

Topi Pasanen

TESTIAUTOMAATIOKEHITTÄJÄN PÄIVÄKIRJA

TESTIAUTOMAATIOKEHITTÄJÄN PÄIVÄKIRJA

Topi Pasanen
Opinnäytetyö
Syksy 2021
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Topi Pasanen

Opinnäytetyön nimi: Testiautomaatiokehittäjän päiväkirja

Työn ohjaaja: Jaakko Kaski

Työn valmistumislukukausi ja -vuosi: Syksy 2021

Sivumäärä: 32

Opinnäytetyö on päiväkirjamuodossa tehtävä tutkielma, jonka tavoitteena on kuvata työtäni testi-automaatiokehittäjänä sekä havainnoida siinä ilmeneviä ongelmia ja ratkaisuja. Toimeksiantajana on kotimainen tietoliikennealan suurtekijä, Nokia Oyj. Opinnäytetyö koostuu johdannosta, taustatiedoista, tavoitteista, päiväkirjasta sekä pohdinnasta.

Johdanto kuvaa tekijän työtehtäviä sekä asemaa yrityksessä. Taustatiedoissa kerrotaan enemmän työpaikasta, osaamisesta ja työtehtävistä. Tavoitteissa kerrotaan työn tavoitteista kuten testiautomaation syvempi ymmärtäminen ja koodaamisen harjoittelu. Päiväkirjassa kuvataan päivä kerrallaan, kuinka työ etenee pohdintojen kera. Pohdinnassa havainnoidaan opinnäytetyöntekoprosessia kokonaisuudessaan.

Opinnäytetyön tekemisen aikana kohokohdiksi ilmaantui ongelmanratkaisun ylä- ja alamäet sekä ratkaisuhetken huuma.

Asiasanat: ohjelmistokehitys, testiautomaatio, Python, Linux, päiväkirjamuotoinen opinnäytetyö

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Software Development

Author(s): Topi Pasanen
Title of thesis: Diary of Test automation developer
Supervisor(s): Jaakko Kaski
Term and year when the thesis was submitted: Fall 2021
Number of pages: 32

My thesis is done in a diary-like study. Its purpose is to portray an image of work from test automation developer's point of view and to analyze problems and their solutions when they come up during the work. The employer is a Finnish company specializing in telecommunications and information technology, Nokia Oyj. The thesis consists of introduction, background knowledge, objectives, diary and conclusions.

Introduction describes the worker's tasks and position in the company. Background knowledge tells more about the workplace itself, capabilities of the worker and of his tasks. In the objectives there is more of the goals of this project. The diary describes the process of workday at a time. Conclusions is for overall observation of the process.

The highlights of this project are the ups and downs of software development with spiced with the high spirits of success.

Keywords: software development, test automation, Python, Linux, diary-like thesis

SISÄLLYS

SANASTO.....	6
1 JOHDANTO.....	7
2 TAUSTATIEDOT	8
2.1 Yritys ja työympäristö	8
2.2 Työtehtävät ja osaaminen	8
3 TARKOITUS JA TAVOITEET	10
4 PÄIVÄKIRJA.....	11
4.1 Viikko 1.....	11
4.2 Viikko 2.....	11
4.3 Viikko 3.....	12
4.4 Viikko 4.....	13
4.5 Viikko 5.....	14
4.6 Viikko 6.....	15
4.7 Viikko 7.....	16
4.8 Viikko 8.....	17
4.9 Viikko 9.....	19
4.10 Viikko 10.....	19
4.11 Viikko 11.....	20
4.12 Viikko 12.....	21
4.13 Viikko 13.....	21
4.14 Viikko 14.....	22
4.15 Viikko 15.....	23
4.16 Viikko 16.....	24
4.17 Viikko 17.....	25
4.18 Viikko 18.....	26
4.19 Viikko 19.....	28
4.20 Viikko 20.....	28
5 POHDINTA.....	30
LÄHTEET.....	32

SANASTO

Git/GitLab	Versionhallintajärjestelmä
GUID	128-bittinen satunnainen luku, joka on tarkoitettu tunnusluvuksi
Im analyser tool	Sovellus, jolla voidaan muuntaa tukiaseman uudelleenkäynnistämisen kriittisten pisteiden aikaleimoja
Iron Python	Pythonista kustomoitu ohjelmointikieli
Pip-paketti	Modulaarisia paketteja, jotka lisäävät ominaisuuksia Python-ohjelmointiin
Proxy server	Käyttäjän ja serverin välillä oleva väline, joka tulkkaa näiden kahden välillä olevan liikenteen
PyCharm	Ohjelmointiympäristö, joka soveltuu hyvin Pythonin koodaamiseen
Python	Paljon käytetty ohjelmointikieli
Virtuaaliympäristö	Ei-fyysinen ympäristö, jonne voi tehdä ohjelmallisia muutoksia ilman, että se on ristiriidassa fyysisten ympäristöjen kanssa
VPN	Virtual Private Network eli virtuaalinen erillisverkko on tapa, jolla kaksi tai useampia yrityksen verkkoja voidaan yhdistää julkisen verkon yli

1 JOHDANTO

Päiväkirjamuotoinen opinnäytetyö kertoo työstäni ohjelmistokehittäjänä viiden kuukauden ajanjaksolta. Opinnäytetyössä pohdin vastaan tulevia ongelmia sekä niiden ratkaisuja. Päiväkirjaseuranta on aikaväliltä 20.1.–11.6.2021. Päiväkirjaosuudessa esitetään jokainen viikko omana osanaan, joissa kirjaan ylös viikon haasteita ja pohdintoja.

Toimeksiantajana toimii Nokia Oyj, joka on yksi Suomen suurimmista yrityksistä, joka on erikoistunut tietoliikennealaan. Koronan aiheuttamien rajoitusten vuoksi työ suoritetaan kokonaan etätyöllä kotoani.

Opinnäytetyöni tehtävät koostuvat pääasiallisesti ohjelmistokehityksestä sekä testaamisesta. Tärkeimmät työssäni käytettävät koodikielet ovat tuorein Python sekä vanhempi Pythonin variaatio, Iron Python. Tärkein käytettävä taito työssäni on ongelmanratkaisutaito.

2 TAUSTATIEDOT

2.1 Yritys ja työympäristö

Toimeksiantajana toimii Nokia Oyj, jonka pääkonttori löytyy Espoosta (1). Suomessa on toimipisteet myös Oulussa ja Tampereella. Nokialla on myös lukuisia toimipisteitä ympäri maailmaa. Nokia on tietoliikennealan yhtiö, jonka pääasialliset tuotteet ovat verkkoinfrastruktuurin, teknologiankehityksen ja lisensoinnin puolella. Nokian suurimpia tuotteita ovat 4G- ja koko ajan kasvavassa määrin 5G-tukiasemat ja niiden komponentit. Yrityksen tavoitteena on luoda teknologiaa, jonka avulla maailma voi toimia yhdessä. Tämän saavuttamiseksi Nokia tuottaa jatkuvaa innovaatiota matkapuhelin-, kiinteissä sekä pilvipalveluverkoissa.

2.2 Työtehtävät ja osaaminen

Aloitin työni Nokialla 2019 kesätyöläisenä. Työni on tuotekehitystiimissä ja työkuvani koostuu pääasiassa LTE-tukiasemien uusien sovelluspakettien testaamisesta. Yleisimmät työtehtäväni sisältävät virheraporttien kirjoittamista, automaation ajamista ja raportointia sekä ongelmanratkaisua yhteistyössä sovelluskehittäjien kanssa.

Viimeisin uutta luova projekti, jossa olen ollut mukana, vaati uuden ominaisuuden testaamista tukiaseman radioiden tehostamiseksi. Vastuulleni kuului yhden asiakasympäristön jatkuva testaaminen ja virheiden raportointi seuloen uuteen ominaisuuteen liittyviä virheitä.

Opinnäytetyön tehtävä on kokonaan uutta osaamista itselleni. Tähänastinen työni on ollut testaamista ja siihen liittyviä prosesseja. Nyt minun tulee opetella ymmärtämään uutta ja vanhaa Pythonia ja ennen kaikkea, kuinka koodata niillä. Python-koodaamisesta olen käynyt kurssuja kouluni puolesta sekä Nokian omilla sisäisillä kursseilla, mutta tiedän valmiiksi, että opinnäytetyöprojektini tulee olemaan erilaista. Tehtävien tekemisen sijaan opinnäytetyössäni luon uutta.

Suurimmat haasteet uskon tulevan Pythonin soveltamisesta. Aloitusvaiheessa on jo selvää, että minun tulee saada Python 2.7 ja Python 3.7 tai uudempi toimimaan harmonisesti, eikä tästä ole ohjeita etukäteen. Toiseksi haastavaa tulee olemaan koodin tekeminen tiimissä käytössä olevalle

automaatiotyökalulle. Työkalulle on jo valmiiksi tuhansia rivejä koodia, joten joudun sopeuttamaan oman koodini näiden mukaisesti.

Koen oman osaamiseni olevan riittävä työn valmiiksi saamiseksi. Ammattikorkeakoulutukseni on suurimmin osin sisältänyt koodaamista, joten perusosaaminen on kunnossa. Päivittäinen työni sisältää myös paljon ongelmanratkaisua, joka tulee olemaan avainasemassa projektini valmistumisessa. Voin myös tehdä opinnäytetyötä työaikana, joten ajan uskon riittävän mainiosti.

3 TARKOITUS JA TAVOITEET

Testiautomaation syvempi ymmärrys

Testaamisessa ajetaan jatkuvasti erinäisiä automaatiokriptejä. Tiedän, mitä ne tekevät ja osaan lukea lokitiedostoista, mitä tukiasemalle on missäkin testaamisen vaiheessa tapahtunut, mutta en siltikään tiedä, millä tapaa asiat tapahtuvat koodissa. Ymmärtämällä paremmin koodin kulkua voin ongelmatilanteissa selkeämmin sanoa, että onko vika automaatiossa, testattavassa sovelluksessa vaiko raudassa.

Uuden ominaisuuden luominen

Ilmeisin tavoite on saada tuleva sovellukseni valmiiksi. Sovellus tulee helpottamaan tukiasemasta saatavan lokidatan syvempää analyysiä, sillä Im analyser tool tuo tärkeitä tarkastuspisteiden ajoituksia yhdessä paketissa.

Koodaamisen oppiminen

Työpaikallani on paljon muutakin tehtävää, kuin testaaminen. Laajentamalla osaamista koodaamisen puolelle voin tulevaisuudessa laajentaa työnkuvaani ja täten edetä urallani. Tiimissäni on myös puhuttu tarpeesta saada osaavia sovelluskehittäjiä uusiin projekteihin.

4 PÄIVÄKIRJA

4.1 Viikko 1

20.1.2021

Kartoitin integraation toteutuksen vaatimuksia. Im analyser tool käyttää Python 3.7.x-versiota ja päivityksien myötä aina uudempaa. Testaamisessa käytettävä automaatiotyökalu puolestaan käyttää Iron Pythonia, joka on 2.7.x-versio, joka ei ole suoraan yhteensopiva Im analyser toolin käyttämän version kanssa. Pehdyin mahdollisen oman kirjaston luomiseen, jota voitaisiin käyttää yhtäaikaiseen ajamiseen.

Erillinen kirjasto yksinään ei välttämättä kuitenkaan riitä. Im analyser toolin ajaminen vaatii tietokoneella olevan Python 3.x.x -version, jota ei oletusarvoisesti ole kaikkien käyttäjien koneella, joten tätä varten pitäisi olla jokin keino, jolla kierretään erillinen Pythonin päivitys. Yksi mahdollisuus olisi luoda työkalu, joka ajaa edellä mainitun työkalun virtuaaliympäristössä, jossa yhteensopiva Python.

Viikkoanalyysi:

Kartoitin integroitavien sovelluksien käyttämiä Python-versioita ja ongelmakohta tulee olemaan 2.7.x- ja 3.6.x-versioiden yhteen sovittaminen. Virtuaaliympäristöt lienevät oikea ratkaisu tähän tilanteeseen

4.2 Viikko 2

28.1.2021

Virtuaalisen ympäristön käyttöönotto osoittautui vaikeammaksi, mitä luulin. Sain asennettua tyhjän ympäristön työkoneelleni, mutta pakettien lataaminen takkuilee paljon. "ERROR: Could not find a version that satisfies the requirement requests" -ilmoitus tulee jokaisen latausyrityksen jälkeen. Koetin selvittää voisikohan työpaikalleni pakollinen VPN-yhteys olla jollakin tapaa tiellä pakettien lataamisessa. On myös mahdollista, että sivusto pypi.org, jota koetin saada kiinni ei vastaa syystä tai toisesta. Virtuaalisen ympäristön asentamisella haetaan vapaata ympäristöä,

jossa voi kokeilla eri versioita Pythonista ilman, että sotken jo vallitsevia Pythonin asennuksia ja asetuksia.

29.1.2021

Iron Pythoniin on luotu keino luoda virtuaaliympäristöjä, mutta yhteensopivuus 3.9 Pythonin kanssa on ongelma. Automaatiossa käytettävä sovellus ajaa Iron Pythonia, joten pitäisi siis luoda sinne testiskripti, joka luo virtuaalisen ympäristön, asentaa sinne Python 3.6+ tai uudemman ja sitten ajaa Im analyser toolin. Suurin osa päivittäisestä tutkimisesta meni Iron Pythoniin virtuaaliympäristöön paneutuessa. Seuraavaksi tulisi koettaa saada pystyyn virtuaalinen ympäristö, jossa on Iron Python, ja katsoa, voiko sen sisään vielä luoda toisen virtuaaliympäristön.

Viikkoanalyysi:

Tutkin virtuaaliympäristöjen toiminta- sekä luomisperiaatteita. Kohtasin kuitenkin paljon ongelmia sopivien Python sekä pip-pakettien lataamisessa. Tulee virheilmoitus, että paketit eivät mene perille, eli vaikuttaisi verkko-ongelmalta. Voisikohan ongelmana olla töissä käytettävä VPN-yhteys.

4.3 Viikko 3

1.2.2021

Pidin Nokian puolelta saamani ohjaajan kanssa palaverin ja totesimme, että minun täytyy opiskella meillä käytettäviä skriptejä, ennen kuin pystyn koodaamaan mitään. Suunnitelmaa hahmoteltiin sen verran, että virtuaaliympäristöön asennetaan tuorein Python-paketti, jolla voidaan ajaa Im analyser tool. Tämä sen takia, että muiden käyttäjien koneella voi olla mikä tahansa Python-versio, joka voi tuoda ongelmia Im analyser toolin kanssa. Virtuaaliympäristöllä vältetään tämä kokonaan. Myös Im analyserille luodaan työkalu, joka päivittää viimeisimmän version käyttöön.

4.2.2021

Käytin aikaa automaation skriptin koodiin tutustumiseen. Execute-osiosta opettelin tapahtumien kulkua yksittäisen tukiaseman uudelleen käynnistämisen sisällä. Latasin PyCharmin, jotta voin paremmin luoda koodia, mutta virtuaaliympäristön luominen ei jostain syystä onnistu. Lienee, että pitää laittaa johonkin kohtaan IDE:ä proxy serverit, jotta yhteys toimii VPN:stä huolimatta.

Viikkoanalyysi:

Tutustuin firman automaatiotyökalun skripteihin ja niiden luomiseen. Latasin myös PyCharm-ohjelmointiympäristön, jotta koodaaminen sujuisi paremmin. Minulla on edelleen ongelmia saada paketit ladattua.

4.4 Viikko 4

8.2.2021

Perehdyin ohjaajani opastamana automaatio-ohjelman skriptien ajoon ja sain myös proxy serverit toimimaan ohjelmointiympäristössä sekä komentoriviltä. Seuraavaksi aion opiskella yksinkertaisen skriptien luomista ja totutella meillä Nokialla käytettävään Iron Pythonin variaatioon. Kului aikaa tajuta, mitkä muuttujat vastaavat tiedoston nimeä ja miten saan muunnetun skriptin ajoon, mutta ongelmana oli, että piti generoida oma uniikki GUID.

10.2.2021

Kokeilin tänään luoda aluksi erillisen kansion erikseen määriteltyyn polkuun, jonne myöhemmin sitten laitan virtuaaliympäristön (2), mutta en toistaiseksi ole onnistunut. En vielä ole sisäistänyt, kuinka Nokian automaatio-ohjelma toimii, että saisin toimivaa skriptiä. Välttääkseni kankeaa koodia käytän os-kirjastoa koodissani.

12.2.2021

Minulla on edelleen ollut kansion luonnin kanssa ongelmia. Virheilmoituksissa valitetaan verkko-yhteydestä, vaikka sitä ei edes tähän tarvitse. Execute-osiossa koodia on jotakin hämärää, koska se ei suotu tulostamaan ulos edes yksinkertaisinta print-komentoa. Virheiden etsimisessä ja poistamisessa on ollut ongelmia, koska en toistaiseksi ole saanut mitään ulos Nokian automaatio-ohjelmalla

Viikkoanalyysi:

Vihdoin sain proxy serverit toimimaan ja data alkoi kulkemaan. Kuitenkin on paljon ongelmia Nokian oman automaatiotyökalun kanssa. En saa yksinkertaisimpiakaan komentoja näkyviin. Ohjaaja lupasi opastaa enemmän sen käyttöä, niin saan varmasti selkeyttä tähän.

4.5 Viikko 5

15.2.2021

Asiat ovat edenneet huomattavasti. Automaatio-ohjelman ajamisen ongelmien syyppäänä oli testiympäristöasetuksiin kummittelemaan jäänyt ympäristö. Sen poistamisen jälkeen skripteihin tehdyt muutokset alkoivat näkyä ja virheiden etsimisestä tuli jälleen mahdollista.

Nyt saimme onnistuneesti luotua kansion virtuaaliympäristöä varten käyttämällä `open()`-funktioita (3). Tämä oli kuitenkin etukäteen määriteltyyn osoitteeseen omalla tietokoneellani, mikä ei riitä, sillä jokaisella käyttäjällä pitää olla vapaus määrittää, mihin ympäristönsä luo. Tästä syystä ohjajani opetti minulle automaatio-ohjelman mukaisien asetusten luomista, jotta käyttäjä saa itse valita selaamalla, mihin kansioon virtuaalinen ympäristö luodaan. Tämä tehtiin luomalla `if not` -lause OS-kirjastoa käyttäen. Tällä tarkistetaan, onko jo valitussa polussa tiedosto valmiina. Työprosessia hieman hidasti PyCharmiin ja muihin oheisohjelmiin tutustuminen, kun huomasin tarpeellisten asetusten puuttuvan. Esimerkiksi Notepad++:ssa piti laittaa kieliasetuksista, että välit lasketaan välilyönteinä eikä TAB-väleinä, jotta Nokian oma automaatio-ohjelma tunnistaa koodin oikeaksi. Seuraavana pitää alkaa tutustumaan varsinaisen virtuaaliympäristön luomiseen oikealla Python -versiolla.

17.2.2021

Yritin saada virtuaaliympäristön uuteen kansiooni, mutta sen luominen os-kirjaston kautta tuottaa ongelmia. Työskentelen yhdessä virtuaaliympäristössä, jotta en sotke työkoneneeni asetuksia. Iron Pythonilla tuntuu olevan hyvin vähän valmiuksia virtuaaliympäristön luomiseen. Tähän asti olen löytänyt, vain käyttäjien omia hyvin epävakaita sovelluksia aiheesta. Itsekin yritän luoda samantyyppisiä.

18.2.2021

Ironista kyllä tajusin, että minunhan tulee luoda virtuaaliympäristö nimenomaan Python 3:lle. Olen koko ajan yrittänyt luoda Python 2.7.x-versioille. Ongelma on silti sama, kuin aikaisemmin Python 2:lla. Virtuaaliympäristön luominen ei tähän asti ole onnistunut edes vallitsevaan Python 3:een. Olen katsonut lukuisia ohjeita, mutta heidän automaatiotyökalun käyttämät kirjastot eivät toisistaan ole antaneet periksi. Tiedustelin työkavereiltani, että millaisen tahtoisivat tulevasta sovelmusintegraatiosta. Pääasiana tuntui olevan log -tiedostojen vaivaton ulossaanti. Muut osat jäävät siis minun vastuulleni.

Viikkoanalyysi:

Tämän viikon tekemiset olivat suurimmin osin käytettäviin työkaluihin perehtymistä. Työni eteni pienin askelin, kun sain jo itsekin muunneltua automaatiotyökalun asetuksia haluamakseni. Suurin voitto tälle viikolle kuitenkin oli, kun tajusin saavani hieman vapautta virtuaaliympäristön luomiseen, sillä tarvitsen sen nimenomaan Python 3.7:lle enkä 2.7:lle.

4.6 Viikko 6

23.2.2021

Virtuaaliympäristön luonti ei vielä ole onnistunut, mutta korjasin ainakin yhden ongelman. Muutetuani kansion, johon virtuaaliympäristö luodaan käyttäjän omaksi valinnaksi, katosi mkdir -komennolta pohja eikä tiedostoa enää luotu mihinkään. Laitoin luotavan kansion nimen ja polun omiin muuttujiinsa, jotka sitten yhdistin os.path.join -komennolla, jolloin kansio taas luodaan. Tämän hetken ongelmien vuoksi selvitin, onko Nokian automaatio-ohjelmassa asennettuna pip virtualenv -paketti ja selvisi, että on. Ongelma ei siis ole siellä. Tällä hetkellä koetan saada os.system() -komennoilla luotua ympäristöä, mutta toistaiseksi en ole onnistunut.

25.2.2021

Windows taistelee vastaan. Luotuaani uudet keinot luoda kansioita, Windows eväsi oikeudet ja piti siitä lähtien avata ohjelmat järjestelmävalvoja -oikeuksilla. Tällä hetkellä työn alla on katsoa, että voiko oikeuksia antaa koodin kautta, sillä käyttäjälle voi olla ongelmallista, jos toiminnallisuuden vuoksi täytyy aina olla järjestelmävalvojan -oikeudet. Virtuaaliympäristön luominen 2.7.14 Pythonista 3.9:ään takkuilee edelleen. Olen koettanut useita erilaisia komentoja, mutta en ole onnistunut vielä siinä. Esimerkkeinä pyvenv ja virtualenv – p 'which Python3.9.' <path/to/new/virtualenv>.

Viikkoanalyysi:

Virtuaaliympäristön luominen ei vielääkään ole onnistunut, mutta sain tällä viikolla määriteltyä asetuksiin polun, jonne se lopulta sitten laitetaan. Välillä mielessäni käy, että jokin perustavanlaatuisen ongelma tässä virtuaaliympäristön luomissuunnitelmassa on. Jokaisen uuden askeleen kohdalla Windows tai käytettävät työkalut takkuilevat vastaan. On hyvinkin myös mahdollista, että oma kokemattomuuteni Python koodaamisessa tulee vastaan.

4.7 Viikko 7

2.3.2021

Päätin kokeilla luoda virtuaaliympäristön ilman sopivaa pakettia (3.9.2), mutta sekään ei onnistunut. Olettaus oli, että automaatio-ohjelma käyttää Python 2.7.14:ää, niin se loisi automaattisesti samalla paketilla virtuaaliympäristön. Tällä hetkellä luon kansion `if not os.path.exists` -komennolla ja pyrkimys olisi, että heti perään tulee virtuaaliympäristön luonti. Kokeilin `if os.path.exists` -komennolla, mutta se loi vain kansion toisen kansion sisään.

3.3.2021

Uudeksi ongelmaksi osoittautui Python 3.9:n asentaminen koneelle, sekä samaisen ohjelman käyttämä Python 2.7.14. Ratkaisuna jätämme Pythonin asennuksen käyttäjälle, kun kyseessä on Windows -ympäristö. Windows koneilla ajetaan siis automaatiotyökalun kautta Command prompt (`cmd`) -käskyjä, jotka lataavat, purkavat ja asentavat uusimman version `IM_analyser_tool`:sta ja suorittaa komennot `cmd`:ssä eikä itse työkalussa. Täten vältetään eri Python -versioiden aiheuttamat ristiriidat. Linux ympäristöissä tarvitaan virtuaaliympäristöä, jotta voidaan asentaa sopiva Python -versio, mutta täytyy myös muokata käytössä olevaa versiota. Näissä oletusarvoisesti käytössä on Python 3.6, kun tarvittaisiin 3.7 tai uudempi. Ongelmana tällä hetkellä on saada vaihdettua käytössä olevaa versiota virtuaaliympäristöön.

4.3.2021

Koetin selvittää, että miten saisin muokattua käytettävää Pythonia. Lähtökohtaisesti, kun tarkistan Pythonin version `"Python --version"` tai `"Python2 --version"` -komennolla tulee tulokseksi 2.7. `"Python3 --version"`:lla saadaan Python 3.6.10. Kun yritin asentaa Python 3.7:n, järjestelmä ilmoitti, että se on jo koneella. Koetin saada muutettua Pythonin-versiota käyttämällä `"update-alternatives"` -komentoja, mutta tuloksetta. Sain manuaalipuolelle 3.6 ja 3.7, mutta auto modella säilyy koko ajan 3.6.10, vaikka kuinka vaihdan prioriteetteja (4). Tämä tarkoittaa, että manuaalisesti voidaan käyttää versioita, mutta automaatiokäytössä en saa haluamaani Python 3.7:ää käyttöön. Seuraavaksi yritin tehdä saman virtuaaliympäristössä siltä varalta, että järjestelmä itsessään estää minua tekemästä näin suuria muutoksia. Päivitin myös `"apt-get update"` ja `"apt-get upgrade"`:lla, jotta varmasti on tuoreet paketit (5). Virtuaaliympäristön kauttakkin `"update-alternatives"` -komennot olivat vailla vaikutusta. Koetin myös tehdä symbolic linkin, mutta jostain syystä tuli vain viesti `"failed"`. Yllätyksekseni, kun poistuin virtuaaliympäristöstä ja asensin `"apt-get install Python3-venv"` -komennolla tuoreet virtuaaliympäristön paketit ja loin sen uudessa ympäris-

tössä. Ympäristöön olikin asennettuna Python 3.7.3, jossain yritysteni aikana. En kuitenkaan saanut toistettua tätä uudestaan puhtaalta pöydältä, joten en voi tarkalleen paikantaa, että mistä onnistuminen johtuu.

5.3.

Päätin hylätä virtuaaliympäristöt ja koettaa symbolic-linkkiä suoraan järjestelmätasolla ja yllättäen se onnistui. Ongelmana tähän asti oli, että koetin koko ajan kohdistaa symbolic-linkkiä /usr/bin/Python3 -osoitteeseen, koska se on oletusarvo. Python3 sen sijaan sijaiti syvemmillä tietokoneen sisällä ja selvitettyäni osoitteen sain yhdistettyä "Python3 -version":n taakse 3.7.3. Onnistuminen tuntuu hyvältä, mutta hieman harmittaa sillä virtuaaliympäristöä ei näyttäisi sittenkään tarvitsevan.

Viikkoanalyysi:

Automatisoitu Pythonin asentaminen joudutaan nyt unohtamaan. Se osoittautui liian vaikeaksi saada asennettua Python-skriptin mukana. Toisekseen alkuperäisessä Im_analyserin asennusohjeessa ohjeistetaan muutenkin asentamaan uusin Python itse. Linuxilla on jo uudempi Python oletusarvoisesti, mutta ei tarpeeksi uusi Im_analyseriä varten. Toistaiseksi uudemman version asentaminen virtuaaliympäristöön ei ole onnistunut. Tämän viikon tekemisissä selvisi myös, että tarvittava Python -versio löytyykin jo meidän Linux-koneiltamme, mutta se pitäisi saada vain käyttöön. Yritin pitkään manipuloida käytössä olevia Pythoneita oikeiksi, mutta tuloksetta. Onnella sain kuitenkin asennettua oikean Pythonin. Edelleenkään en voi olla varma, mistä se johtui mutta lopulta sain oikean Python-version käyttöön symbolic-linkillä. Aikaisemmin kohtaamani ongelmat tällä toiminnolla olikin vain huolimattomuusvirhe.

4.8 Viikko 8

15.3.2021

Erinäisten päivitysten myötä on hieman muuttunut automaatiotyökalun toiminta, joten koodi ei kulje enää kuten aikaisemmin. Nyt koetan saada skriptin ajamaan lähinnä command prompt -komentoja Windowsin puolella, mutta koodissa ei nyt saa enää määrätä kansiota mihin asennan IM_analyser toolin. Seuraavaksi otan selvää, mistä ongelmat johtuvat.

16.3.2021

Nyt on ollut sama ongelma pitkään päällä. Hieman edistystä, kun skriptin sisennykset olivat hajallaan, mutta en edelleenkään saa tiedostoja haluamaani paikkaan. Otin käyttöön urllib -kirjaston, jolla pyrin lataamaan GitLabista zip-tiedoston.

17.3.2021

Pientä koodin viilaamista ja sain taas luotua tiedostot haluamaani paikkaan. Zip-tiedoston lataaminen GitLabista puolestaan osoittautuu ongelmalliseksi. Olen jo jonkin aikaa koettanut eri komennoilla cmd:n kautta saada lataukset toimimaan, mutta tuloksetta (kokeilin curl ja wget eri variaatioineen) (6). Tämä ei selvästi onnistu, joten koetan tehdä sen suoraan koodin kautta käyttäen urllib- ja urllib2 -kirjastoa. Kollegani huomautti, että hakiessa tiedostoja meillä täytyy myös tunnistautua sivuille, joten loin käyttäjälle mahdollisuuden syöttää oman käyttäjätunnuksen sekä salasanan kirjautumista varten. Jonkin aikaa painittiin luullusti Windowsin oikeuksien kanssa, mutta syynä olikin tiedostonimestä puuttuva ".zip" -pääte. Sain ladattua tiedostosta pienen pätkän, mutta lataus keskeytyi sittemmin melkein heti. Epäilyksenä on edelleen Gitiin tunnistautuminen, kun meillä ei Windowsilla ole oletusarvoisesti Gitin sovelluksia. Seuraavaksi tulee tutkia, että kuinka saadaan Python 2:lta Git-tunnistautuminen tai mahdollisuuksia kiertää se.

19.3.2021

Tutkin Gitin tunnistautumista ja näyttäisi siltä, että kun lataan zip-tiedoston en tarvitse sen kummempia tunnistautumisia. Sen sijaan "urllib2.urlopen" ja write(foo.read()) ei saa kokonaista tiedostoa ladattua. Koetin saada ohitettua tämän laittamalla 'file://' URL:n, eteen, jotta operaattori tajuaisi heti, että kyseessä on file, mutta se kaatoi koko yhteyden. On mahdollista, että paketti pitää ladata eri kokoisina palasina, jotta saan lopulta kokonaisena ulos.

Viikkoanalyysi:

Lomaviikosta johtuneen tauon jälkeen oli normaalia enemmän ongelmia, mutta sain lopulta asiat vanhoille raiteilleen. Kokeilin kokonaan uusia kirjastoja Im_analyserin lataamisessa ja edistyin vain hieman saadessani pieniä palasia ladattua. Epäilen ongelmien johtuvat Git-sovelluksen puutteesta, mutta sen lataaminen erikseen monimutkaistaisi asioita liikaa. Myös tämän lisääminen käyttäjän harteille vaikuttaisi liialliselta. Tulevalla viikolla kokeilen, josko saisin ladattua tiedostot palasina.

4.9 Viikko 9

25.3.2021

Virheenetsintä sessioiden jälkeen ongelmaksi paljastuikin GitLabissa oleva tunnistautuminen. Kun yrittää ladata tiedostoa suoraan URL:sta GitLab ohjaa uudelle sivulle, jonne tulisi syöttää firman tunnukset sisäänkirjautumista varten. Tähän ei pelkkä raaka "username, password" -tunnistautuminen riitä, joten seuraavana pitää pohtia GitLabin oman kirjaston lataamisen mahdollisuuksia, sillä sieltä pitäisi luoda erillinen sessio, jossa hoidetaan tunnistautuminen.

26.3.2021

Toteutusta hieman uudelleen arvioidaan. Tällä hetkellä järkevimmäksi ratkaisuksi on laadittu suunnitelma, että Windows-koneilla jätetään käyttäjän vastuulle Im_analyserin asentaminen ja automaatiotyökalusta sitten kohdistetaan oikeaan ajettavaan kansioon, josta komentoriviä käyttäen koodissa ajetaan ohjelma. Linuxille puolestaan voidaan asettaa Git-komentoja, sillä niissä Git on oletusarvoisesti asennettuna, jolloin joka sessiolle voidaan automatisoidusti asentaa Im_analyser. Koodissa tämä toteutetaan `if os.name == "nt" → do something`.

Viikkoanalyysi:

Pidimme ohjaajani kanssa palaverin ja totesimme, että jätämme Im_analyserin lataamisen käyttäjän harteille, kun kyseessä on Windows -ympäristö. Tähänkin asti sitä on käytetty itse asennettuna ja sille on myös ohjeet valmiina. Käyttäjän tarvitsee siis automaatiotyökalusta vain hakea ajettava .py-tiedosto IM_analyserille ja muuten työn hoitaa skripti. Toteutus toimii yksinkertaisella `if` -lauseella, joka jo olemassa olevassa skriptissä on osittain käytössä.

4.10 Viikko 10

31.3.2021

Tänään pääasiassa Windows puolen koodin tekemistä. Kansion sijaan täytyy kohdistaa Im_analyserin py-tiedosto, josta ajetaan kyseinen softa. Tämän hetken ongelmana on saada oikea analysoitava tiedosto kiinni, jotta softaa ei ajettaisi tyhjillään. Kyseinen tiedosto löytyy kyllä jo olemassa olevan automaation skriptistä, mutta hieman tuottaa ongelmia löytää juuri oikeaa palasta. Myös lokitiedoston määrääminen olisi hyvä saada samaan kohteeseen, mihin muutkin tulevat automaatioissa, mutta toistaiseksi en ole löytänyt siihen keinoa.

1.4.2021

Sen sijaan, että luotaisiin oma skripti niin vaikuttaisi olevan järkevintä sulauttaa itse tekemäni jo valmiin kanssa. Täten ei esimerkiksi tarvitse määrittää erikseen lokitiedostojen polkuja ja edelleen kaikki data menee samaan paikkaan. Toiseksi käyttäjän kannalta on paljon helpompaa, kun uusiin toimintoihin pääsee käsiksi jo tutusta skriptistä ilman, että tarvitsisi alkaa opettelemaan kokonaan uusia käytänteitä.

Viikkoanalyysi:

Tällä viikolla tuli jälleen uusi "restart" projektille. Tähän asti olen luonut kokonaan omaa skriptiäni, mutta selvisikin, että tämä ei ole tarkoituksen mukaista. Meillä on jo toimivaa skriptiä ja tietenkin haluamme sulauttaa tämän mahdollisimman sujuvasti siihen. Vielä en osaa sanoa, kuinka tämä tapahtuu. Sujautanko oman pätkäni vain johonkin väliin vai onkohan tässä jokin muu keino, joka olisi järkevämpi.

4.11 Viikko 11

8.4.2021

Sain koulutusta automaatiotyökalun käytöstä ja tuloksena toteutus tulee olemaan, että luodaan erillinen luokka, jossa tarvittavat `Im_analyserin` metodit ja parametrit. Lokitiedostot haetaan koodin sisällä ja ne tulee erikseen osoittaa samaan paikkaan, kuin muutkin lokitiedostot.

9.4.2021

Loin uuteen luokkaani metodin, jolla haen analysoitavat lokitiedostot sekä alustavat funktiot tuleville ominaisuuksille. Erillisillä metodeilla hoidetaan esimerkiksi `Im_analyserin` hakeminen, sen ajaminen, sekä lokitiedostojen kohdistaminen oikeaan kansioon. Linuxille täytyy ainakin luoda metodi, jolla vaihdetaan ajettava Python.

Viikkoanalyysi:

Tämän viikon tuotokset olivat lähinnä uuden suunnittelua. Toteutukseni muuttuu kokonaisesta omasta skriptistä, erilliseksi luokaksi, jota sitten vain kutsutaan itse automaatiaskriptissä. Tämä on ohjaajani mukaan "Hyvä koodauskäytäntö" kun ominaisuuksia jaetaan moduuleihin. Käsittääkseni tämä on Python-koodauksen isoimpia valtteja, kun asioita voidaan helposti ottaa käyttöön ja pois käytöstä. Uusi ratkaisu vaikuttaa hyvin fiksulta, mutta hieman harmittaa, että taas pitää aloit-

taa alusta. Lienee tosin tavanomaista luovalle prosessille, että välillä pitää palata taas lähtöpisteeseen, että muistaa, mihin oltiin menossa.

4.12 Viikko 12

13.4.2021

Aloitin integroimaan omaa koodia jo olemassa olemaan automaation skripteihin. Olen yrittänyt saada käyttäjälle erilaiset toiminnot riippuen, että onko ajossa Linux vaiko Windows, mutta vielä en ole saanut toimimaan. Toistaiseksi itselleni on myös hämää, että mihin kohtaan laitan oman luokkani kutsun. Prosessi on jo valmiiksi yli 1000+ riviä pitkä, joten oikean kohdan haarukoiminen on vienyt aikansa.

14.4.2021

Olen edelleen koettanut saada `if os.name == "nt"` -komennolla asetukset toimimaan siten, että saan haluamani asetukset näkyville molemmille käyttöjärjestelmille. Päätin aluksi keskittyä, että saan Windowsin toimimaan ja sitten lisään Linux puolen `Else` -lausekkeelle

Viikkoanalyysi:

Omien skriptieni integroiminen tuotti paljon ongelmia, mutta kyllä se eteni. Omat koodini on pääosin sijoitettu omaan luokkaansa ja jo olemassa olevaan skriptiin olen melkein saanut asetukset valmiiksi. Asetuksille on pitänyt myös luoda selitykset koodiin, jotta koodin lukija sekä automaatiotyökalun käyttäjä tietävät, mistä on kyse.

4.13 Viikko 13

20.4.2021

Asetukset automaatiotyökalussa näyttävät oikeilta Windowsin puolella. Pystyn valitsemaan, että ajetaanko `Im_analyserin` skriptejä. Valittaessa "Kyllä" pyytää työkalua osoittamaan `im_analyser.py` tiedoston sijainnin niin kuin on tarkoituskin. Huomenna pääsen kokeilemaan skriptiä toiminnassa.

21.4.2021

Luomani luokka ei näytä toimivan vielä. Hakiessa luokan tuomia arvoja automaation käyttöön, jokin parametri palautuu tyhjänä. Asentelin Im_analyserin uudelleen ja päivitin Pythonin, mutta ei toistaiseksi en ole saanut ajoa tapahtumaan koodin sisällä. Tällä hetkellä koetan ajaa subprocess -kirjaston funktioilla cmd-komentoja, mutta on mahdollista, että joudun valitsemaan jonkun toisen.

23.4.2021

Skriptille piti vaihtaa command promptin:n kautta ajoon "Python3" tilalle, "Python" sillä "Python3":n taakse ei ollut tallentunut mitään. Analysoitavaan tiedoston hakeminen on edelleen ongelmallista. Metodi, jolla haen lokitiedostoja palauttaa tällä hetkellä tyhjää. Käytössä on check_output -komento, jolle syötän "Python", im_analyser.py -tiedoston sijainnin, sekä analysoitavan tiedoston. Koetan löytää alkuperäisistä skripteistä vihjeitä, kuinka saan lokitiedostot kiinni, mutta toistaiseksi en ole onnistunut.

Viikkoanalyysi:

Sain vihdoon koodini integroitua ja ensimmäiset koeajokin putkeen. Windows-puolen asetukset näkyvät juuri niin kuin pitääkin, mutta koodi ei vielääkään toimi. Jokin parametri ei saa arvoa ja palautuu tyhjällä arvolla. Olen yrittänyt sijoittaa Print-komentoja sinne tänne, jotta paikantaisin ongelmakohdan, mutta tuloksetta. Jotain en ole nyt huomannut. Otin tuoreeltaan käyttöön subprocess-kirjaston komentoja os-kirjaston sijaan, sillä jo olemassa olevat skriptit käyttävät samaa. En ole kokenut käyttäjä tällä kirjastolla, mutta saa nyt nähdä, toimiiko se minun käyttötarkoituksiini. Tällä hetkellä ei.

4.14 Viikko 14

26.4.2021

Sain vihdoon lokitiedostojen haun toimimaan ja ensimmäinen onnistunut ajo Windows -ympäristössä tehty. Automaatio-ohjelmalle piti laittaa lokien keruu laajemmaksi, jotta minun skripti löytää tarvitsevansa tiedoston. Seuraavaksi pitää alkaa tekemään muokkauksia skriptiin, jotta se sopeutuu myös Linux ympäristöön.

29.4.2021

Lisäsin Windows -toteutuksen koodiin pari riviä, joilla estän skriptiä kaatumasta, jos tarvittavia lokitiedostoja ei löydy. Tämä kuitenkin on käyttäjän näkökulmasta hyvin tavanomainen tilanne, joka tapahtuu. Linux toteutukselle loin metodin, joka luo aikaisemmin mainitun symbolisen linkin "Python3":lle, jotta se käyttäisi Python3.7.4:sta. Työstin myös toteutusta, jotta automaatio-ohjelman valikko sopeutuisi käyttöjärjestelmään mukaan, mutta siinä on vielä hiomista. GitLabista kloonauksen vuoksi pitää käyttäjällä syöttää tunnuksensa ohjelmaan, jotta automaatio voi pyöriä, mutta niille tarpeellisten palkkien luomisessa on ongelmia.

Viikkoanalyysi:

Ensimmäinen onnistunut koeajo on takana. Ongelmana olikin testaaja itse eli minä. En ollut asetuksista laittanut täyttä lokien keruuta, jolloin myös tiedosto, jota yritän analysoida ei ole olemassa. Onnistumisesta huolimatta sitä piti silti vielä hioa. Asetin itseni tulevan käyttäjän kenkiin ja hoksasin, että skriptini tuottamia lokitiedostoja ei välttämättä aina tarvita. Skripti nimittäin kaatui, jos luomaani toimintoa ei käytetä. Nyt koko ominaisuus on periaatteessa palautuskelvollinen Windowsille, mutta Linux ratkaisu puuttuu vielä. Sen ei tosin pitäisi olla mahdottoman vaikeaa, sillä iso osa prosessista on siellä samaa.

4.15 Viikko 15

4.5.2021

Päivän ongelmana on ollut Linux puolen toteutusten asetusten luominen. Jostain syystä if os.name != "nt" palauttaa vain erroria automaatiotyökalussa. Ongelmaan kuitenkin löytyi vastaus. Yksinäinen ":" koodissa piti korvata "=":lla ja ongelmat kaikkosi. Löysin myös virheitä Windowsin puolen toteutuksesta. Aikaisemmin, jos ei valinnut luomaani uutta ominaisuutta käyttöön, skripti kaatui loppupäässä. Tämä johtui siitä, että luomastani luokasta odotettiin arvoa, jota ei koskaan anneta, koska ominaisuus ei ollut käytössä.

5.5.2021

Tein GitLabista kloonauksista varten metodin. Pitää vielä testata toimiiko pelkät perättäiset subprocess.output -komennot komentoriviltä GitLabiin kirjautumiseen. command_im_analyser -metodille todennäköisesti täytyy tehdä if-lauseke, joka erottaa Linux- ja Windows ympäristöt, sillä im_analyserpath tulee hieman eri keinoin käytettäväksi.

7.5.2021

Tänään pääsin kokeilemaan luomaani skriptiä Linux-ympäristössä, mutta ensimmäiset kaksi ympäristöä, jotka sain haltuuni eivät jostain syystä antaneet ajaa automaatiotyökalua, sillä lisenssi oli kyseisiltä ympäristöiltä vanhentunut. Myöhemmin sain sopivan ympäristön käyttöni ja selvisi, että skriptini kaataa osia alkuperäisestä koodista, johon en ole koskenut. Tämä siis ainoastaan Linux-ympäristössä.

Viikkoanalyysi:

Tämän viikon ongelmina ovat olleet Gitistä kloonaminen sekä toimivan ympäristön löytäminen. Gitistä lataaminen ei tunnu onnistuvan ilman, että sinne kirjautuu sisään, mutta en ole vielä löytänyt keinoa tehdä sitä skriptin kautta. Toiseksi ongelmaksi on osoittautunut toimivan Linux-testiympäristön löytäminen. Joistakin on puuttunut lisenssit ja toiset ovat olleet kiireisiä korkeamman prioriteetin hommissa. Sain kuitenkin yhden testattavaksi ja olen hieman edennyt GiLab-tutkimuksieni kanssa. Täytyy sanoa, että tämä Linux puolen toteutus ei ollutkaan niin helppoa, kuin luulin sen edellisten onnistumisieni pohjalta olevan.

4.16 Viikko 16

10.5.2021

Aikaisempi ongelma taisi olla enemmän ympäristön, kuin koodin vika. Sain käyttöni ympäristön, jossa pääsen koodiini asti, mutta luokan uusien muuttujien, käyttäjätunnuksen ja salasanan tuominen automaatioon on tuottanut ongelmia. Alkuperäinen Windowsin koodi käyttää jo ennen luokan kutsumista `im_analyserpath`, mutta Linux ratkaisussa se tulee vasta myöhemmin käyttöön. Laitoin sen kiinni muuttujaan ja `if`-lausekkeella määritän, että missä kohtaa se tulee käyttöön.

11.5.2021

Yritin koko päivän saada GitLabista kloonamista onnistumaan. Aluksi pyörittelin `subprocess.check_output` -komentoa, mutta ei onnistunut. Komentorivin kautta ladatessa ensin pitää syöttää osoite mistä haluaa kloonata ja sitten erillisesti kysytään tunnukset ja salasanat. Edellä mainitulla komennolla kaikki syötetään yhteen instanssiin, joten se ei aja asiaani. Seuraavaksi kokeilin `subprocess.call` -komennolla, mutta sen kanssa olen saanut aikaiseksi vain "Syntax error" viestin. Linux ympäristöillä on lähtökohtaisesti Git asennettuna, joten sen ei pitäisi tuottaa ongelmia. Ehkä jokin vielä vanhempi osa koodistani on viallinen.

12.5.2021

Sain skriptin taas toimimaan ja varmistin, että `change_Python()` -metodi toimii. Skripti ei enää kaadu, mutta en saa kloonausta GitLabista vielä onnistumaan. Minulla on username ja password välitettynä funktioon `format()` -metodilla, mutta kloonaus ei mene läpi. `subprocess_call` ja `check_output` ovat molemmat olleet kokeilussa. Koetan vielä ensiviikolla uudestaan näitten paris-
sa, mutta jos ei onnistu joudun todennäköisesti vaihtamaan lähestymistapaani

Viikkoanalyysi:

Linuxin toteutuksen aloittaminen on tuntunut hieman toivottomalta. Käyttämäni komennot eivät toimi ja yksinkertaiselta vaikuttavien ominaisuuksien lisääminen tuo vain mutkia matkaan. GitLabista kloonaukselle on kuitenkin saatu mukaan käyttäjätunnus ja salasana. Palaset ovat valmiina. Pitäisi vain saada ne toimimaan yhdessä. Enköhän ensiviikolla saa tämän osion koodista toimimaan, jotta voin edetä seuraavaan vaiheeseen.

4.17 Viikko 17

18.5.2021

GitLab:sta lataaminen toimii vihdoin. Ongelmana oli latauslinkin URL, jossa yksittäinen kaksoispiste piti vaihtaa kauttaviivaksi. Seuraavana työn alla on saada sopivat pip -paketit ladattua, jotta `Im_analyserin` saa ajettua. Laitoin myös ehdon `Im_analyserin` hakuun Gitlabista, jolloin sitä ei ladata uudestaan, jos asennus on jo olemassa koneella.

19.5.2021

Loin Linux puolellekin polun `Im_analyser` -sovellukseen. Pip-paketit eivät tulekaan automaattisesti symbolisen linkin myötä, joten olen tutkinut erilaisia vaihtoehtoja saada paketit käyttöön. Pip:stä löytyy myös "pip" sekä "pip3" aivan, kuten Pythonistakin "Python" sekä "Python3". Molemmat ovat kuitenkin linkattuna Python 3.6:lle, kun itse käytän 3.7, joten paketti eivät tule komentojeni mukana. Koetin poistaa linkityksen "Python3":een, jotta saisin sen takaisin Python 3.6:ksi, mutta komennolla "symlink" onnistuin harmikseni poistamaan Python 3.7:n binääritiedoston. Loppupäivä meni tätä selvitellessä. Väliaikaiseksi ratkaisuksi saatiin, että kopioitiin toiselta koneelta samanlainen tiedosto.

20.5.2021

Väliaikainen ratkaisu muuttui pysyväksi ja pääsin takaisin töihin. Jokainen oma Python -versionsa tarvitsee omat pip-pakettinsa, mutta toistaiseksi en ole onnistunut lataamaan mitään paketteja apt-get-komennoilla. Mahdollisesti pitää näihinkin saada proxy server, niin kuin Windowsilla. Yritin jo laittaa sen suoraan cmd-komentoihin, mutta ei ole toistaiseksi toiminut.

21.5.2021

Proxy server ei ollut ongelmana, vaan "apt install xxx" vaati apt/source.list:n päivittämisen. Se onnistui sed, eli Stream Editorin komennoilla, joilla lisäsin osoitteen, josta uudet paketit sai. Kuitenkin saatuaani pip-paketin ladattua "Python3":lle en saanut ajettua itse im_analysatoria. Se ei enää kaadu ajettaessa, mutta on mahdollista, että symbolic linkistä huolimatta "Python3":lle ladatut paketit olivat Python 3.6:lle 3.7:n sijaan. Tämä vaatii vielä tutkimista ja testaamista, mutta alkaa näyttää hyvältä. Kunhan apt ja pip -pakettien asentamisen saa liitettyä automaatioon sekä im_analyserin ajon viimeisteltyä, alkaa koko projekti olla loppuillaan.

Viikkoanalyysi:

Pienellä muutoksella kirjoitusasuun sain korjattua kloonaamiseen liittyvät. Sain hieman optimoitua jo olemassa olevia toimintoja ja selviteltyä pip-pakettien käyttöä. Linuxilla selvästi täytyy olla varovainen puuttuessa Pythonin binääritiedostoihin, sillä niillä on koko käyttöjärjestelmän kannalta tärkeitä riippuvuuksia. Symbolic linkin kautta toimiminen on hieman edennyt, mutta alkaa hiljalleen vaikuttaa siltä, että siitä ei vastauksia löydy. Ensi viikolla kuitenkin perehdyn asiaan lisää, jotta voin olla varma, mitä teen seuraavaksi.

4.18 Viikko 18

25.5.2021

Tilanne vaatisi uuden Pythonin asentamista koneille. Ongelmana on, että koneilla ei ole täysiiä lupia käydä verkossa, joten esimerkiksi "www.Python.org" ei ole saavutettavissa. Olen koettanut asentaa Proxy-serveriä sekä komentoriville "--proxy":llä sekä suoraan wgetrc:hen muokannut aktiivisesti käytettävän proxy serverin, mutta ilmoituksena tulee, että "network unreachable". Todennäköisimmin minun pitää kysellä lupia verkosta, jotta saan tarvitsemani paketit. Muistaakseni jotkut työntekijät ovat asentaneet manuaalisesti Linux -koneille uudempaa Pythonia, joten kyselen huomenna heiltä neuvoja.

26.5.2021

Aikaisemmin onnistunut Pythonin asennus Linux ympäristöillä oli mitä luultavimmin Pythonin virtualenv -moduulilla. Edelleen on ongelmia saada ladattua mitään verkon välityksellä. Omituista kyllä Gitistä kloonaminen onnistuu, mutta muille sivuille pääsy ei. Linuxilla on oletusarvoisesti Git asennettuna, mutta ei sieltäkään ilman verkkoyhteyksiä mitään saa. Ehkä tässä pitää vielä säätää proxyjen kanssa. http- ja https-protokollilla en ole vielä saanut onnistumisia, mutta ehkä ftp:llä onnistaa.

27.5.2021

Verkko-ongelmat ratkesivat lopulta proxy servereillä. wget-komennon eteen piti laittaa export "käyttämäni proxy"; wget "haluamani osoite" (7). Saatuani ladattua haluamani Python version Linux-ympäristöön, vielä purkaa, valmistella ja asentaa paketti. Tässä oli hetken jumi, kun en aluksi huomannut siirtää purkamisen jälkeen saamaani tiedostoon (os.chdir -komennolla). Jokaisen testauskerran ensimmäinen kierros tulee olemaan huomattavasti pidempi, sillä valmisteluissa ja asentamisessa menee yhteensä parhaillaan 10 minuuttia, mutta sen jälkeen automaatio toimii vauhdikkaasti. Sain tänään ensimmäisen onnistuneen testikierroksen Linux-ympäristössä. Seuraavaksi pitää vielä testata uudelleen Windows-ympäristössä ja tarkistaa, että mikään ei ole mennyt rikki. Toisekseen Linuxin kanssa pitää vielä kokeilla aivan paljaassa ympäristössä, jossa ei ole tehty aikaisemmin mitään. Näiden jälkeen ei ole jäljellä, kuin pientä hiomista ja raportointi.

31.5.2021

Linux-puolen toteutus on testattu nyt monen erin skenaarion kautta ja näyttäisi toimivan. Windowsin testaamisessa huomasin, että Linuxille luodut im_username sekä im_password tarvitsevat myös jonkin tyhjän alustuksen Windowsin puolelle, jotta luokka saadaan tuotua. Koetin luoda if-else-rakenteen sisään omat, mutta luokka välittää joka tapauksessa kuusi objekta, joten tämä ratkaisu ei käy. Seuraavaksi kokeilen, jos laitan ne vain tyhjäksi.

Viikkoanalyysi:

Suuria askelia tällä viikolla. Sain haettua tarvitsemani paketit wget-komennolla. Aikaisemmin kohtaamani ongelmat alkavat tuottaa tulosta, sillä enää en jumita montaa viikkoa yksittäisen sanan tai merkin puuttuessa. Linux-puoli on nyt ensimmäistä kertaa toimiva. Enää täytyy varmistaa, että Windowsin puolella on myös sama tilanne, sillä viime ajoista siellä on jo kulunut aikaa ja koodikin on hieman muuttunut, kuten aiemmin mainitut uudet muuttujat. Viimeaikaiset ongelmatkin ovat luonteeltaan olleet enemmän hiomista kuin puuttuva palanen. Loppu hämmöttää.

4.19 Viikko 19

1.6.2021

Ideani toimi eli annoin Windows -ympäristössä tarpeettomille parametreille tyhjän string:n arvon ja koodi toimii taas. Jouduin myös lisäämään Windowsin käyttöpuolen ohjeisiin xlsxwriter pip-paketin, joka on projektin teon aikana tullut pakolliseksi IM_analyserin käyttöä varten. Linuxilla piti tehdä pieniä muokkauksia. Aikaisemmassa koodissa käytin os.chdir -komentoa, jotta pystyin asentamaan uudemman Pythonin. Lisäsin komennot, joilla pääsen takaisin Linuxin oletuskansioon. Putsasin myös paljon vanhentunutta sekä kommentoitua koodia, jotta kokonaisuus näyttää siistimmältä. Linux toimii taas ja jälleen kokeilen, että toimiiko muokattu koodi Windowsissa.

4.6.2021

Olen viilaillut toimintaa molemmilla Linuxin ja Windowsin puolella ja nyt molemmat ovat toimivia. Tein myös käyttöohjeet luomilleni toiminnoille, jotta työkaverinkin voivat ottaa käyttöön uudet muutokset. Käytimme iltapäivän ohjaajani kanssa yrittäessä välittää tuotoksiani Gittiin. Tämä oli hieman työläämpi prosessi, kuin olisi pitänyt sillä Git-linkitys olisi pitänyt tehdä heti alkuun. Saimme kuitenkin "commitattua" ja pusketta koodini, mutta jotain vikaa siellä vielä on. Seuraavalla viikolla korjaamme ongelmat.

Viikkoanalyysi:

Ohjelmani integraatio olemassa olevaan automaatioon näyttäisi olevan valmis. On ollut hieman mutkia matkassa, mutta pian saadaan nähdä, kuinka ohjelmani toimii todellisessa testaustoiminnassa. Kiire on selvästi tuottanut ongelmia toteutukseen. Ohjaajani sekä omat kiireet lisäsivät toteutukseen lisäpäiviä, sillä olisimme voineet hoitaa Gitin toimintojen opetuksen heti alkupäässä, jolloin toiminta olisi nyt ollut paljon sulavampaa.

4.20 Viikko 20

8.6.2021

Tänään luomani sovellus ja sen integroiminen julkaistiin käyttöön. Toistaiseksi on toiminut eikä ole ainakaan tullut isompia ongelmia. Kehitysehdotuksia kuitenkin tuli. Tällä hetkellä pilviauto-

maatiota, joita ajetaan Linux -koneilla, on työlästä muokata. Ensin täytyy käydä Linux -koneella tekemässä tarvittavat muutokset automaatioon ja vasta sen jälkeen voidaan se siirtää oikeaan osoitteeseen. Myös Windowsin puolelle tarvitaan helpompi väylä, jotta hakemiini lokitiedostoihin päästään käsiksi.

10.6.2021

Kaiken touhun keskeytti uuden kriittisemmän bugin löytäminen. Linuxilla ajaessa käyttäjän syötämä salasana tulee näkyviin, kun automaatio listaa käytettävät parametrit. Mietin, että kuinka tämän voi kiertää, mutta löytyikin parempi keino. Hankimme yhteisen käyttäjätunnuksen koko tiimille, jolloin kenenkään ei tarvitse syöttää tunnuksia enää, jos ne toimivat.

11.6.2021

Yhteiset tunnukset ratkaisivat ongelman. Samalla myös sain optimoitua Linux -puolen koodia siten, että ei tarvitse syöttää mitään tunnuksia, sillä ne ovat jo koodissa. Tämä helpottaa testaajan elämää huomattavasti sekä parantaa tietoturvaa.

Viikkoanalyysi:

Paketti on nyt saatu julkaistua käyttöön ja kehitysehdotuksia sekä ohjelmavirheitä on jo korjattu. Minua kyllä varoitettiin heti alkuun, että yksikään tämän kokoinen uusi toiminto ei toimi heti kättelyssä. Työni on siis virallisesti valmis ja opinnäytetyöstä puuttuu enää raportointi. Hieman pitkittyi tekeminen, sillä ohjelman luonti oli arvioitua haastavampaa.

5 POHDINTA

Päätin yhdessä opettajani kanssa tehdä opinnäytetyöni päiväkirjamuodossa. Tärkeimpänä syynä tähän oli täyspäiväinen työskentely. Työt itsessäänkin haastavat ja vaativat jatkuvaa ongelmanratkaisua ja itsensä kehittämistä, joten opinnäytetyöstä oli hyvä saada mahdollisimman sujuva työarkeen. Normaali toteutus olisi varmasti vaikuttanut työni laatuun, mikä ei olisi ilahduttanut työnantajaani. Korkeimman prioriteetin tavoitteiksi valittiin itse sovelluksen luominen, mikä helpotaisi tiimin työntekoa, sekä sovelluskehityksen taitojen kehittäminen, jotta voin myöhemmin auttaa koodaamista vaativien muutosten kanssa.

Päiväkirjatoteutuksen hankalin osuus oli saada teksti pysymään yhtenäisenä. Kiireisen työn vuoksi jouduin usein pitämään usean päivän taukoja itse sovelluksen kehittämisestä, jolloin päiväkirja alkoi vaikuttaa riekaleiselta. Toinen haastava puoli oli tuottaa tarpeeksi yksityiskohtaista tekstiä päivän tuotoksista. Pään sisäiset prosessit eivät aina kääntyneet paperille, kuin olisi toivonut. Välillä myös ongelmakohdat olivat niin haastavia, etten saanut kirjoitettua ollenkaan uusia havaintoja, kun sovelluksen kehitys ei edennyt. Positiivisena puolena tässä toteutuksessa on koko prosessin kirjaaminen. Ongelmatilanteita ja umpikujia tuli vastaan paljon ja joskus ne kestivät viikkoja. Kaikesta tästä huolimatta ne haasteet voitettiin usein poiketen kokonaan alkuperäisestä suunnitelmasta.

Sovelluksen tekeminen oli pääasiassa itsenäistä työtä. Minulla oli työpaikalta valittu ohjaaja, mutta hänkään ei pystynyt antamaan minulle rajattomasti aikaa. Myös omat työkiireeni rajoittivat usein työn etenemistä, mistä seurasi tavoitteiden viivästyminen. Tämä on täysin sattumaa, sillä on lähes mahdotonta ennustaa, koska työstäni tulee kiireistä. Isomman ongelman löytyessä niin testaajat kuin sovelluksen kehittäjätkin tekevät pitkää päivää. Avun pyytämiseksi ei kuitenkaan ollut kynnyksiä. Pystyin aivan vapaasti kysymään myös oman tiimini ulkopuolisilta työntekijöiltä ja auttoivat, jos pystyivät. Tämä oli ongelmanratkaisun kannalta hyvin tervetullutta, sillä siitä sain paljon uutta näkökulmaa.

Opinnäytetyön teon aikana opin paljon uutta Pythonista. Opin hyödyntämään Pythonin modulaarisuuden vahvuuksia ja luomaan siistiä sekä toimivaa koodia itsenäisesti. Sain myös arvokasta kokemusta koodikielen eri versioiden ristiin käyttämisestä niin Linux- kuin Windows-ympäristöillä. Ylipäätään Linux-ympäristöjen ymmärrys ja ajaminen on nykyään paljon helpompaa kuin aluksi.

Myös aikaisemmin työssä käyttämäni sovellukset ovat nyt tutumpia, kun jouduin sukeltamaan ominaisuuksien luonnin syövereihin.

Sovelluksen kehittäminen on eksponentiaalisesti kasvattanut kykyjäni ohjelmoida. Ensimmäistä kertaa urallani loin jotain, mistä on aidosti hyötyä kilpailullisessa yrityksessä sekä mikä helpottaa useamman työntekoa. Hieman pakon kautta jouduin kehittämään suunnittelutaitojani. Useaan otteeseen piti nosta projekti pöydälle ja tarkastella kokonaisvaltaisesti, mikä toimii ja mitä pitää kehittää.

LÄHTEET

1. Nokia Oyj. 2021. Nokia Oyj. Saatavissa: <https://www.nokia.com/contact-us/>. Hakupäivä 22.11.2021.
2. Sajip, Vinay 2021. Creation of virtual environments. Python.org Saatavissa: <https://docs.Python.org/3/library/venv.html>. Hakupäivä 29.10.2021.
3. Sinha, Pravin Kumar 2017. Accessing Files Using Python. Developer.com. Saatavissa: <https://www.developer.com/design/accessing-files-using-Python/>. Hakupäivä 29.10.2021.
4. Shovon, Shahriar 2021. How to Use update-alternatives Command on Ubuntu. Linux-hint. Saatavilla: https://linuxhint.com/update_alternatives_ubuntu/. Hakupäivä: 29.10.2021.
5. StackExchange – Updating Python on Ubuntu system. 2011. StackExchange. Saatavilla: <https://superuser.com/questions/241865/updating-Python-on-ubuntu-system>. Hakupäivä: 29.10.2021.
6. Pocentek Gauvain, Mäenpää Mika. 2013. Getting started with the API. Python-gitlab. Saatavilla: <https://Python-gitlab.readthedocs.io/en/stable/api-usage.html>. Hakupäivä: 29.10.2021.
7. Stack Overflow – How to do a wget call via proxy with Python. 2012. Stack Overflow. Saatavilla: <https://stackoverflow.com/questions/6980575/how-to-do-a-wget-call-via-proxy-with-Python>. Hakupäivä: 29.10.2021.