitte ainult alamhulka
- verifitseerida keerukaid transformatsioonireegleid
- vähendada manuaalse testimise osakaalu, mis võib kujutada endast tuhandeid SQL päringuid
- identifitseerida sobimatud ning puuduvad andmed
- arendada regressioonitestide kogumik süsteemitestimiseks
- valideerida andmed allika ning lõpliku sihtkoha vahel

(Yaddow *et al* 2012, 176)

Käesoleva töö neljandas osas analüüsitakse, milline on efektiivne viis viia läbi andmeaida testimist läbi testide automatiseerimise. Näitena tutvustatakse turul pakutavat Querysurge töövahendit, mis on loodud spetsiaalselt andmeaida testide automatiseerimiseks. Lisaks toob autor näited töövahenditest mida ta kasutab oma igapäevatöös finantsasutuse andmeaida testimisel ning analüüsib läbi statistika, millist mõju need töövahendid on avaldanud testimise

efektiivsemaks muutmisele ning kas selline praktika on midagi, mida võiks pakkuda soovitusena välja teistele testitiimidele, kes sarnaste andmeaitadega tegelevad.

3 KITSASKOHAD ANDMEAIDA TESTIMISEL

Andmeaida testimine on mingil määral jäänud maha tarkvaratestimisest. Kui tarkvara testimise töövahendeid on nii palju, et neid on pea võimatu kokkugi lugeda, siis andmeaida testimise vahendeid on vaid käputäis. Põhiline takistus, miks testimisvahendeid napib, on

andmeaitade erinev sisemine struktuur. Selles valdkonnas puudub kindel standard. Suuremate finantsvõimalustega ettevõtted saavad ehitada vastavalt oma andmeaidale ka automatiseerimise platvormi, et hoida kokku kulusid ning aega. Kuid suur enamus andmeaida testimist sisaldab liigset manuaalset tööd, mis lisaks ajakulule võib tähendada ka inimeste hooletusvigu.

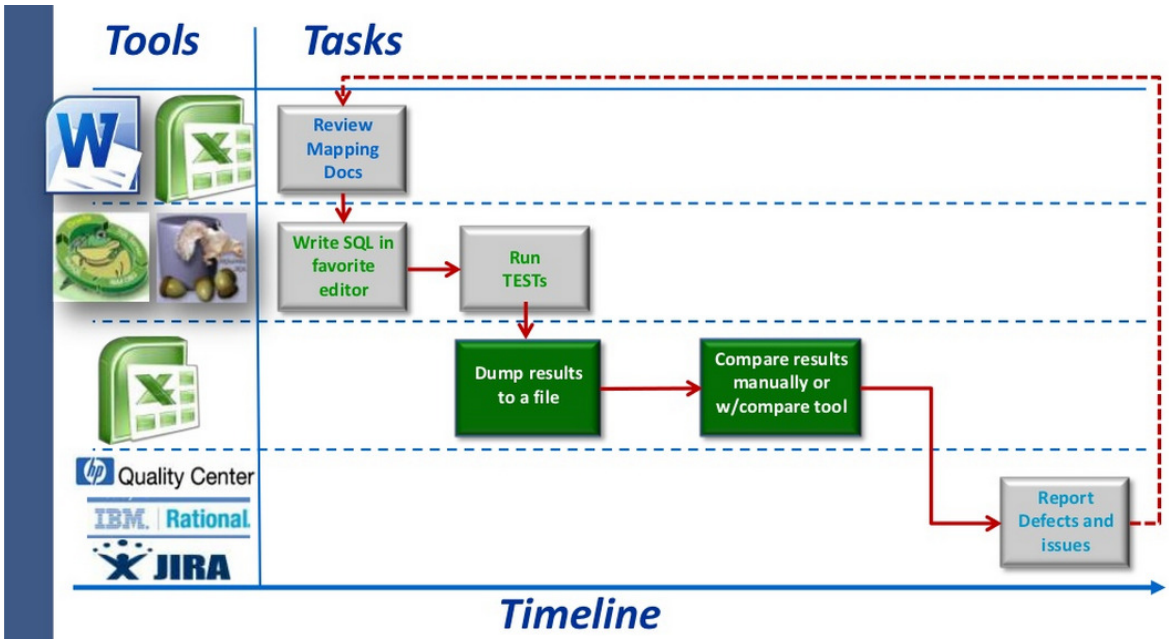
Eraldi murekohana tuuakse käesolevas töös välja testijate kompetents. Kvalifitseeritud andmeaida testijatest on tööturul puudus, seetõttu võetakse tööle väheste kogemustega tööjõudu. Lähtudes palgad.ee andmetest on testijate töötasu väiksem arendajate ning analüütikute töötasust. Sellest võime järeldada, et paariaastase kogemusega testijad soovivad pürgida ametikohale, kus nende palgatase on töökoormusele vastavam.

Järgnevas peatükis kirjeldatakse manuaalset andmeaida testimise protsessi, kus töö autor näeb etappe, mida saaks ja tuleks parendada eesmärgiga muuta testimine efektiivsemaks.

3.1 Manuaalne testimine

Enamik andmeaida testimist toimub kas manuaalselt või kombinatsioonis mõõduka automatiseerimisega. Manuaalne testimine tähendab, et testpäringud kirjutatakse käsitsi ning tulemused kopeeritakse käsitsi keskkonda, kus toimub võrdlus. Üldjuhul toimub ka võrdlus ise manuaalselt, see tähendab, et testija võrdleb ise, kas andmed ühtivad seal, kus tarvis.

Protsessi iseloomustab järgnev joonis.



Joonis 3. Manuaalse testimise protsess (What is a Data Warehouse…, 2012)

Testijale on sisendiks vastavusdokument. Läbi vastavusdokumendi omandab testija arusaama loogikast, mille järgi andmed läbi süsteemi rändavad.

Vastavusdokument kirjeldab andmete liikumist allikast sihtkohta ning sisaldab endas järgmist:

- allika sisendi kirjeldus
- väljundi kirjeldus
- äri-ja transformatsioonireeglid

Mõningad mapping-dokumendi eesmärgid:

- kehtestada ühtne ja järjepidev andmete liikumise analüüs; disaini ning kodeerimismuster
- kehtestada parim praktika ning järjepidevus kodeerimise ning nime standardites
- vähendada arendus- ning ülalpidamiskulusid
- täiendada kontrolli andmete liikumise üle, et kindlustada andmete kvaliteet ning terviklikkus

(Yaddow *et al* 2012, 79)

Järgmise sammuna kirjutatakse manuaalselt meelepärases redaktoris SQL päringud ning käivitatakse need. Tulemused salvestatakse Exceli tabelisse, kus võrreldakse neid käsitsi.

Manuaalse testimise puhul on kõige aeganõudvamad sammud SQL päringute kirjutamine ning andmete võrdlemine. Lisaks ajakulule on manuaalsel testimisel suur risk teha SQL päringutesse vigu. Just sellised tegevused peavad olema suunatud automatiseerimisele, et vähendada inimfaktorist tulenevaid vigu.

3.2 Vähene regressioonitestimine

Nagu ka eespool on välja toodud, siis regressioonitestide eesmärk on veenduda, et kogu toimiv funktsionaalsus jääks toimima ka peale uute muudatuse installeerimist. Seetõttu on äärmiselt oluline rõhutada regressioonitestide olulisust. Regressioonitestimine on fundamentaalne osa kvaliteediprotsessist. Regressioonivead võivad olla peidus ilma, et midagi silmnähtavalt valesti oleks.

Üldlevinud kaks põhilist viga, mis on allikaks regressiooni tekkimisele:

1) Arendaja usaldamine
2) Ajaraam, mis sunnib testijaid välja jätma osad, millel pole otsest silmnähtavat seost arendatava komponendiga

Regressioonitestid tuleks läbi viia iga redaktsiooni lõpuks, seda lisaks uue funktsionaalsuse testimisele. Käesolevas töös analüüsib autor, kuidas oleks regressioonitestimist kõige parem läbi viia andmeaidas ning millised on parimad praktikad, mida võiks samuti rakendada finantsasutuse andmeaidas, kus autor töötab testijana.

3.3 Andmeaida testijad

Levinud viga on usaldada testimisega seotud kohutused inimestele, kes ei oma piisavat kogemust antud valdkonnas. Nii suurendatakse projektikulusid ning ajaraami.

Üks esikerkiv probleem on seoses sellega, et kogemustega testijaid on tööturul vähe saadaval. Testija ametikohta peetakse hüppelauaks, kus pikemalt ei peatuta. Suur osa testijatest on noored, kes on sisenemas tööturule ning saamas oma esimest töökogemust infotehnoloogia valdkonnas. Õnneks on olukord siiski muutumas ning testimise rolli tähtsustatakse üha enam. Luuakse isegi ettevõtteid, mis keskendunud tarkvara kvaliteeti tagavate teenuste pakkumisele. Näitena võib tuua eestlaste loodud ASA Quality Services OÜ, kes on sõltuvalt projektide iseloomust ja klientide soovidest pakkunud pikaajalisi QA teenuseid kogu tarkvaraarenduse

osakonnale, teostanud konkreetse toote riist- ja/või tarkvaralist testimist ning teinud kvaliteedijuhtimise ja arenduskvaliteedi parandustöid. (Kompetents 2015)

Äriteabe ning andmeaida testimine nõuab tugevaid testimise oskusi ning aktiivset osalemist nõuete kogumisel disainimise faasis. Lisaks on äärmiselt olulised põhjalikud teadmised äriteabest ning andmeaida konseptsioonist ning tehnoloogiast. (Yaddow 2014)

Üldiselt on andmeaida testijatel vähemalt balaureusekraad või rakenduskõrgharidus ITK valdkonnas. Selle puudumisel oodatakse pikaajalist töökogemust või vähemalt mõne teemakohase kursuse lõpetamist. Eestis on hinnatud ka kutseharidus, mille võib omandada peale põhikooli.

Järgnevalt tuuakse välja mõned üldised ülesanded, mis võivad kuuluda andmeaida testijate tööülesannete hulka. Siinkohal tahab autor rõhutada, et nagu eelpool töös mainitud, siis andmeaitade arhitektuurid ning arendusmeeskondade tööjaotus on erinev. Mistõttu testimist võivad teha ka arendaja (ühistestid), andmebaasi andministraatorid (jõudlustestid, sealhulgas ka testkeskkonna ülesseadmine) ning ärikasutajad (vastuvõtutestid).

- Luua testkeskkond
- Dokumenteerida testjuhtumid
- Luua ning valideerida testskriptid
- Luua ühiktetsid ning kinnitada, et iga komponent funktsioneerib korrektselt
- Luua testid, mis kinnitavad, et iga komponentide grupp vastab spetsifikatsioonile
- Luua testid, mis kinnitavad, et uus tarkvara uuendus ei oma mõju olemasolevale töötavale funktsionaalsusele
- Luua vastuvõtutestid kinnitamaks, et äriteabe raportid töötavad nii nagu on nagu neilt oodatakse
- Luua redaksioonitestid ning toodangu valmisoleku testid
- Osaleda ETL protsesside disaini arvustustel
- Peale ETL protsessi käivitust hinnata andmeid igas tabelis eesmärgiga kontrollida, et poleks valesid andmeid, ebakorrektseid andmete formaate, duplikaate. Selleks võib testida kasutada näiteks TOAD rakendust või Excelit.
  (Yaddow *et al* 2012, 83)

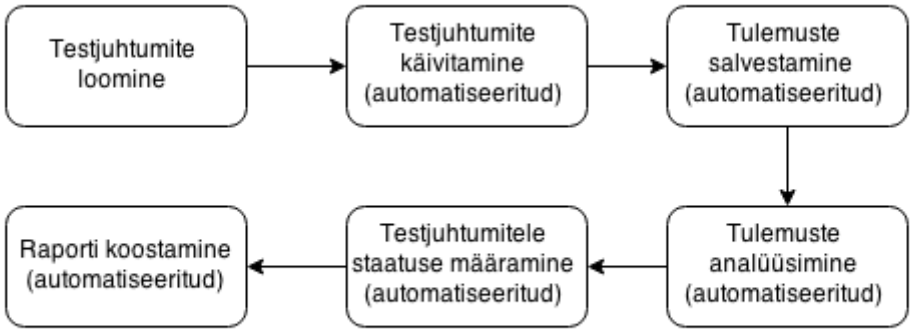
Et töö oleks efektiivsem ning väiksemate kuludega, annab autor mõnedest parimatest praktikatest nõuandeid, kuidas valida kompetentset testijat testimaks andmeaita.

4 ANALÜÜS

Käesolevas peatükis tutvustatakse saadaolevaid automatiseeritud töövahendeid andmeaida testimiseks. Analüüsitakse finantsasutuse andmeaida näitel, kas automatiseerimine muudab testimise protsessi efektiivsemaks ning kui palju teha regressioonitestimist ning millised ohud valitsevad vähesel regressioonitestimisel. On oluline, et eelnimetatud tegevusi viiksid läbi vastava kompetentsiga testijad. Analüüsitakse millised omadused ning kogemused peavad kuuluma andmeaida testijale ning esitatakse kaalutlused testijate otsinguil.

4.1 Automatiseerimine andmeaidas

Allpool oleval joonisel on välja toodud andmeaida testimise puhul läbitavad sammud. Etappidele, mis on potentsiaalseks sihtmärgiks automatiseerimisele, on lisatud märge „automatiseeritud“.



Joonis 4. Automatiseerimine andmeaidas

Nagu võib jooniselt järeldada, siis on võimalik suurel osal andmeaida testijate tööst vältida ajakulukat manuaalselt tööd. Kui palju automatiseerida ning millised etappe, sõltub olemasolevatest vahenditest ning valmidusest viia läbi muudatusi olemasolevas testimise protsessis. Testimise automatiseerimine nõuab arendustiimilt suuremat pingutust ning ei tarvitse sugugi lihtne olla, kuid kord kui see paigas on ei kujutata ette miks ei hakatud varem automatiseerimisele keskenduma.

Automatiseerimiseks on kaks erinevat teed:

- Otsida endale sobiv lahendus turul olevatest andmeaida testimisvahenditest. Kahjuks ei ole valik suur ning vahendeid, mis pakuvad komplektseid lahendusi, on vaid üksikuid. Teine puudujääk on seoses andmeaida erinevate arhitektuuridega, mistõttu ei tarvitse kõik töövahendid kohaneda ettevõtte andmeaidale. Valmis lahenduse eelisena saab välja tuua töövahendi arendajapoolse kasutajatoe, kui seda pakutakse. Vabatarkvara puhul on võimalik arendajal endal täiustada või luua juurde sobivat funktsionaalsust töövahendi rakendamisel andmeaidas.

- Arendada ise spetsiaalne lahendus, mis on kohandatud konkreetsele keskkonnale. Eelisena saab luua töövahendile lisafunktsionaalsust näiteks testjuhtumine hoiustamise näol jms. Miinuseks on arendusressursi leidmine, kuna tihti on tiimiliikmed hõivatud oma igapäevatööga ning lisaarendusi on keeruline finantseerida.

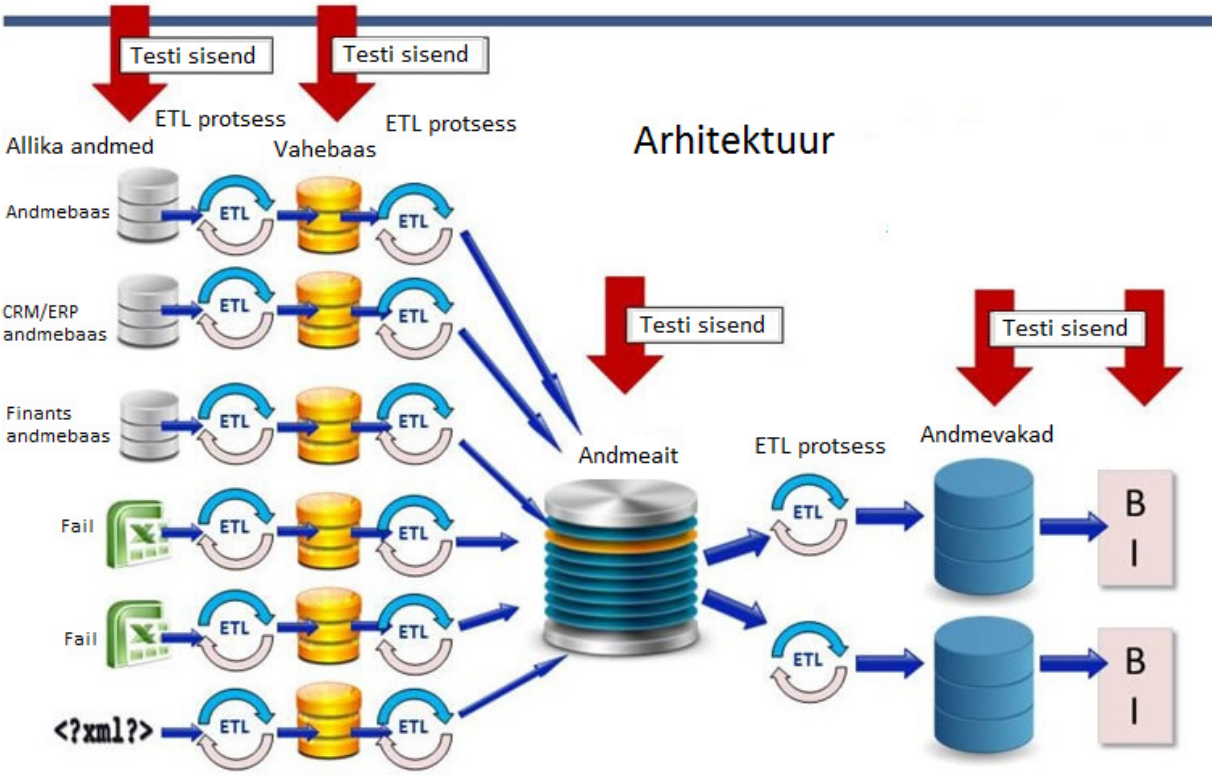
Antud töös tutvustatakse paari turul saadaolevat tarkvara, mis on loodud eesmärgiga andmeaida testimist automatiseerida. Lisaks tuuakse näitena finantsasutuse andmeait, kus testijate tiim arendab töövahendeid eesmärgiga automatiseerida oma teste ning muuta kogu testimise protsessi efektiivsemaks. Tuginedes statistikale, analüüsitakse arendatud töövahendite tegelikku mõju testimisele.

4.1.1 Testimise automatiseerimiseks arendatud töövahendid

QuerySurge

QuerySurge on spetsiaalselt andmeaida automatiseeritud testimiseks loodud töövahend, mis loodi 2004 aastal. Sellest ajast peale on jätkatud arendusega ning loodud mitmesuguseid vahendeid testimaks erinevaid projekte nagu

- Andmeaida ETL protsesside testimine
- Andmebaaside migratsioon
- Admebaaside ülendus (ingl. k *upgrade*)



Joonis 5. Querysurge arhitektuur

Querysurge automatiseeritud vahend võrdleb andmeid allikates, ETL protsesside vahebaasides ning andmeaida sihtpunktides. Vastavad sihtpunktid on näha joonisel 5 kus nendele suunavad punased nooled. Et andmeid võrrelda, tõmmatakse kõik andmed Querysurge agentide poolt operatiivandmebaasidest või failidest ning sihtkoha andmeaidest kokku QuerySurge andmebaasi, kus viiakse läbi andmete võrdlemine. Seetõttu ei oma selline teguviis mõju andmeaida jõudlusele.

- Querysurge teste saab käivitada viivitamatult, panna automaatselt jooksma vabalt valitud ajal või lasta neid käima minna mingi sündmuse tagajärjel, näitel ETL protsessi lõpetamine.
- Querysurge käivitab testid, mis automaatselt viivad läbi andmete võrdlust allika ning sihtkoha vahel, kontrollides minutite jooksul miljoneid ridu.
- Querysurge on veebipõhine rakendus. See tähendab, et pole oluline, kus andmeaida testijad füüsiliselt maailmas asuvad.
- Querysurge toetab enamikke JDBC (Java Database Connectivity) poolt toetatud andmebaase.
- Käivituse lõppedes väljastab Querysurge kokkuvõtte ning detailse raporti
- Querysurge valideerib 100% andmetest.

4.1.2 Nõuanded testide automatiseerimiseks finantsasutuse andmeaida näitel

Testides ETL protsessi ei piisa testimisest vaid vabalt valitud mõne näitega, testida tuleks andmeid 100% ulatuses. Käesolevas töös tutvustatav QuerySurge ongi loodud antud otstarbeks, kuid alati ei ole võimalik testijal selliseid automatiseeritud töövahendeid kasutada. Näiteks võib ettevõte kokku hoida taolise tarkvara litsentside pealt ning testijatel tuleb ise luua sobivad automaattestid.

Siinkohal soovib töö autor tuua välja paar näidet, mis aitavad testijal automaatselt sarnaseid andmete kontrollimise teste läbi viia. Selline praktika on rakendust leidnud finantsasutuse andmeaidas, kus on arendatud vahendid, mis testija poolt abjekti ette andmisel skripti genereerivad. Lähemalt on nendest vahenditest juttu allpool.

Et paremini iseloomustada testide automatiseerimise vajalikkust, siis olgu meil näitena tabelid *EDW.T03_CAR_old ja EDW.T03_CAR.* Mõlemal tabelil on sama ülesehitus.

ID	Brand	Model	Engine	Power	Seats	Fuel_Type	ABS	Weight
100	Jaguar	F-type	5.0 V8	495 hp	2	Petrol	Y	1665
101	BMW	X6	50i	450 hp	5	Petrol	Y	2245
102	MG	Xpower SV	4.6 i V8	320 hp	2	Multi-point injection	Y	1540
103	Dodge	Nitro	4.0 i V6	258 hp	5	Petrol	Y	2392

Tihtipeale kasutab testija tabeli ridade koguarvu, kontrollimaks, kas andmeid jõuavad korrektselt oma sihtkohta. Alljärgnev päring väljastab ridade koguarvu tabelist:

SELECT Count(*) FROM EDW.T03_CAR;

Selline testimine läbi ridade arvu summa ei tarvitse välja tuua andmeerinevused tabelites. Ridade arv võib olla sama, kuid andmed nendes ridades erinevad. Peale vaadates näeme loomulikult erinevust, kuid reaalsetes oludes on ridu miljoneid ning selline andmete kontrollimise viis ei saa olla korrektne.

Veel üks viis, kuidas testijad taolisi andmekontrolle läbi viivad, on *minus* päringud. See tähendab, et tulemusena väljastatakse read, mis eksisteerivad ühes tabelis mida pole teises tabelis.

SELECT Brand, Model, Engine, Power, Seats, Fuel_Type, ABS, Weight FROM EDW.T03_CAR
minus
SELECT Brand, Model, Engine, Power, Seats, Fuel_Type, ABS, Weight FROM EDW.T03_CAR_old

Ülalolev päring toob välja read, mis eksisteerivad tabelis *EDW.T03_CAR* ning ei eksisteeri tabelis *EDW.T03_CAR_old.* Kuid viies teste läbi *minus* päringutega, võib ette tulla mitmeid tehnilisi probleeme, mistõttu ei ole soovitatav sellist meetodit rakendada. Tehnilised takistused, mis esile kerkivad on järgmised:

- *Minus* päring on ühesuunaline. Et testida peab päringu käivitama kaks korda. Tänu sellele kahekordistub ajakulu
- Tulemus pole täpne, kui tabelis on tegemist duplikaatsete ridadega

Seetõttu ei anna *minus* päringud piisavat infot, mida me andmeerinevuste kohta tahame teada.

Finantsasutuses, kus käesoleva töö autor töötab andmeaida testijana, on arendatud taoliste olukordade testimiseks kaks skriptigeneraatorit, mis on viimase arendusena ühendatud testimise malli töövahendi külge. Täpsem kirjeldus arendatud vahenditest asub käesoleva töö järgmises peatükis. Skriptigeneraatorid on järgmised:

- tabeli veergude null väärtuste kontroll
- tabelite veeru andmete võrdlus

NULL väärtuste kontroll

Sisendina antakse ette tabeli nimetus ning väljundina genereeritakse skript, mis annab tabeli iga veeru kohta summa null väärtustest. Veerud, mis sisaldavad kahtlaselt palju NULL väärtusi, võivad olla heaks indikaatoriks vigadele. Kui palju on liiga palju peab otsustama testija, tuginedes on teadmistele antud valdkonnas või konsulteerides valdkonna spetsialistiga.

SELECT COUNT(*) cnt,
SUM(CASE WHEN Brand IS NULL THEN 1 ELSE 0 END) Acct_Source_Instr_End_Date,

SUM(CASE WHEN Model IS NULL THEN 1 ELSE 0 END) Process_Run_Id,
SUM(CASE WHEN Engine IS NULL THEN 1 ELSE 0 END) Account_Instrument_Role_Code,
SUM(CASE WHEN Power IS NULL THEN 1 ELSE 0 END) Account_Nbr_Modifier,
SUM(CASE WHEN Seats IS NULL THEN 1 ELSE 0 END) Account_Nbr,
SUM(CASE WHEN Fuel_Type IS NULL THEN 1 ELSE 0 END) Country_Context_Code,
SUM(CASE WHEN ABS IS NULL THEN 1 ELSE 0 END) Acct_Source_Instr_Start_Date,
SUM(CASE WHEN Weight IS NULL THEN 1 ELSE 0 END) Source_Instrument_Id
FROM EDW.T03_CAR
WHERE (1=1)

Näiteks selline päring annab järgneva tulemuse.

cnt	Brand	Model	Engine	Power	Seats	Fuel_Type	ABS	Weight
1021	0	0	0	1003	0	0	438	0

Ülalolevat tulemust hinnates võib järeldada, et atribuut „Power“ nõuab lähemat läbivaatust, kuna ligikaudu 98 protsendil ridadest puudub selles veerus väärtus. Võimalik, et ETL protsessi andmete transformatsioonis on viga, kuid see selgub juba lähemal vaatusel.

Andmete võrdlus

Eelmisele skriptigeneraatorile sarnaselt antakse testija poolt ette tabeli nimetused, mida soovitakse omavahel võrrelda. Väljundina genereeritakse skript, mis toob välja iga veeru kohta ridade arvu milles andmeid ei ühti kasutades selleks *full join* klauslit. Näiteks kui ETL protsess kasutab liiga palju ressursse, on tarvidus teha paketti muudatused nii, et andmed ise ei muutuks, kuid muutuks ainult paketi jõudluse parameetrid. Sellisel juhul teeb testija ETL protsessi sihttabelist varutabeli ning kontrollib peale muudetud ETL protsessi käivitust, et andmetes poleks erinevusi.

SELECT COUNT(*) cnt,
SUM(case when a.Fuel_Type <> b.Fuel_Type OR a.Fuel_Type IS NULL AND b.Fuel_Type IS NOT NULL OR a.Fuel_Type IS NOT NULL AND b.Fuel_Type IS NULL THEN 1 ELSE 0 END) Fuel_Type,
SUM(case when a.Weight <> b.Weight OR a.Weight IS NULL AND b.Weight IS NOT NULL OR a.Weight IS NOT NULL AND b.Weight IS NULL THEN 1 ELSE 0 END) Weight,

SUM(case when a.Power <> b.Power OR a.Power IS NULL AND b.Power IS NOT NULL OR a.Power IS NOT NULL AND b.Power IS NULL THEN 1 ELSE 0 END) Power,
SUM(case when a.Model <> b.Model OR a.Model IS NULL AND b.Model IS NOT NULL OR a.Model IS NOT NULL AND b.Model IS NULL THEN 1 ELSE 0 END) Model,
SUM(case when a.Brand <> b.Brand OR a.Brand IS NULL AND b.Brand IS NOT NULL OR a.Brand IS NOT NULL AND b.Brand IS NULL THEN 1 ELSE 0 END) Brand,
SUM(case when a.ABS <> b.ABS OR a.ABS IS NULL AND b.ABS IS NOT NULL OR a.ABS IS NOT NULL AND b.ABS IS NULL THEN 1 ELSE 0 END) ABS,
SUM(case when a.Seats <> b.Seats OR a.Seats IS NULL AND b.Seats IS NOT NULL OR a.Seats IS NOT NULL AND b.Seats IS NULL THEN 1 ELSE 0 END) Seats,
SUM(case when a.Engine <> b.Engine OR a.Engine IS NULL AND b.Engine IS NOT NULL OR a.Engine IS NOT NULL AND b.Engine IS NULL THEN 1 ELSE 0 END)
Source_Instrument_I
***FROM** EDW.T03_CAR_backup a*
FULL JOIN EDW.T03_CAR b ON
a.Engine=b.Engine AND
a.Model=b.Model AND
a.Power=b.Power AND
a.Brand=b.Brand AND
a.Seats=b.Seats
WHERE (1=1)

Tulemusena saadakse järgnev tabel.

cnt	Brand	Model	Engine	Power	Seats	Fuel_Type	ABS	Weight
1021	0	0	0	101	0	0	0	0

Siinkohal on näha andmete erinevus „Power“ veerus ning kui see pole lubatud, siis on test läbikukkunud.

4.1.3 Arendatud testimise vahendid finantsasutuses

Et oma töö efektiivsust tõsta, on finantsasutuse andmeaida testijate tiim arendanud järgnevad vahendid:

Testimise mall – testjuhtumite genereerimine, hoiustamine

Batman – ETL protsessi käivituse skriptide loomine

ET töövahend – ETL protsesside testid, andmeaida objektide testid

Skripti generaator – edasiarendus Batman töövahendist, pakub mugavamat käsitlust ning rohkem võimalusi.

Järgnevalt on välja toodud iga vahendi detailsem kirjeldus.

Testimise mall (test template)

Testjuhtumite haldamiseks on arendatud Exceli töövahend mis võimaldab järgnevaid tegevusi:

1) testjuhtumite jooksutamine - testija saab teha märked, milliseid testjuhtumeid ta soovib käivitada. VB macro skript loob ühenduse Teradata andmebaasiga, käivitab andmeaidas päringu ning salvestab tulemuse koos ajatempliga tagasi Exceli tabelisse. Andmeid mitte ei kirjutata üle, vaid säilitatakse testjuhtumi eelmine tulemus ning uue päringu tulemus lisatakse eelmisele otsa koos uue ajatempliga. Selline viis võimaldab anda testijale ülevaate testjuhtumi tulemuste ajaloost.
2) test casede SQL päringu genereerimine - et testjuhtumite alusel saaks saata päringuid andmeaita, genereerib macro eraldi veergu SQL päringud.
3) test casede põhjal raporti genereerimine - kui testija on oma töö lõpetanud, on tal tarvis oma tulemused vormistada testraportisse, mis saadetakse projektijuhile. Käesolev testimise mall võimaldab genereerida tekstifaili tabeli, mis sisaldab endas infot testitavate komponentide, testjuhtude ja tulemuse kohta. Testijal tarvitseb vaid käsitsi sisestada CRQ number, muudatuse valdkond ning kuupäev.

	A	B	C	D	E	F	G	H
1	Tester	Test case Id'	Component*	Description*	SQL with dynamic	Expected	Actual result*	Status*
20	Tester name	TC_01	Package_name	Performance check	SELECT CASE WHEN	OK	OK --27.04.2015	Passed
21	Tester name	TC_02						
22	Tester name	TC_03						

Joonis 6. Testimise malli kuva

BATMAN

Batman on töövahend, millega luua ETL protsesside käivitamiseks pakkfaile. Sarnaselt eelnevalt tutvustatud töövahendile on ka Batman realiseeritud Exceli macro skriptiga. Sisendina antakse testija poolt ette:

- paketi nimetus
- ajatempel – tähistab, millise perioodi kohta pakett jooksutatakse

- paketi individuaalsed parameetrid

Üldjuhul on paketil vaikimisi parameetrid seatud, kuid soovi korral saab testija neid vajaduse põhjal muuta. Näiteks kui soovitakse käivitada paketti, mis genereerib andmed arvutades need olemasolevatest andmetest andmeaidas, siis võib seada ette, et pakett arvutaks ainult kindla riigi andmed.

Batmani väjund on pakkfail (ingl. k *batch file*), mis käivitab paketi etteantud parameetritega. Testijal on võimalus luua ainult pakkfaili skript, kuid soovi korral käivitab Batmani töövahend pakkfaili automaatselt.

ET töövahend

ET töövahend on loodud automatiseerimaks andmeaida komponentide ning ETL protsesside testimist. Lahendus on samuti realiseeritud Exceli ning VB makrodena, mis ühenduvad andmeaita, kus käivitatakse päringud vastava komponendi kohta. Sisendina annab testija ette kas tabeli või vaate nime. Tulemusena genereerib töövahend testijuhtumid koos kirjelduse, SQL päringu, oodatava tulemuse ning reaalse tulemusega Exceli tabelisse samas formaadis nagu on test mallis ette nähtud. Testijal piisab vaid tulemuste ümber kopeerimisest malli. Mallis, nagu eelpool mainitud, saab testija mugavalt need testid uuesti käivitada.

ET töövahend viib läbi järgmised andmeaida komponentide testid:

1. Tabel või vaade sisaldab andmeid
2. Tabelil või vaates eksisteerib kirjeldav kommentaar
3. Kõigil tabeli või vaate veergudel eksisteerib kommentaar
4. Võrdlus domeenilistiga
5. Erwinis (Erwin Data Modeler) eksisteerib vastav mudel. Erwin on andmete modelleerimise vahend, kuhu peavad olema lisatud kõik andmeaidas kasutuses olevad objektid.
6. Tabeli või vaate reaalsed veerud ning Erwin mudelil olevad veerud ühtivad
7. Tabeli või vaade ei sisalda duplikaate
8. Tabel või vaade ei sisalda perioodilisi tühimikke. Seda testi viiakse läbi juhul, kui objekt sisaldab kahte kuupäeva, algus- ning lõppkuupäeva. Tabeli elementide perioodid peavad üksteisele järgnema, kuupäev, millal lõppeb üks periood, peab olema järgmise perioodi alguskuupäevaks.

9. Tabel või vaade ei sisalda perioodilisi ülekattuvusi. Sarnaselt eelmisele testile on ka antud juhul objekti algus- ja lõppkuupäev. Elementide perioodid ei tohi kattuda.

ET töövahend viib läbi erisugused testid, kuid andmete võrdlus jääb siiski testija ülesandeks. See tähendab, et testija, kasutatades mapping dokumenti, peab veenduma andmete korrektses teekonnas. Testid, mis ET töövahend ETL protsessidega iseseisvalt läbi viib, on järgmised:

1. Kõikidel ETL protsessidel peab olema teenuse omaniku valdkond märgitud
2. ETL protsessil on teenuse komponendi nimi defineeritud teenuse nimekirjas
3. ETL protsess peab määrama oma eeldusprotsessid
4. ETL protsess peab määrama ETL protsessid kellele ta on eeldusprotsess
5. ETL protsess peab omama Process_Status_Shortname väärtust kas 'ACTIVE', 'TO_STOP' või 'DELETED'

Lisaks on hetkel arendatud järgmised automaatsed testid, mis on testijatel töös olnud viimasel paaril redaktsioonil.

6. ETL protsessi jõudlustest
7. Kontroll, kas vahetabelid mis ETL protsess loob, on andmetega täidetud
11. Skriptide genereerimine, mis näitab, kas ETL protsessi sihttabelid on andmetega täidetud

Tulemused väjastatakse samuti Exceli tabelisse, kus testija saab need kopeerida estjuhtumite malli.

Jõudluse vaade

Et aidata testijatel silma peal hoida ka jõudlusel koostas andmebaasi administraator andmeaita spetsiaalse jõudluse vaate, mis võimaldab kergemat ligipääsu paketi käivituste jõudluse parameetritele. Iga paketi käivituse kohta on vaates saadaval järgmine info:

- ParseCpu - protsessori ressurss, mida tarbitakse SQL päringute parsimisel. CPU summa, mida kasutatakse et genereerida käivitusplaan (ingl. k execution plan). See näitaja peaks olema küllaltki väike. Normaalsel juhul paar sekundit, kõik näitajad, mis on suuremad kui 50 sekundit, on problemaatilised.

- AmpCpu – protsessori ressurss, mis on vajalik SQL päringute käivitamisel. See ei sisalda endas SQL päringu parsimist, mis on toodud välja kui ParseCPU. See number võib olla küllalt suur, kuid on ikkagi sobiv. Kõik sõltub tööst, mis on tehtud. Seetõttu pole see alati parim indikaator, näiteks kui väärtus on suur võib olla see akptseteeritav kuna tehti ka palju tööd ning vastupidi. See tähendab, et selle näitaja väärtusi tuleks vajadusel lähemalt uurida.
- TotCpu - AmpCpu ja ParseCPU summa
- IOcount – näitab sisend-väljund (ingl k. input-output) operatsioonide koguarvu SQL päringu käivitusel. Sõltub samuti töö suurusest
- SpoolMB – näitab andmemahtu, mis oli kasutusel SQL päringu käivitamisel. Sellist andmemahtu kasutatakse erinevate päringu sammude vahetulemuste salvestamiseks. Ilma sellise andmemahuta pole võimalik ühtegi päringut käivitada.
- ImpactCPU - Näitab kui palju protsessori ressursist on reserveeritud päringu käivitamisel.

Jõudluse vaade võeti kasutusele redaktsioon 72 käigus ning sellest järgneval kuul said paljud paketid paranduse, mistõttu on selles vahemikus pakettidega märgatavalt rohkem probleeme. Redaktsioon 73 vältel olid paar paketti väga problemaatilised ning seetõttu on probleemsete käivituste arv mõnevõrra suurem.

Järgnevalt on välja toodud erinevate pakettide käivitused ning kui palju nendest pakettidest on probleemsed. Näiteks redaktsioon 67 käigus on käivitatud 300 erinevat paketti, millest 53 olid jõudlusprobleemidega.

Redaksioon	**Käivitatud pakettide arv**	**Problemaatiliste pakettide arv**	**Problemaatiliste pakettide osakaal protsentides**
67	300	53	18%
68	256	47	18%
69	288	43	15%
71	348	68	20%
72	230	27	12%
73	212	26	12%
74	193	24	12%

Tabel 2. Erinevate pakettide käivitused

Allpool tabelis on näidatud kui palju tehti käivitusi kokku ning kui paljudega nendest käivitustest kaasnesid jõudlusprobleemid. Näiteks redaktsioon 67 käigus käivitati pakette kokku 5064 korda, millest 534 olid jõudlusprobleemidega.

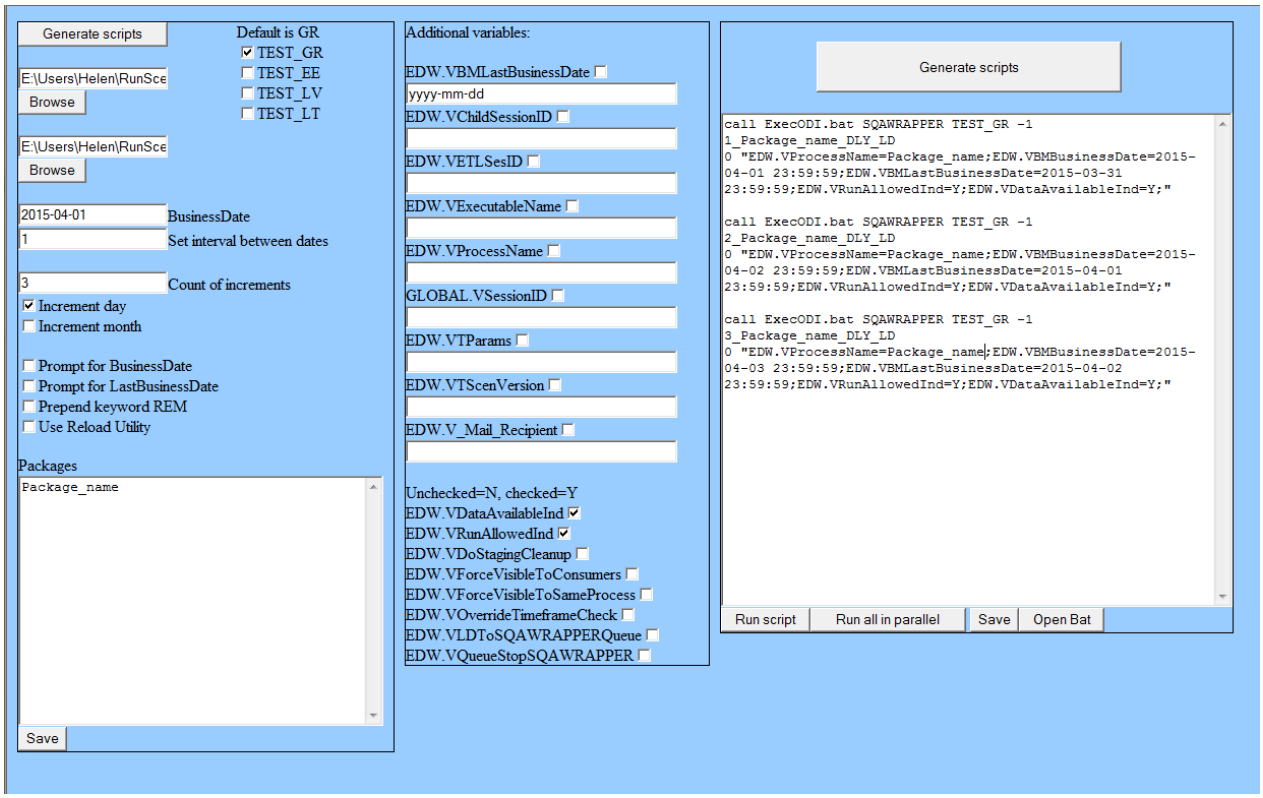
Redaksioon	**Käivitatud pakettide arv**	**Problemaatiliste pakettide arv**	**Problemaatiliste pakettide osakaal protsentides**
67	5064	534	11%
68	2871	621	22%
69	2466	609	25%
71	4402	1289	29%
72	2324	477	21%
73	1458	653	45%
74	1548	220	14,00%

Tabel 3. Kõikide pakettide kogukäivitused

Ülalolevast võib järeldada, et jõudlusproblemaatiliste pakettide osakaal on vähenenud tänu jõudlusvaate kasutuselevõtule.

Skriptigeneraator

Hetkel on koondatud ET töövahendi testid testjuhtumite malli alla. See tähendab, et testija ei pea korraga hoidma lahti kahte eraldiseisvat Exceli faili, vaid tegutsetakse ainult testimise malli failiga. Eesmärk on protsessi testija jaoks mugavamaks muuta. Nupuvajutusega määratakse, millist sorti teste soovitakse genereerida, kas andmeaida objekti või ETL protsessi teste ning valitakse testitav komponent ja pannakse testid käima. Kui testi tulemused on valmis, kuvatakse need eraldi Exceli vahelehele *Results* samas dokumendis, kus testija kopeerib need testjuhtumite vahelehele edasi. Mugavuse mõttes võiks testid olla kohe genereeritud samasse testjuhtumite lehele, et mingisugust käsitsi kopeerimist poleks vaja. Selline funktsionaalsus on hetkel arendamisel.



Joonis 7. Skriptigeneraatori ekraanikuva

ETL protsesside käivitamiseks on äsja valmis saanud skriptigeneraatori töövahend, mis pakub oma eelkäijast Batmanist rohkem võimalusi testijale. Skriptigeneraatori ekraanikuvand on nähtaval joonisel 7. Tööpõhimõte on neid kahel vahendil sama - käivitada ETL protsesse vastavalt testija ette antud parameetritele. Välja võib tuua mõned skriptigeneraatori eelised:

- Kui Batman võimaldas korraga käivitada vaid ühte ETL paketti, siis Skriptgen puhul võib testija ette anda terve nimekirja pakette, mida on tarvis käivitada. Samuti saab määrata, kas pakette soovitakse käivitada paralleelselt või järjest. Kui soovitakse pakette jooksutada paralleelselt, tuleb valida *Run script* asemel *Run in parallel*. Paralleelselt on kasulik pakette käivitada siis, kui nad kasutavad erinevaid andmeaida objekte. Siis pole ohtu, et tekivad ummikud või järjekorrad ühe objekti taga. Kui sama paketti käivitatakse mitu korda, siis tuleks neid jooksutada alati järjest.

- Kui paketti on tarvis käivitada järjest mitme päeva kohta, siis saab testija märkida *Count of increments* väljale, mitu korda järjest soovitakse paketti käivitada. Batmani miinus oli olukord, kus testijal oli tarvis käivitada näiteks kuist perioodi terve aasta kohta. Paketti jõudlust silmas pidades tuleb paketti jooksutada kuude kaupa ehk antud näite puhul 12 korda. Batmaniga opereerides tuli genereerida 12 skripti ning panna

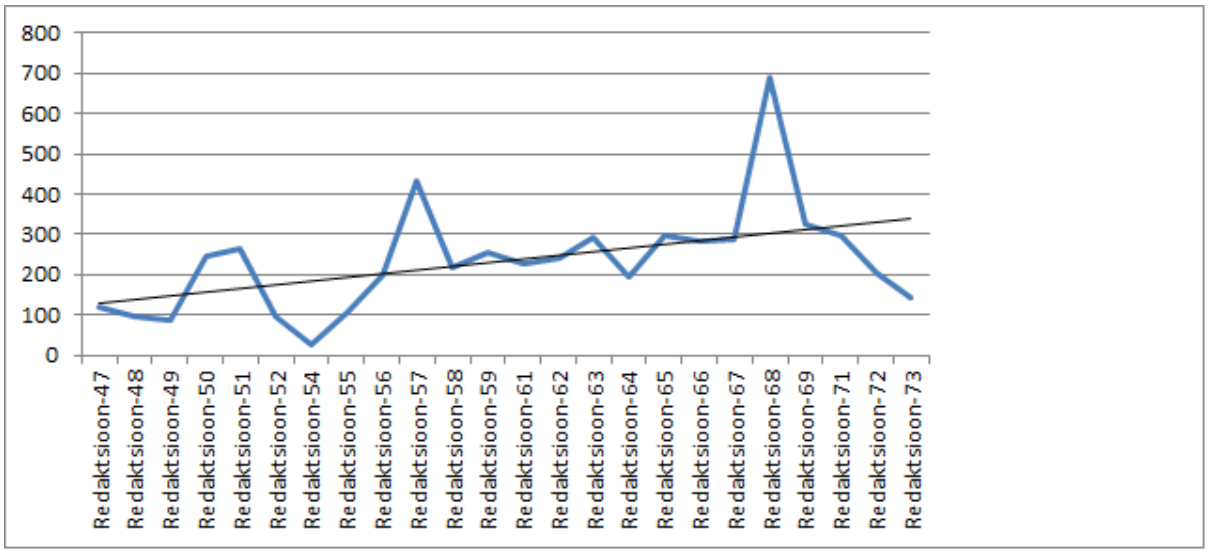
need manuaalselt kokku ühte pakkfaili, kuid Skriptgen töövahend teeb sama ülesande automaatselt.

- Pakkfaili asukoht on võimalik testijal määrata käsitsi oma soovi järgi

Kogu muu funktsionaalsus, mis oli Batmanil, on olemas ka Skriptgen töövahendil.

4.1.4 Statistika

Statistilise näitajana on suurenenud komponentide arv ühes muudatuses (CRQ). Üks muudatus sisaldab endas erinevaid komponentide muudatusi. Näiteks võib ühe muudatuse alla koondada kaks vaate muudatust, 3 tabeli muudatust ning ühe uue paketi loomise. See tähendab, et see konkreetne muudatus sisaldab kuut erinevat komponenti.



Joonis 8. Komponentide arv redaktsioonides

Ülelolevale diagrammile tuginedes saab järeldada, et sama testijate ressursi juures on suurenenud muudatuste mahud komponentide arvu näol. See tähendab, et testija on võimeline tänu testimist automatiseerivatele töövahenditele oma tööd efektiivsemalt läbi viima. Omakorda vähendab see projektikulusid ning tõstab tootlikust

4.2 Regressioonitestimine andmeaidas

Regressiooniteste tuleks teha vaid muudatuses mõjutatud komponentidele. Aeg omab suurt väärtust, seetõttu tuleb enne testimist teha kindlaks, millised komponendid on muudatusest mõjutatud.

Regressioonitestimist tuleb teha vähemalt iga kord enne muudatuse toodangusse installeerimist. Testid, mis on püsivad, korduvalt kasutatavad, lihtsad ning ei nõua testija sisendit, on head kandidaadid automatiseerimisele. Regressioonitestimist tuleb võtta sama oluliselt kui komponendi esmakordset testimist. Efektiivne regressioonitestimine säästab nii firma raha kui aega.

Et aidata otsustada, millised testjuhtumid tuleb uuesti jooksutada, tuleks testjuhtumid hoiustada prioriteetses järjekorras, arvestades riski. (Theobald, 2007). Kogu süsteemi uuesti testimine otsast peale on äärmiselt kulukas. Järgnevalt on välja toodud 3 viisi kuidas vähendada sääraseid kulusid regressioonitestide korral:

- Testjuhtumite kulu tuleneb testi tulemuse valideerimisest. Regressioonitestide puhul on tihti võimalik see faas vahele jätta ning kontrollida vaid, et testi tulemus ühtiks tulemusega, mis genereeriti eelmises iteratsioonis.
- Testide automatiseerimine aitab kaasa testimistegevuste efektiivsuse tõstmisele, seetõttu on eelnevate testide korduvkasutamine tunduvalt odavam.
- Et piirata testimise skoopi võib kasutada mõjuanalüüsi (ingl. *k impact analys*). Kokkuvõtvalt võib öelda, et mõjuanalüüsi eesmärk on määrata kindlaks, millised teised objektid on mõjutatavad konkreetse muudatuse poolt.

(Tanuška *et al* 2009, 7-11)

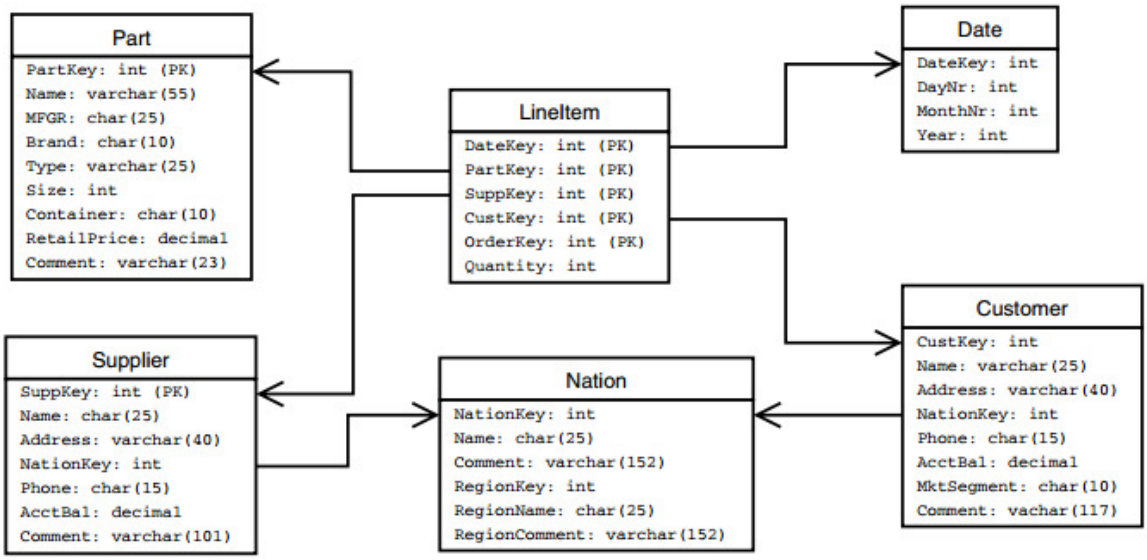
4.2.1 Regressioonitestimise automatiseerimine

Esimest regressioonitestimisele pühendatud pool-automatiseeritud töövahendit tutvustati 2006 aastal. Christian Thomsen ja Torben Bach Pedersen esitasid raamatus Data Warehouse and Knowledge Discovery ETLdiff vahendit. ETLdiff oli esimene vahend, mis oli loodud regressioonitestide läbiviimisele andmeaidas. Järgneb põgus ülevaade, mida ETLdiff vahend endas sisaldab.

ETLdiff on pool-automatiseeritud töövahend, mis mõeldud kasutamiseks regressioonitestide andmeaidas. See raamistik suudab andmemudeli põhjal automaatselt luua vajalikud testjuhtumid ning soovitused, milliseid andmeid omavahel võrrelda. Kasutaja võib siiski täpsustada vajalikke andmeid mida kaasata andmete võrdlusesse, sealhulgas tabeleid, *join*-klausleid, veerge. Seejärel loob ETLdiff nõnda nimetatud autonoomse koopia andmeaida

andmetest. Millal iganes ETL protsessi muudetakse saab vastavad autonoomset koopiat võrrelda uute andmetega ning igasugused andmeerinevused tuuakse välja.

Siinkohal olgu toodud mudelinäide, mida ETLdiff autorid oma vahendi tutvustuses esitavad. Keskel on faktitabel, *LineItem*, ning neli dimensioonitabelit.



Joonis 9. ETLdiff mudelinäide

ETLdiff tuvastab automaatselt kuus *join*-klauslit, mida täita ning millised attribuudid võrdlusest välja jätta. Et kasutada ETLdiff vahendit, on kasutajal vaja määrata kaks asja: kuidas ETL protseduuri käivitada ning kuidas saada ühendust andmeaidaga.

Samas, kui taolist regressioonitestimist teha manuaalselt, on ajaline kulu märkimisväärselt suurem. Testija peab sooritama järgmised sammud:

1. kirjutama SQL päringu, ühendama asjakohased tabelid ning valima veerud, mida võrrelda
2. kontrollima, et päring on korrektne ning sisaldab võrdluseks kõike vajalikku
3. käivitama päringu ning ning panema tulemused faili
4. kirjutama rakenduse, mis suudaks tulemusi võrrelda ning erinevused välja tuua

Lisaks eelnimetatud ülesannetele peab testija läbima veel järgmised sammud iga uue ETL versiooni puhul:

5. käivitama uue ETL versiooni
6. jooksutama päringu, mis kirjutati 1. punktis
7. käivitama rakenduse, mis võrdleb tulemusi, mis saadi punktis 6 ning punktis 3.

(Tjoa *et al* 2006)

Andmeid kirjutatakse XML dokumenti, kus neid saab hõlpsalt võrrelda, seda just jõudluskaalutlustel, kus XML pakub efektiivset mälukasutust.

Juba aastal 2006 avaldati esimene töövahend, mis võimaldas automatiseerida regressioonitestimist. Sellest on möödas 9 aastat, turul on väga lai valik automatiseerimisvahendeid tarkvaraarenduse testimise jaoks ning peame tunnistama, et andmeaida testimine on jäänud ajas maha. Uuring näitab, et 50-60 protsendiline ajakokkuhoid vähendab pingutust ning maksumust 84 protsendi võrra. (Manjunath *et al* 2012, 239-243)

4.3 Andmebaasi testijad

4.3.1 Testijate kompetents

Piiratud oskustega testijatest võib kasu asemel pigem kahju olla. Testimist kui osa arendusprotsessist kiputakse alahindama, seetõttu on kvalifikatseeritud ning motiveeritud testijad võtmekomponent. Joonas Iivonen, Mika V. Mäntylä ning Juha Itkonen avaldasid uuringu, kus nad tõid välja kõrgetasemeliste testijate põhilised tunnusjooned. Uuring tugines andmetele, mis saadi intervjueerides arendusjuhti ning kolme testijat kolmest erinevast ettevõttest. Samuti mitteametlik suhtlus inimestega arendustiimidest. Nad leidsid, et kogemused, mõtlemine, motivatsioon ning isukuomadused on vundament millele toetuda.

Kogemused ning Oskused	**Mõtlemine**	**Motivatsioon**	**Isikuomadused**
Kogemus valdkonnas	Arusaamine „suurest pildist“	Suhtumist võib iseloomustada kui „meeldib leida vigu“	Põhjalikkus, kohusetundlikkus, kannatlikkus oma töös
Kogemus programmeerimises	Arusaamine toodangukeskkonna mõjust ning puudustest	Tunnetab testimise tähtsust	Täpsus
Kogemus erinevate testimistehnoloogiatega	Iseseisev oma töös ning omab realistliku ettekujutust oma võimetest ja piiridest		Loovus
Oskus kirjutada selgesõnalisi testreporteid	Omab võimet toodet ning protsessi kritiseerida		

Tabel 4. Kõrgetasemeliste testijate karakteristikud (Iivonen *et al* 2010)

Testijate oskused, mis on kasuks andmeaida testimisel:

- arusaamine fundamentaalsetest andmeaida mõistetest. Oskus tunnetada selle kohta infotehnoloogia keskkonnas.
- Aru saada testimise protsessist kui osast andmeaida arenduses.
- Andmeaida teststrateegiate, testplaanide ning testjuhtumite arendamine. Arusaamine nende eesmärgist.
- Efektiivsete testjuhtumite ning stsenaariumite loomine andmeaidale, tuginedes äri ning kasutajanõuetele.
- Oskus lüüa kaasa andmemudelite, vastavusdokumentide ning ETL protsesside disaini ning kodeerimise ülevaatustes, et pakkuda tagasisidet disaineritele ning arendajatele.

4.3.2 Kaalutlused testijate valimisel

Testijaid värvates tuleks teha firmasisest eeltööd, milliste kogemustega testijad enim keskkonda sobivad. Määrav asjaolu võib olla kasvõi andmeaida arhitektuur. Näiteks milliste andmehoidlatega peab tulevane testija kokku puutuma. Testiiimi liikmed, kes planeerivad ning viivad läbi andmebaasitestimist, peaks omama võimalikult paljusid oskusi või kogemusi, mida Wayne Yaddow välja toob.

- Vähemalt viieaastane kogemus testimise või andmebaaside testimise valdkonnas, klient-server tehnoloogiad, mis sisaldavad viie aastast ulatuslikku kogemust andmeaitades, kasutades Informatica, SSIS või muud sarnast ETL töövahendit
- Kogemus Informatica töövahendiga või SQL Serveriga, talletatud protseduridega (stored procedures) SQL testimisega.
- Kogemus ühik ning integratsiooni testimisega, mis on seotud ETL protsessidega või talletatud protseduuridega
- Kogemus andmete verifitseerimise ühik ning integratsiooni testiplaanide ning testjuhtumite loomises, mis põhinevad tehnilisel spetsifikatsioonil.
- Demostreeritud võime kirjutada SQL liitpäringuid hõlmates mitmeid tabeleid
- Suurepärased kogemused ETL ning äriteabe valdkonnas.
- Kogemus OLAP raporteerimise töövahenditega nagu Business Objects
- Kogemus andmete migratsiooniga, andmete profileerimisega ning andmete korrastamisega
- Teadmised ETL arhitektuurist.
- Teadmised jõudlusprobleemidest ning oskused häälestada ETL protsesse jõudluse edendamiseks.
- Kogemus Oracle'i andmebaasi versioonidega 10g, 11g või 12c. Oskus kirjutada PL/SQL koodi. Teadmised salvestatud protseduuridest, funktsioonidest ning trigeritest.

(Yaddow *et al* 2012, 85)

4.3.3 Testijate tiim

Finantsasutuse andmeaidas koosneb testijate tiim ühest testijuhist ning testijatest. W. Yaddow kirjeldab oma 2012. aastal avaldatud andmeaida testimise raamatus testijuhi ning testijate tööülesanded.

Testijuht – koostab testplaani ning annab üle testide tulemused. Testplaan määratleb lähenemisviisi, kuidas toodet testitakse ning oodatud tulemused, samuti eraldab testitava toote selgelt eristatavateks osadeks ning identifitseerib toote iseärasused, mida testitakse.

Testijuht tagab, et testimine oleks graafikus ning töö saaks õigeaegselt tehtud. Vajadusel jagab testijate vahel töökoormust, kui näeb, et selleks on tarvidus.

Testija – disainib ning käivitab funktsionaalsed testjuhtumid ning analüüsib tulemusi. Samuti raporteerib testimise käigus ilmnenud vigadest ning annab testimise staatusest informatsiooni testijuhile. (Yaddow *et al* 2012, 277)

4.4 Ettepanekud

Analüüsides maailma parimaid praktikaid ning uurinud andmeaida ning andmebaaside testimist ühes Eesti suurimas finantsasutuses, on autor jõudnud tulemusteni, mille puhul leiab, et neid meetodeid rakendades on võimalik testimist efektiivsemaks muuta. Käesolevas peatükis on vastavad ettepanekud.

Autor külastas antud kirjutise raames finantsasutuse laenude osakonna andmebaasitiimi, et uurida, millised meetmed on neil kasutusele võetud eesmärgiga oma testimist efektiivsemaks muuta. Siinkohal tuuakse välja mõned suuremad erinevused kahe osakonna, laenude ja andmeaida, testimise protsessis. Kui andmeaidas testitakse muudatusi konkreetse CRQ raames, siis laenude osakonnas testitakse äriloogika järgi läbi mitme CRQ ning testijate kompetents on andmebaasi põhine. Vastavalt, kes tunneb Oracle või Microsoft SQL Serveri baase paremini. Arenduse ja reliisi testijad on samad. Ideaalis on testija sisend ärinõuete document BRS (Business Requirement Specifications), kuid tegelikult ei tarvitse BRS antud testimise objekti kohta eksisteerida. Seetõttu koguneb info testijale teisi teid pidi, näiteks koosolekud, e-kirjad, projektide wiki leheküljed, JIRA taskid jne. Viimane punkt kehtib kahjuks mõlema osakonna puhul.

Järgnevalt on välja toodud meetmed, mille puhul leiab autor, et neid võiks rakendada ka andmeaida osakonnas testimise puhul.

Regressioonitestimine

Hetkel puudub andmeaida osakonnal korrapärane regressioonitestimine, seevastu laenude osakonnas on regressioonitestimine leidnud oma koha. Iga projekti puhul on kokku on lepitud funktsionaalsus, mis on kriitiline. Ideaalis peaks regressiooninimekirja uuendama iga kuu, kuid laenude osakonnas uuendatakse seda kord kvartalis. Põhjus, miks ei tehta seda iga kuu, on täiendav ajakulu. Testijad ei jõua muude kohustuste kõrvalt pühendada oma aega regressiooninimekirja uuendamisele.

Logid andmebaasi

Tavapäraselt tekstifailides paiknevad logid kirjutatakse hoopis andmebaasi tabelitesse ning väljastatakse ka HTML kujul. Selline lähenemine annab suurepärase käideldavuse logidele. Andmebaasis korrastatud logid võimaldavad informatsiooni kiiremat sorteerimist ning leitavust.

Agiilne lähenemine

Agiilne lähenemine soosib testija võimalikult varast kaasatust projekti ning tihedat suhtlust tiimiliikmete vahel, mis omakorda annab testijale parema sisendi, kuna omatakse rohkem informatsiooni arenduse kohta. Lisaks jaotatakse arendus tükkideks ning testitakse väiksemate osade kaupa. Andmeida osakond on hetkel juurutamas samuti agiilselt lähenemist testimises ning käsil on esimene projekt, kus testijal viib testimist läbi arenduse keskkonnas. Laenude osakonnas on praktika leidnud rakendust ning agiilse arendusega jätkatakse.

Probleem laenude osakonnas

Redaktsioon viiakse läbi iga kuu, kuid äripool avaldab survet lähimat redaktsiooni üle koormata. Tähtaegadest kiputakse mitte kinni pidama. Võib öelda, et eksisteerivad distsipliiniprobleemid. Selline olukord on otsene oht vigadele, arendusel on suurem oht vigadele tekkele ning testijate ajanappus võib lasta vead toodangusse, kus vigade parandamine tekitab lisaväljaminekuid ning ajakulu. Seetõttu on andmeaida osakond loonud eraldi töökoha redaktsiooni koordinaatorile, kes vastutab selle eest, et tähtaegadest peetakse kinni ning ei tekiks ülekoormatust redaktsioonidel. Selline lahendus on kenasti toiminud juba aastaid ning on kindlasti vajalik neile ettevõtetele, kes tunnevad sama probleemi olemasolu.

4.4.1 Testvahendite arendamine finantsasutuse andmeaidas

Esimesed automatiseeritud töövahendid loodi kolm aastat tagasi. Sellest ajast saadik tegeleb testijate tiim andmeaida testimise parendamisega ning abitöövahendite arendamisega. Eelmises peatükis tutvustatud töövahendid on sellise arenduse väljund.

Edaspidiseks eesmärgiks võiks olla veebiliidese arendus, kus kõik automaattestid on arendatud ühtsele platvormile, kus testija saab teste läbi veebiliidese jooksutada. Sellises keskkonnas saab testija viia läbi nii andmeaida komponentide kui ka ETL protsesside testid. Samuti hallatakse samas keskkonnas ka testjuhtumeid.

4.4.2 Uute testijate väljaõpe

Tulles tagasi autori finantsasutuse laenude osakonna külastusele, siis lisaks eelnevale kõneldi ka uute testijate väljaõppest ning võrreldi osakondade talitusviise. Üldiselt ühtivad kahe osakonna meetodid, kuidas uusi testijaid välja õpetatakse ning järgnevalt on välja toodud praktikad, mida autor leiab hästi sobivat testija koolitamisel.

- Testimise juhtnöörid - testimise meeskonnal dokumenteeritud juhtnöörid, mis on abiks nii uuele kui kogenud testijale. Kuna testitavaid komponente on erinevaid - tabelid, makrod, vaated, paketid siis eksisteerivad igal komponendil üldised testid mis iga kord tuleb käivitada. Näiteks on kirja pandud millised testijuhtumid tuleb läbida vaate muudatuse või paketi kustutamise puhul.
- Mentorlus - igale uuele testijale määratakse mentor, kes on kogemustega testija ning suudab tuge pakkuda alustavale testijale. Mentori ülesanne on hoolitseda selle eest, et uus testija kohaneks tööga ning varustada teda vajaliku informatsiooniga. Lisaks aitab mentorlus testijal paremini tiimi sisse sulanduda tänu tihedale suhtlusele teiste tiimi liikmetega.
- Iga uus testija täiendab testimise juhtnööre vastavalt sellele, kuidas ta näeb antud dokumendis puudujääke. Kui kogenud testijad koostavad juhtnööre, siis nad ei tarvitse leida ühist vaatenurka uue ning kogenematu testijaga. Seetõttu on hea praktika lasta testijal endal kirjutada juhtnöörid tulevasele testijale, kes alustab tööd samas ametis peale teda.

5 Kokkuvõte

Selle töö eesmärk on aidata andmeaida testijatel leida efektiivsemaid viise oma töö läbiviimiseks. Keskendutakse kolmele valdkonnale: testimise automatiseerimine, regressioonitestimine ning testijate kompetents.

Et saavutada vastused püstitatud probleemidele, tugineti maailma parimatele praktikatele ning kogemustele nii palju kui nende kohta infot võis leida. Tuleb tunnistada, et andmebaasi testimise töövahendid ning kirjandus on maha jäänud enamlevinud tarkvara testimisest, kuid viimastel aastatel on antud teema muutunud märgatavamalt populaarsemaks. Töö autor alustas käesoleva töö jaoks materjalide otsimist 2014. aasta sügisel ning nüüd, 2015. aasta kevadel veendus oma silmaga, kuidas töövahendeid ning artikleid on internetiavarustesse silmanähtavalt juurde tekkinud.

Püstitatud probleemidele - nagu ebaotstarbekas manuaalne testimine, regressioonitestide alahindamine ning testijate kompetents - on autoril esitatud mitmed võimalikud lahendused.

Käesolevast järeldame esiteks, et andmeaida testimise automatiseerimine (osaliselt regressioonitestidele) ning vastavad töövahendid on edu võti. Puhas manuaalne testimine on nii ülemäära aeglane kui ka ebaefektiivne. Ettevõtted ei saa lubada defektseid andmeid oma andmeaidas, millel põhinevad kaalukad otsused firma käekäigus.

Teiseks järeldame, et testitiimi kokkupanemisel tuleb hoolikalt läbi mõelda, milliste kogemustega inimesed toodavad suurimat väärtust. Siin töös toob autor välja, millised on edukate testijate karakteristikud ning milliseid meetmeid saavad kasutada testitiimid uue testija puhul, et ta saaks kiiremini oma panust anda.

6 Summary

The aim of this paper is to help data warehouse testers to improve their work efficiency and quality.The three main fields are concentrated on: test automation process, regression testing and the competence of testing.

World's best practices and experiences were studied, although quite little resources are available and these are often outdated. Still, recently the topic area has gained more publicity and popularity.

The outcome of this research paper is that purely manual testing is too slow and inefficient. Companies cannot afford defective data in their data warehouse, therefore test automation process could be a workable and feasible solution.

Secondly, human resources cannot be underestimated. The author brings out the characteristics of the successful testers and suggests how new workers can adjust more quickly to the testing procedures.

Kasutatud kirjandus

Chhaperia V. (2015). ETL Testing / Data Warehouse Testing – Tips, Techniques, Process and Challenges [WWW] http://www.softwaretestinghelp.com/etl-testing-data-warehouse-testing/ (01.04.2015)

Deshpande K., Desai B. (2014). A critical study of requirement gathering and testing techniques for data warehousing. - *International journal of information technology & management information system (IJITMIS).* 5(1), 60-71. [WWW] http://www.iaeme.com/MasterAdmin/UploadFolder/50320140501006/50320140501006.pdf (27.02.2015)

Golfarelli M., Rizzi, S. (2009). A Comprehensive Approach to Data Warehouse Testing – DOLAP. [WWW] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.511.7898 (02.04.2015)

Gupta S. L., Pahwa P., Mathur S. (2012) Classification of data warehouse testing approaches - *International Journal of Computers & Technology*, 3(3), 381-386. [WWW] http://cirworld.com/journals/index.php/ijct/article/viewFile/558/pdf (23.03.2015)

Iivonen J., Mäntylä V. M., Itkonen J. (2010). Characteristics of High Performing Testers – A Case Study [WWW] http://www.soberit.hut.fi/espa/publications/ESEM2010_Short_papers_characteristics_v10.pdf (17.01.2015)

Inmon, W. H. (1995). What is a Data Warehouse? - *Prism Tech Topic,* 1(1)

Jovanovi, I. Software Testing Methods and Techniques [WWW] http://www.internetjournals.net/journals/tir/2009/January/Paper%2006.pdf (14.03.2015)

Khan M. E., Khan F. (2012). A Comparative Study of White Box, Black Box and Grey Box Testing Techniques [WWW] http://thesai.org/Downloads/Volume3No6/Paper%203-A%20Comparative%20Study%20of%20White%20Box,%20Black%20Box%20and%20Grey%20Box%20Testing%20Techniques.pdf (01.05.2015)

Kimball R. (1996). The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. Canada : John Wiley & Sons, Inc.

Kimball R. (2004). The Data WarehouseETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Canada : Wiley Publishing, Inc.

Kompetents [WWW] https://www.asaquality.ee/ettevottest/kompetents (01.05.2015)

Liu, X., Song, X. (2011). An approach for designing, modeling and realizing etl processes based on unified views model. - *International Journal of Software Engineering & Knowledge Engineering.* 21(4), 543-570. [Online] EBSCOhost Web. Business Source Complete (03.03.2015).

Manjunath T.N, Ravindra S. H., Yogish H.K., Archana R.A.,Umesh I.M. (2012). A case study on regression test automation for data warehouse quality assurance - *International Journal of Information Technology and Knowledge Management.* 5(2), 239-243

Rouse M. (2009) Business intelligence (BI) [WWW] http://searchdatamanagement.techtarget.com/definition/business-intelligence (14.03.2015)

Tanuška P., Moravčík O., Važan P., Miksa F. (2009). The Proposal of Data Warehouse Testing Activities - Proceedings of the 20th Central European Conference on Information and Intelligent Systems, 7-11. [WWW] http://www.ceciis.foi.hr/app/index.php/ceciis/2009/paper/download/235/178 (01.05.2015)

Theobald J. (2007). Strategies for Testing Data Warehouse Applications. - DM Review. 17(6), 21-41. [Online] EBSCOhost Web. Business Source Complete (03.03.2015).

Tjoa A. M., Juan Trujillo. (2006). Data Warehousing and Knowledge Discovery. Poland : Springer-Verlag Berlin Heidelberg

Vallaste, H. E-teatmik [WWW] http://vallaste.ee/index.htm (09.04.2015)

What is a Data Warehouse and How Do I Test It? (2012) [WWW] http://www.slideshare.net/RTTS/what-is-a-data-warehouse-and-how-do-i-test-it (01.05.2015)

What is Software Testing? [WWW] http://istqbexamcertification.com/what-is-a-software-testing/ (14.03.2015)

Williams L. (2006). Testing Overview and Black-Box Testing Techniques [WWW] http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf (05.02.2015)

Yaddow W. (2013). Data Warehouse Testing: Part 1 [WWW] http://ibmdatamag.com/2013/07/data-warehouse-testing-part-1/ (10.04.2015)

Yaddow W., Vucevic D. (2012). Testing the Data Warehouse Practicum. United States of America : Trafford Publishing.

Yaddow, W. (2014). Meeting the Fundamental Challenges of Data Warehouse Testing. - Business Intelligence Journal. 19(3), 28-34. [Online] EBSCOhost Web. Business Source Complete (03.03.2015).