Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Ville Sutinen

PolicyCenterissä tapahtuvan yhtiöiden fuusioitumisen tukitoimenpiteet

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

3.5.2021

Tekijä Otsikko Sivumäärä Aika	Ville Sutinen PolicyCenterissä tapahtuvan yhtiöiden fuusioitumisen tuki toimenpiteet 38 3.5.2021
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Ohjelmistotuotanto
Ohjaajat	Lehtori, Ilpo Kuivanen Johtaja, Laura Hautsalo

Insinöörityössä käytiin läpi vakuutuksiin erikoistuneen Guidewire Inc.:n tuottamia InsuranceSuite-ohjelmistoperheen sovelluksia. Näistä sovelluksista käytiin läpi maksu- ja laskutustoimintaan erikoistunututta BillingCenteriä, korvauksiin erikoistunutta ClaimCenteriä ja vakuutusten hallintaan erikoistunutta PolicyCenteriä. Itse projekti liittyi PolicyCenterissä tehtäviin muutoksiin, ja sen vuoksi PolicyCenterin toiminnallisuutta käytiin läpi tarkemmin. PolicyCenterissä käytiin läpi esimerkiksi sen käyttämiä ui-komponentteja, pcf-tiedostoja, joihin nämä ui-komponentit pohjautuvat, datamallia ja yleisesti PolicyCenterin käyttämien esimerkkien avuilla.

Projektissa tuotettiin kahden alueellisen vakuutusyhtiön fuusioitumiseen liittyviä toimenpiteitä. Nämä toimenpiteet eivät mahdollistaneet muutoksia vaan olivat fuusioitumista tukevia toimia. Projektissa tehtiin vanhalle Poistuvalle alueyhtiölle muutoksia. Poistuvan yhtiön suhteen tehtiin muutoksia käyttöliittymässä, jotta Poistuva yhtiö ei enää näkyisi käyttöliittymässä. Myös uusien vakuutusten luonti estettiin Poistuvalle yhtiölle. Uudelle Fuusioidulle yhtiölle tehtiin tarvittavat muutokset käyttöliittymään, jotta Fuusioidun yhtiön nimi näkyisi jatkossa oikein tarvittavissa aktiviteettijonoissa ja käyttöliittymän komponenteissa. Sen lisäksi käytiin läpi projektissa vastaan tulleita ongelmia ja sitä, kuinka niitä lähdettiin ratkaisemaan. Lopuksi kerrotaan, mitä parannettavaa projektissa olisi voinut olla ja mitä projektia tehdessä opittiin. Projekti on vielä keskeneräinen, sillä kaikkia vastaan tulleita ongelmia ei ole kyetty ratkaisemaan vielä.

Avainsanat	Guidewire, PolicyCenter, Gosu

Author Title Number of Pages Date	Ville Sutinen Related Measures for fusion of two regional companies in PolicyCenter 38 pages 3.5.2021
Degree	Bachelor of Engineering
Degree Programme	Information and Communictions Technology
Professional Major	Software Development
Instructors	Ilpo Kuivanen, Senior Lecturer Laura Hautsalo, Manager

The topic of the final year project covered an application from the InsuranceSuite software family produced by Guidewire Inc, a company specialized in insurance software. Of the InsuranceSuite applications, BillingCenter, specialized in payment and billing, ClaimCenter,specialized in claims and PolicyCenter,specializes in insurances were reviewed. The project itself was related to the changes made in PolicyCenter so PolicyCenter is also introduced in more detail. From PolicyCenter UI-components, the PCF-files that the UI-components are based on, PolicyCenter datamodel and examples used in general in PolicyCenter are reviewed.

The project produced measures related to and supporting the merger of two regional insurance companies. The project made changes to the old Removed Company. The changes included e.g. hiding the Removed Company from the UI and making sure that there is no way to create new policies for the Removed Company. For the new Merged Company the changes included modifications in the UI so that the new name shows correctly in activity queues and other UI-components.

In addition, also the problems and errors that came up during the development are discussed The paper also discusses what could have been done better or differently and what was learned within the project.

Keywords	Guidewire, PolicyCenter, Gosu

Sisällys

Lyhenteet

PCF	Page Configuration File. Guidewire-järjestelmässä käytettävä käyttöliittymäkomponenttitiedosto.
XML	Extennsible Markup Language. Merkintäkielien standardi, jonka mukaan määritetään tietojen merkintämuodon looginen rakenne.
UI	User inteface eli käyttöliittymä.
AUW	Underwriter authority eli underwriterin valtuudet.
TTX	Typelist extension file. Guidewire-järjestelmissä käytettävän typelistin ominaisuuksien lisäämiseen käytettävä tiedosto.
TTI	Typelist declaration file. Guidewire-järjestelmissä käytetettävien typelistien määrittelytiedosto.
UW	Underwriter eli vakuutuksenantaja.
UAT	User Acceptance Testing eli asiakkaan toteuttama hyväksymistestaus.
FUNC	Functional environment. Perustoiminnallisuuksien testaamiseen soveltuva testiympäristö.
SIT	System integration testing eli järjestelmien välisen integraatioiden testiympäristö.
ADL	Admin data loader eli admin tiedot sisältävä tiedosto joka ladataan sovelluksen käynnistyessä.

1 Johdanto

Työssä on tarkoituksena käydä läpi Guidewiren luomaa Insuransesuite-yritysohjelmistoa. Kyseinen yritysohjelmistoperhe koostuu lukuisista sovelluksista. Tässä insinöörityössä käydään läpi lyhyesti kyseisestä ohjelmistoperheestä seuraavat järjestelmät:

- BillingCenter
- ClaimCenter
- ContactManager
- PolicyCenter.

PolicyCenterin toimintaa avataan tarkemmin, sillä asiakkaalle tehty projekti on tehty nimenomaan tälle kyseiselle ohjelmistolle. Sen lisäksi olen suorittanut kyseisen ohjelmiston konfigurointisertifikaatin ja toivon, että tuosta osuudesta on hyötyä jatkossa myös muille. Tämän lisäksi kerrotaan lyhyesti vakuutusalan perusasioista eli riskeistä ja vakuutuksista.

Lopussa kerron itse asiakasprojektista. Projektissa toteutettiin kahden alueyhtiön fuusiota tukevia toimenpiteitä. Itse työ ei ole mahdollista näiden yhtiöiden fuusiota, sillä se on jo toteutettu aiemmin. Projekti sen sijaan tukee tätä aiempaa toteutusta muun muassa käyttöliittymässä näkyvillä muutoksilla.

2 Vakuutuksen perusteet

2.1 Vakuutuksen perusteet

Insinöörityö keskittyy hyvin pitkälti vakuutusalan käyttämään ohjelmistoon ja siihen tehtäviin muutoksiin. Koska vakuutusala ei välttämättä ole jokaiselle kovin tuttu aiheena, on hyvä käydä alkuun hieman läpi vakuutuksien tärkeimpiä termejä eli riskiä ja vakuutusta. Näiden termien avaaminen auttaa hahmottamaan paremmin vakuutusalaa, miksi kohteita vakuutetaan ja sen myötä paremmin vakuutusalalla tarvittavia ohjelmistoja.

2.2 Mikä on riski?

Ihmisillä ja yrityksillä on elämässään monia epävarmuustekijöitä. Ihmistä ja tämän perhettä voivat uhata sairaudet, huoltajan kuolema, murto tai vaikka tulipalo. Liikeyrityksellä tuotantolinjastolla yrityksen toiminnalle tärkeä laitteisto voi odottamattomasti rikkoutua tai tehtaassa voi syttyä tulipalo, jonka myötä koko tehdas palaa ja toiminta keskeytyy. Tällaisia epävarmuustekijöitä kutsutaan riskeiksi eli epävarmuustekijöiksi, jotka voivat aiheuttaa vahingonvaaran.

Riskeillä voidaan kuitenkin vakuutusalan sisäisessä kielessä tarkoittaa muutakin. Voidaan puhua itse vakuutettavasta kohteesta. Esimerkiksi taloista puhuessa voidaan puhua puu- ja kiviriskeistä. [1.].

On myös erilaisia riskejä kuten staattiset ja dynaamiset riskit. Staattiset riskit ovat usein muuttumattomia vahingonvaaroja. Staattiset riskit ovat usein kelpoja riskejä vakuuttaa. Tällaisia voisivat olla esimerkiksi tulipalo tai murtovarkaus. Toinen mihin, riskit voi luokitella ovat dynaamiset riskit. Dynaamisissa riskeissä on herkästi muuttuvia riskejä. Tällainen voi olla äkillinen hintojen muuttuminen markkinoilla ja siitä koituva vaara aiheutuville tappioille.

Sattumanvaraisuus on yksi riskin ominaisuuksista. Tällä tarkoitetaan sitä, ettei riskien toteutumista tai toteutumattomuutta voi täysin ennustaa. Kerryttämällä tarpeeksi tietoa voidaan kuitenkin sattumanvaraisuudestakin löytää säännönmukaisuuksia. Vuosien ajalla, jos seurataan vaikka puutalojen ja niiden piippujen nuohoamisen yhteyttä, löytyisi todennäköisesti jokin prosentuaalinen määrä taloja, joissa on syttynyt tulipalo. Tästä voisi mahdollisesti päätellä tilastomatematiikan avulla, että piippu, jota ei huolleta säännöllisesti, nostaa riskiä tulipalolle.

2.3 Mikä on vakuutus?

Vakuutuksen ideana on yhteisesti kantaa riskin taakka hieman yksinkertaistetulla esimerkillä havainnollistaen. Aikanaan Suomessakin on ollut pitäjissä useita tiloja. Kuvitellaan, että tällaiselle tilalle sattuisi tulipalo ja sen myötä palanut navetta pitäisi rakentaa uudelleen. Tilanomistaja joutuisi nyt maksamaan omista varoistaan uuden navetan rakentamisen. Tällainen maksaminen olisi tietysti hyvin kallista. Pitäjän tilanomistajat kuitenkin huomasivat yhteisen riskin kantamisen edut. Sen sijaan että yksi henkilö maksaisi koko navettaa, jokainen tilanomistaja osallistuisi kuluihin pienemmällä inhimillisellä summalla. Kulut jakautuisivat kaikkien kesken riskin sattuessa. Myöhemmin kun varojen kerääminen muuttui systemaattisemmaksi ja niitä alettiin kerätä ennakkoon, syntyi vakuutuslaitoksen pohja.

Jukka Rantalan ja Esko Kivisaaren kirjassa Vakuutusoppi selitetään vakuutustoiminta täsmällisesti [2].

> ”Tietyn riskin alaiset yksiköt, vakuutuksenottajat, sopivat vahinkojen tasaamiseen erikoistuneen laitoksen, vakuutuslaitoksen eli vakuutuksenantajan, kanssa siitä, että riskin toteutuessa vakuutuksenantaja korvaa siitä aiheutuneen vahingon. Korvauksensaantioikeuden vastikkeeksi vakuutuksenottajat suorittavat vakuutusmaksun vakuutuksenantajalle.”

Vakuutuksen kuten riskinkin olennainen ominaisuus on sattumanvaraisuus. On sattumanvaraista toteutuuko riski. Sattumanvaraisuus kuvaa myös arpapelejä ja vedonlyöntejä, mutta erona näihin vakuutuksella on ehto, että vahingon on täytynyt tapahtua, jotta

voi saada korvauksen. Mikäli riski on jo tapahtunut, vakuutusta ei voi enää ottaa korvaamaan tätä riskiä. Kotivakuutuksen ottaminen tulipalon jälkeen ei enää korvaa jo palanutta taloa.

Vakuutusmaksun tulee vastata riskin suuruutta. Mitä suurempi riski on, sitä korkeammat maksut ovat. Kalliin uuden urheiluauton vakuutusmaksut ovat todennäköisesti korkeammat kuin muutaman vuoden vanhan farmari auton. Samoin vanhan ihmisen riski kuolla on suurempi kuin nuoren ihmisen. Siksi henkivakuutuksissa hinta saattaa olla halvempi nuorille.

Yleensä vakuutuksenottaja ottaa vakuutuksen turvaamaan itseään. Mutta on myös olemassa vakuutuksia, joita otetaan myös muiden suojelemiseksi. Esimerkiksi aiemmin mainitun kalliin urheiluauton vakuutus on ottanut huomioon myös riskin sille, että jalankulkija jäisi vahingossa alle. Tällöin vakuutus turvaisi myös jalankulkijan.

Vakuutuskelpoisuus on vaatimus myös vakuutettaessa. Tällä tarkoitetaan, että kohteen riskit voidaan ennustaa riittävän tarkasti ja että kohteen vakuutusmaksut voidaan hinnoitella oikein. Myös vahingon riskien on oltava edunsaajasta riippumattomia, jotta tämä voidaan hyväksyä vakuutuskelpoiseksi. Esimerkiksi liikeriskejä ei voida tämän vuoksi vakuuttaa.

Suuret vahingot ja niiden korvaukset kyetään myös rahoittamaan jälkikäteen tarvittaessa vakuutusmaksujen korotuksilla.

Vakuutus itsessään saattaa sisältää erilaisia turvia turvaamaan vakuutettua kohdetta erilaisilta riskitekijöiltä. Esimerkiksi vakuutettu kohde voi olla vene ja veneellä voi olla turvana myrsky ja luonnonilmiö. Tällaisessa tapauksessa veneeseen myrskyn aiheuttamat vahingot vakuutuslaitos korvaisi sopimuksen mukaan.

Ennen korvauksen saamista on usein toimenpiteenä korvaushakemus, jossa haetaan tapahtuneelle vahingolle korvaus. Prosessi itsessään saattaa olla helpoimmillaan itse täytettävä lomake aiheutuneista kuluista mutta saattaa myös joskus tarvita ulkopuolista arvioimaan vakuutetulle kohteelle aiheutuneet vahingot.

Vakuutukset saattavat sisältää ehtoja ja rajoituksia. Rajoitus voi rajoittaa esimerkiksi lemmikin kohdalla korvauksia koiran silmäoireista, jos koiralla on esiintynyt esimerkiksi tulehdusta silmissä useasti kuluneen vuoden aikana.

3 Guidewire ja InsuranceSuite

3.1 Guidewire

Guidewire Software Inc, yleisemmin tunnettu pelkällä nimellä Guidewire on yhdysvaltalainen vuonna 2001 perustettu ohjelmistoyritys [3].

Guidewire-teknologia on alusta joka tuo käyttöliittymän, datamallin, Gosu-ohjelmointikielen ja mahdollisuuden integroida alustan järjestelmiä ulkopuolisiin sovelluksiin. Kaikki Guidewire InsuranceSuiten ohjelmistoperheen ohjelmat käyttävät alustanaan Guidewirea. Guidewiren InsuranceSuite koostuu PolicyCenteristä vakuutuksia varten, ClaimCenteristä korvauksia varten, BillingCenter laskutusta varten ja ContactManager asiakastietojen hallintaa varten.

3.2 BillingCenter

BillingCenter on verkkopohjainen laskutus- ja saamishallintajärjestelmä. BillingCenter perustaa toimintansa automaatioon, joustavuuteen ja ohjaukseen. Automatisoitu laskuttamisen elinkaari mahdollistaa integroinnin nopeasti mahdollisiin ulkoisiin järjestelmiin tai kolmansien osapuolien tuottamiin sovelluksiin.

Näin asiakas voi ottaessaan käyttöön BillingCenter hyödyntää vanhaa laskutusjärjestelmäänsä. Laskutuspyynnöt BillingCenter vastaanottaa joko Guidewiren omasta PolicyCenter-järjestelmästä. Tarvittaessa järjestelmän voi määrittää ottamaan vastaan pyyntöjä myös ulkopuolisestajärjestelmästä. BillingCenter itsessään sisältää myös omia komponentteja datamallissa, oman käyttöliittymän ja tarvittavat yhteydet BillingCenterin integrointiin ulkopuolisen sovelluksen kanssa. [6.]

3.3 ClaimCenter

ClaimCenter on verkkopohjainen yritysohjelmistosovellus. ClaimCenter on hallintatyökalu raportointiprosessin, todentamisen ja vakuutuskorvausten maksamisen hallintaan. ClaimCenter hallitsee koko korvausprosessin elämänkaaren vahingon tapahtuma päivästä korvauksen maksuun.

ClaimCenter hallitsee korvausten finanssitoiminnan. Tämä finanssitoiminnan hallinta sisältää esimerkiksi seuraavia ominaisuuksia:

- Maksetaan korvauksia
- Varaus tehdään ennakoivasti rahalle, jonka odotetaan menevän maksettavaksi korvauksen yhteydessä.
- Pyydetään hyväksyntää käyttäjältä, jolla on korkeammat käyttöoikeudet, mikäli haettava korvattava summa on hyvin suuri.
- ClaimCenterissä on integrointi asiakastietojärjestelmää varten. Guidewiressa tällaisenä toimii ContactManager. ClaimCenter sisältää myös sisälleen rakennetun automaattisen järjestelmän mahdollisten vakuutuspetosten tarkasteluun.
- Hallitaan työtilan korvauksia.

ClaimCenter sisältää näiden lisäksi lukuisia muita ominaisuuksia, joita kaikkia en luettele tässä tiivistelmässä. [5]

3.4 ContactManager

ContactManager on verkkopohjainen asiakastietojen hallintaan keskittynyt järjestelmä, joka kommunikoi saumattomasti muiden Centerien kesken.

3.5 PolicyCenter

PolicyCenter on verkkopohjainen vakuutusjärjestelmä, jossa voi hallita vakuutuksia, luoda uusia vakuutuksia, migroida vanhoja vakuutuksia ja tehdä kaikki tarvittavat toiminnallisuudet vakuutuksen elinkaaren aikana. PolicyCenterissä vakuutusyhtiöt voivat luoda omia vakuutuksiaan tai käyttää valmiita out of the box-toteutuksia. Valmiit vakuutuspohjat on kylläkin toteutettu mukailemaan yhdysvaltalaisia vakuutuksia ja Suomessa vaativat todennäköisesti kustomointia. PolicyCenterin kautta vakuutusyhtiö pystyy arvioimaan riskejä, tekemään tarvittavia muutoksia tarvittaessa vakuutuksiin, katsomaan ja luomaan vakuutuksia [4].

4 PolicyCenterin toiminnallisuus

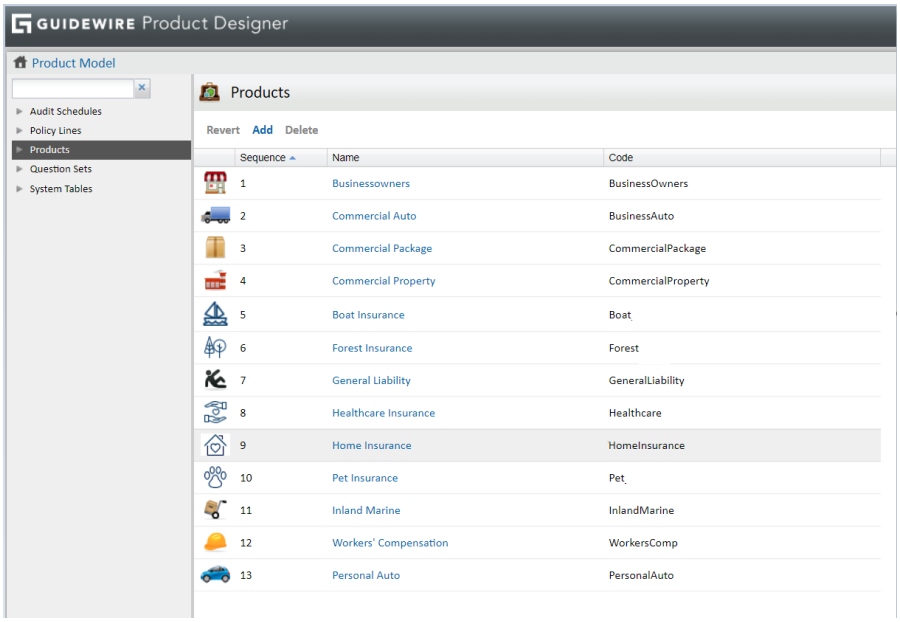
PolicyCenter tallettaa tietoja vakuutukseen liittyen kuten vakuutetut kohteet, sovellettavat ehdot, maksukaudet ja muuta vakuutukseen liittyvää tarpeellista tietoa. Jokaisen onnistuneen PolicyCenterin voimaansaattamisen jälkeen PolicyCenter lähettää lopuksi laskutustiedot BillingCenterille tai tarvittaessa ulkoiselle laskutusjärjestelmälle. PolicyCenterissä pystyy räätälöimään sopivan vakuutustuotteen vastaamaan oman vakuutusyhtiön tarpeita.

Valmiita vakuutuslinjoja/perheitä on PolicyCenterissä:

- Businessowners
- Commercial Auto
- Commercial Property
- General Liability
- Homeowners
- Inland Marine
- Personal Auto
- Worker’s compensation.

Valmiiden vakuutuslinjojen lisäksi on mahdollisuus räätälöidä ihan omia vakuutusyhtiölle sopivampia vakuutuslinjoja esimerkiksi metsävakuutus, sairasvakuutus tai venevakuutus.

Valmiiden ja uusien muokkaaminen tapahtuu ProductDesignerin avulla. ProductDesigner on Guidewiren työkalu nimenomaan vakuutustuotteiden luomiseen sekä muokkaamiseen.



Kuva 1. Kuva Product Designerin sivusta, jolla voidaan lisätä uusia vakuutustuotteita.

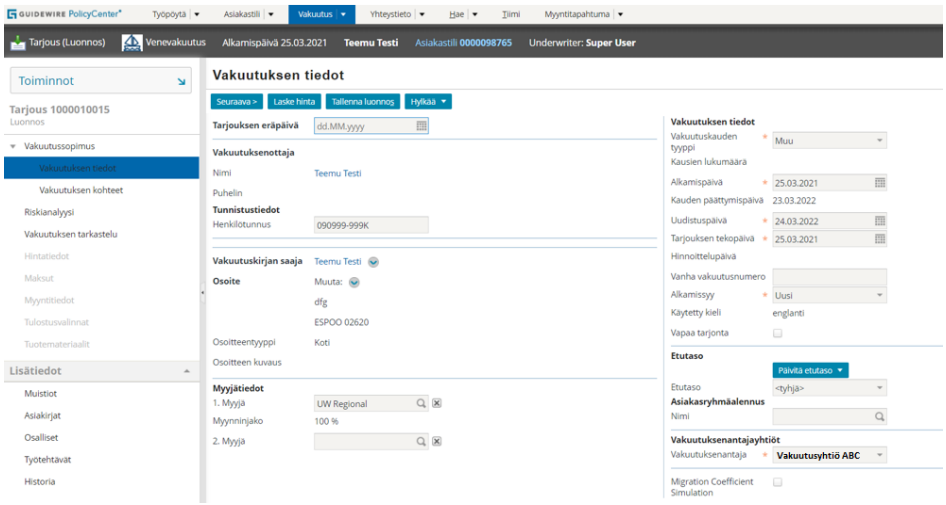
Yksittäinen vakuutuslinja voi sisältää useita vakuutustuotteita esimerkiksi, jos linja olisi tarkoitettu venetuotteille. Sen sisällä voisi olla omia vakuutustuotteita erilaisille veneille moottorivene, purjevene tai ilmatyynyalus.

PolicyCenterin ja vakuutusten hallinta tapahtuu selaimen kautta. Web-käyttöliittymässä on eräänlainen työpöytä, jossa voi hallita vakuutuksia. Tämän työpöydän kautta voi luoda, muuttaa, laskea tarjouksen, saattaa voimaan, uusia, peruuttaa, reinstate eli peruutetun vakuutuksen tuonti uudelleen voimaan, kirjoittaa uudelleen kokonaan vakuutuksen ja auditoida vakuutuksia.

4.1 Vakuutuksen elämänkaari PolicyCenterissä

Vakuutuksen elämä alkaa usein tarjouksena. Asiakkaalla on tarve vakuutukselle esimerkiksi venevakuutukselle. Vakuutustarjoukselle tarvitaan alkuun alla lueteltuna tärkeimmät:

- alkamispäivä, uudistuspäivä ja tarjouksen eräpäivä.
- henkilötiedot kuten puhelinnumero, nimi ja osoite
- vakuutuskauden tyyppi Tasan vuosi/muu
- alkamissyy uusi tai migroitu
- mahdolliset edut esimerkiksi kuulumalla insinööriliittoon tai vastaavaan organisaatioon.
- myyjäntiedot
- vakuutuksenantajayhtiö.

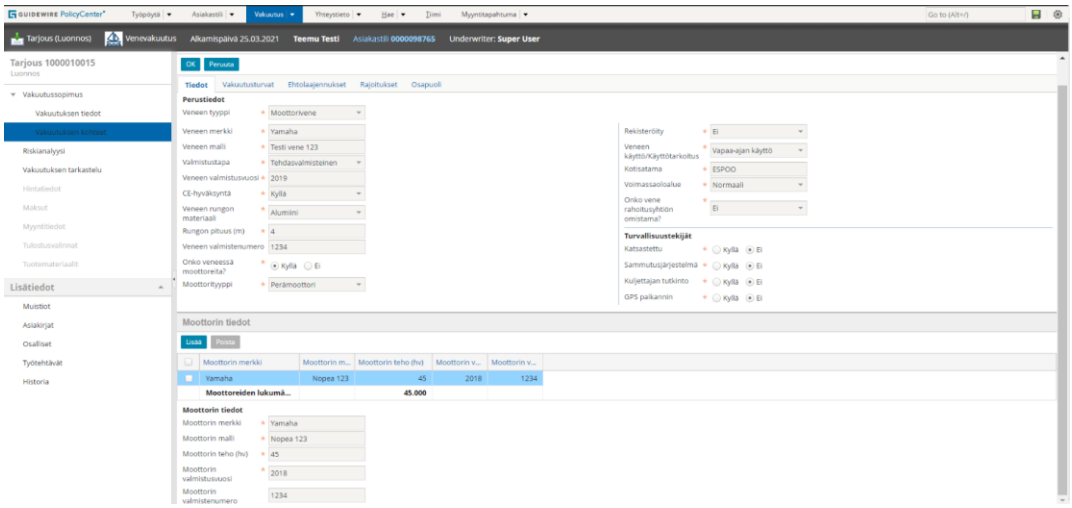


Kuva 1. Kuva PolicyCenterista uuden venevakuutuksen vakuutuksen tiedoista

Seuraavaksi täytettävänä on vakuutuksen kohteet tässä esimerkissä vene. Venevakuutuksessa veneelle tarvitsee valita tyyppiesimerkiksi moottorivene tai purjevene. Tämän jälkeen täytyy veneen tiedot täyttää. Osa näistä tiedoista on pakollisia kuten:

- veneen tyyppi, merkki, malli ja valmistustapa
- rungon materiaali ja pituus

- ce-hyväksyntä
- tieto mahdollisista moottoreita ja niiden tyypistä
- tieto onko vene rekisteröity
- tieto veneen käyttötarkoituksesta
- tieto veneen kotisatamasta
- tieto voimassa oloalueesta

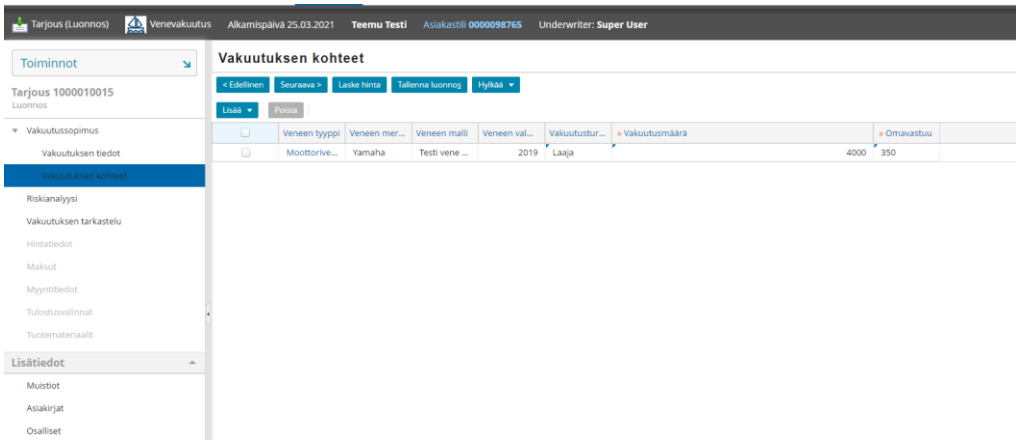


Kuva 1. Kuva välilehdeltä, jossa täytetään vakuutetun kohteen tiedot

Seuraavalta välilehdeltä pääsee vaikuttamaan veneen turviin esimerkiksi säätämään venevastuuvakuutuksen määrää. Ehtolaajennukset ovat myös mahdollisia tarvittaessa. Kolmannella välilehdellä on rajoitukset. Esimerkiksi jos eläinvakuutuksessa on koiralla toistuva silmätulehdus, joka on tullut selville koiran pakollisessa terveystarkastuksessa voisi, mahdollisesti rajoittaa silmään liittyvät oireet vakuutukselta. Viimeisellä välilehdellä voi lisätä osapuolten tiedot esimerkiksi:

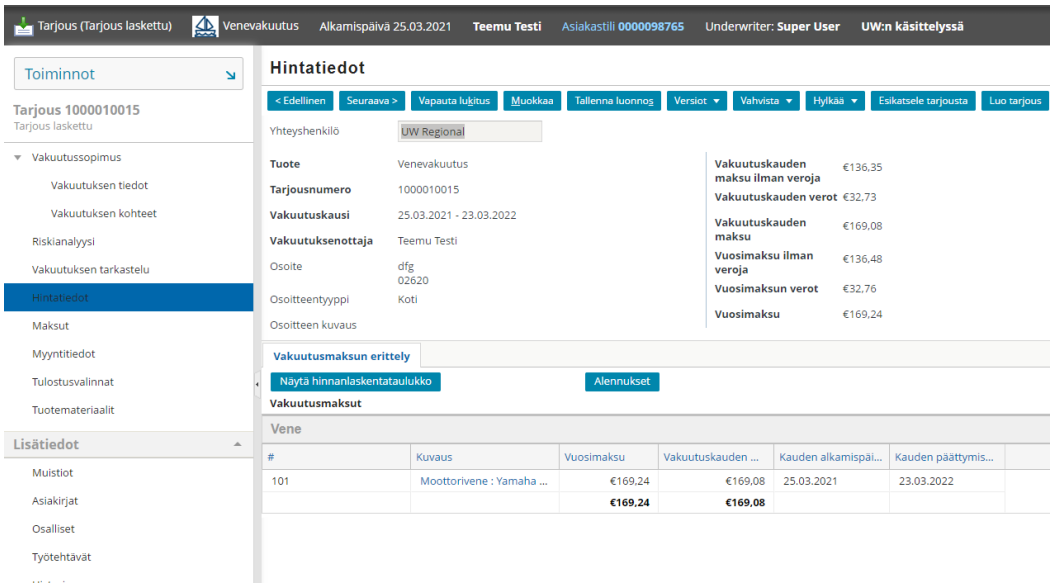
- omistajan
- haltijan
- pantinhaltijan.

Kun kohteen tiedot on täytetty, täytyy vielä syöttää vakuutusmäärä eli korvattavan veneen arvo.



Kuva 1. Kuva vakuutettujen kohteiden listasta, jossa voi asettaa vakuutusmäärän kohteelle.

Tällä hetkellä tarjous on yhä luonnos eli draft-tilassa. Kun klikataan laske tarjous-painiketta, muuttuu tarjous lasketuksi eli quoted-tilaan. Tässä tilassa voidaan nähdä, paljonko vakuutusmaksaisi. Hintatiedoilta näkee myös mahdolliset alennukset kuten liittojen jäsenyyksien tuomat alennukset. Hintalaskentataulukosta näkee tarkemmin hintaan vaikuttavat tekijät.



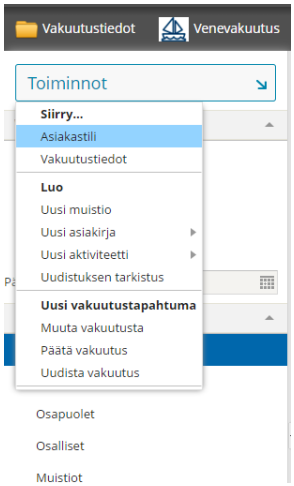
Kuva 2. Kuvakaappaus vakuutuksen hintatiedoista.

Seuraava vaihe olisi päättää maksusuunnitelma vakuutukselle. Mikäli asiakkaalla ei ole ennestään vakuutuksia yhtiössä, täytyy tämä valita. Vaihtoehtoja maksuerille on useita

niitä voi olla esimerkiksi maksu kerran vuodessa tai kolmesti vuodessa. Mikäli asiakkaalla on aiempia vakuutuksia järjestelmässä, pyrkii järjestelmä tällöin saamaan uuden vakuutuksen vanhaan maksusuunnitelmaan. Esimerkiksi aiemman vakuutuksen maksusuunnitelma on ollut kolmesti vuodessa tarjoaisi järjestelmä kyseistä suunnitelmaa uudelle vakuutukselle. Tämä tapahtuisi myös silloin, kun vakuutusotettaisiin kesken vakuutuskauden. Vakuutus 1 olisi otettu 1.1.2020 kolmesti vuodessa maksu suunnitelmalla ja Vakuutus 2 3.3.2020 tarjottaisiin samaa suunnitelmaa. Tällöin vakuutuskahden kausi olisi vuosittaisen sijaan muu, ja kun vakuutuskausi Vakuutus 1 uusiutuisi kirjoitettaisiin Vakuutuskausi 2 uudelleen alkamaan samana päivänä ja sen jälkeen toistumaan vuosittain. On kuitenkin mahdollista tehdä vakuutuksille erilliset suunnitelmat tarvittaessa. Kun maksusuunnitelma on valittu, voidaan tarjous saattaa voimaan. Voimaan laittamisen jälkeen PolicyCenter toimittaa maksutiedot BillingCenterille, joka hoitaa laskutuksen.

Vakuutukselle seuraavaksi voi tehdä esimerkiksi muutoksen tai peruuttaa sen kokonaan. Muutos voi olla, vaikka uuden veneen lisääminen se tapahtuisi Toiminnot-valikon kautta. Prosessi olisi hyvin pitkälti samanlainen kuin uuden vakuutuksen luonti, mutta sen uuden sijaan päivitetään vain aiemman tietoja. Läpikäytävät toimenpiteet ovat lähes identtiset. Vakuutusmuutokselle pitäisi valita alkamispäivä. Tämän jälkeen täytyy tarkistaa vakuutuksen tiedot esimerkiksi nimet, ja osoitteet ovat yhä oikein, muokata joko näitä vakuutuksen tietoja, vakuutettua kohdetta tai vakuutetun kohteen turvia. Tehtyjen muutosten jälkeen lasketaan jälleen tarjous, jos tarjous on hyväksytty asiakkaan puolesta, voidaan muutokset laittaa voimaan.

Toinen mahdollinen mitä vakuutukselle voi tapahtua, on vakuutuksen päättäminen. Päättäminen tehdään myös Toiminnot-valikon kautta painamalla päätä vakuutus.



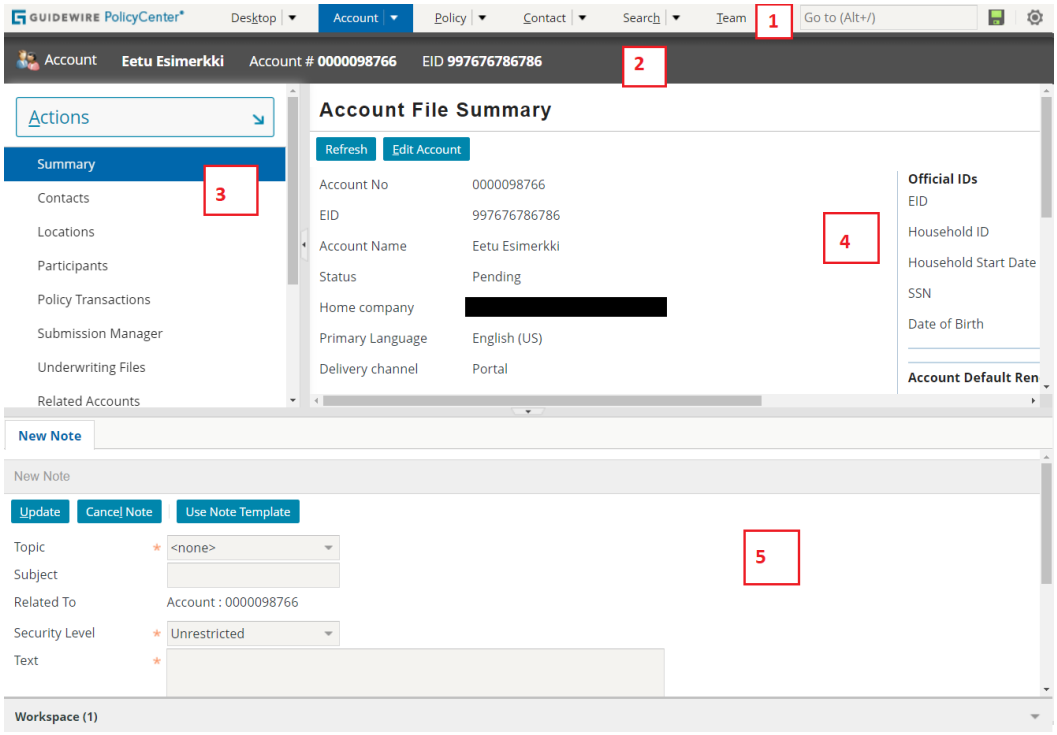
Kuva 3. Kuva Toiminnot-pudotusvalikosta, josta voi aloittaa vakuutukselle muutoksen, päättämisen tai uusimisen.

Vakuutuksen päättämiselle tarvitaan irtisanoja, joka on aina joko vakuutuksenantaja tai vakuutuksenottaja. Sen lisäksi tarvitaan syy vakuutuksen päättämiselle esimerkiksi vaihto toiseen vakuutusyhtiöön tai maksamattomat vakuutusmaksut. Tämän yhteyteen on mahdollista myös lisätä kuvaussyystä tekstikenttään. Vakuutukselle tarvitaan tämän lisäksi tietysti päättämispäivä. Tämän jälkeen voi päättää vakuutuksen joko heti tai ajastaa se myöhemmälle ajankohdalle.

Peruutetun vakuutuksen voi tuoda takaisin PolicyCenteriin voimaansaattamalla sen uudelleen. Voimaansaaton voi tehdä joko vakuutuksen alusta asti eli uudelleen kirjoittaa kokovakuutuksen eli rewrite tai jäljellä olevalle vakuutuskaudelle eli reinstate. Voimaansaatolle pitää antaa syy ja kuvaus aivan kuten peruutuksellekin. Sen jälkeen vakuutukselle lasketaan hinta, täytetään tarvittaessa muita kenttiä, valitaan maksusuunnitelma ja saatetaan voimaan.

4.2 PolicyCenter-käyttöliittymä ja UI

PolicyCenterin käyttöliittymä koostuu viidestä osasta. Nämä osuudet taas koostuvat pcf-tiedostoista. Niistä myöhemmin tässä luvussa.



Kuva 4. Kuvakaappaus PolicyCenterin UI:sta ja numeroidut kohdat, joiden mukaan kerrotaan lisää käyttöliittymästä.

1) The Tab bar eli yläpalkki

Tab bar on yläpalkki, joka sisältää nopeat ja helpot reitit accounteille, contacteille, policyille ja takaisin alkunäkymään eli desktopille. Tab bar on kuvassa numero 1 (kuva 7).

2) The info bar eli infopalkki

Info bar näyttää tärkeimmän informaation liittyen siihen, mitä tällä hetkellä katsotaan. Accountia tutkiessa siinä näkyisivät tilin tiedot kuten nimi ja tilin yksilöity Account Number eli tilinumero. Vakuutusta tutkiessa siinä näkyisi policynumber eli vakuutusnumero. Info Bar on kuvassa numero 2 (kuva 7).

3) Sidebar eli sivupalkki

Sivupalkki sisältää tutkittavaan asiaan liittyen linkit. Vakuutusta tutkiessa sieltä löytyisi vakuutuksen tiedot, maksutiedot ja esimerkiksi korvattavat kohteet. Yksi tärkeä ominaisuus sivupalkilla on actions eli toiminnot, jonka alla ovat tärkeät toiminnallisuudet kuten

luoda vakuutusmuutos tai tehdä tilille uusi vakuutustarjous. Side bar on kuvassa numero 3 (kuva 7).

4) Screen Area

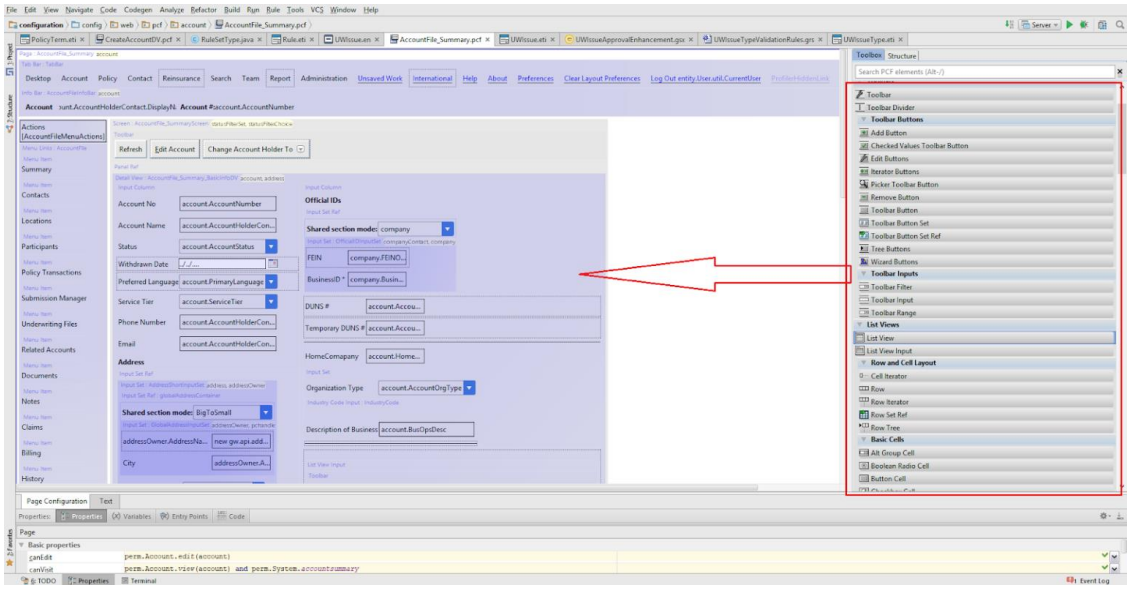
Screen Area näyttää liiketoiminnan kannalta tärkeät tiedot esimerkiksi yhteenvedon tarkasteltavan vakuutuksen tiedoista tai muista sivupalkin valittavista kohteista. Screen on kuvassa numero 4 (kuva 7).

5) Workspace

Workspace-työtila on dymaaminen alue Screen Arealla. Siinä voidaan näyttää helposti esimerkiksi virheviesti, jos käyttäjä jättää täyttämättä pakollisen kentän tai siihen voi avata pienen kirjoitusikkunan, mikäli vakuutukselle haluaa tehdä muistiinpanoja. Workspace on kuvassa numero 5 (kuva 7).

4.3 PCF

PolicyCenterin UI on rakennettu suurimmaksi osaksi Page Configuration Formien pohjalle lyhyemmin pcf-tiedostojen varaan. Nämä pcf-tiedostot voi joko luoda drag and drop -menetelmällä GuideWire-studiossa.



Kuva 5. Kuva Guidewire-studiossa auki olevasta pcf-tiedostosta ja drag and drop-työkaluista.

Esimerkiksi jos haluaisi lisätä uuden kentän tai napin, sen voisi vain vetää ja pudottaa suoraan haluamaansa kohtaan suoraan toolboxista.

Nämä tiedostot itsessään koostuvat xml:stä ja voi myös tarvittaessa muuttaa suoraan xml-tiedostoa. Tämä tapahtuu valitsemalla studiossa Page Configurationin sijaan Text (kuva 8). Tällöin avoinna oleva pcf-tiedosto on katsottavissa suoraan xml muodossa.

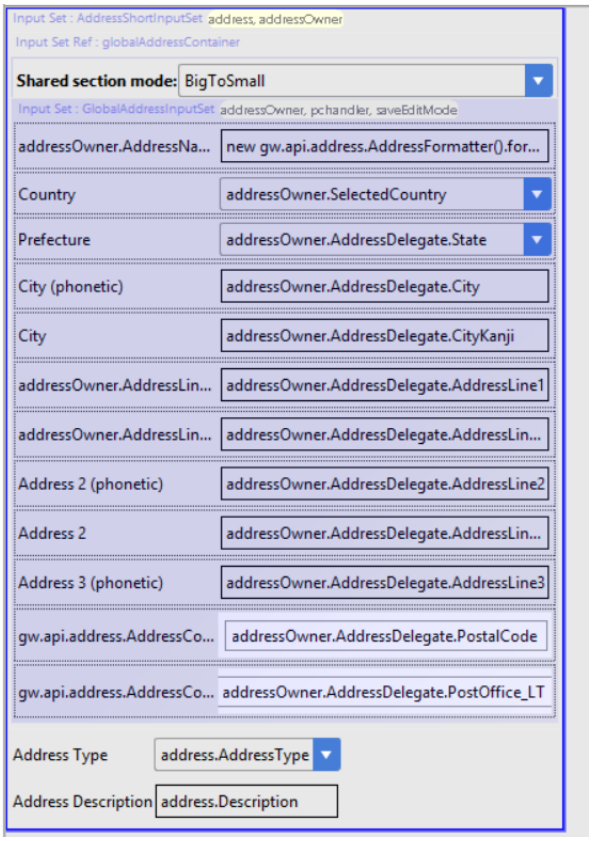
```
<InputColumn>
  <TextInput
    id="AccountNumber"
    label="DisplayKey.get(&quot;Web.AccountFile.Summary.AccountNumber&quot;)"
    value="account.AccountNumber"/>
```

Kuva 6. Esimerkki miten AccountSummaryllä näkyvä AccountNumber näkyy xml:llä.

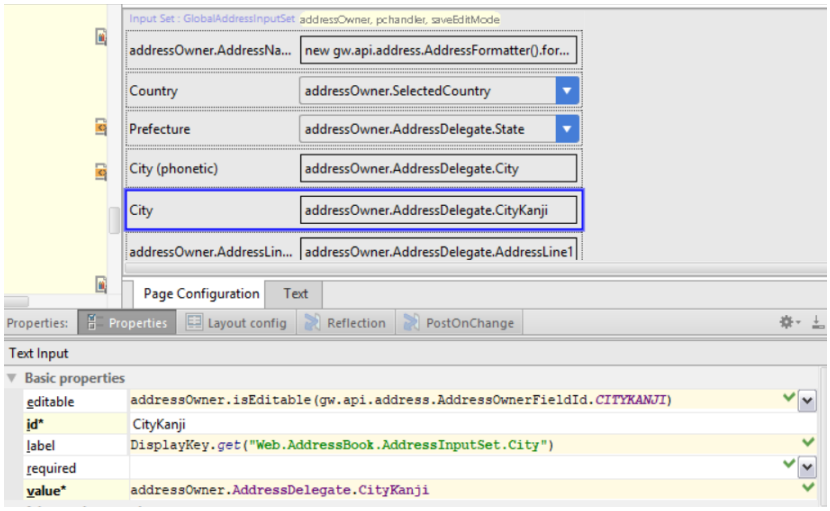
Yllä olevassa esimerkissä nähdään myös DisplayKeyn käyttö. DisplayKey hakee tässä tapauksessa Web.AccountFile.Summary.AccountNumber-avaimelle kuvauksen kielellä,

joka on tällä hetkellä valittuna. Tuon DisplayKeyn myötä käyttöliittymällä näkyy englanniksi teksti Account No, suomeksi teksti Asiakastilin numero ja ruotsiksi Kundkonto Nr.

Jokainen pcf voi koostua luikuisista pienemmistä pcf-osa-alueista. AccountFile_Summary.pcf näkyvä Account No koostuu AccountFile_Summary_BasicInfoDV.pcf:stä. Studiossa katsoessa nämä erilliset alueet erottuvat myös väreistä. Mitä pienempi osa-alue on, sitä tummempi väri on studiossa. Studiossa näiden välillä pääsee liikkumaan helposti klikkaamalla aina värillistä aluetta kahdesti ja pääsee kyseiseen pcf:ään. Näin voi edetä Screenistä lopulta osoitteen syöttämisen kaupunkiin asti. Esimerkiksi ensin on Screen eli jo aiemmin UI:ta käsitellessä mainittu Screen Area. Screenin sisältä löytyy AccountFile_Summary_BasicInfoDV. Kun tutkitaan AccountFile_Summary_BasicInfoDV, huomataan sen sisältävän AddressShortInputSet.pcf.



Kuva 7. Kuva AddressShortInputSet.pcf:stä



Kuva 8. Kuva AddressInputSetin syötettävästä osoitetiedosta kaupunki.

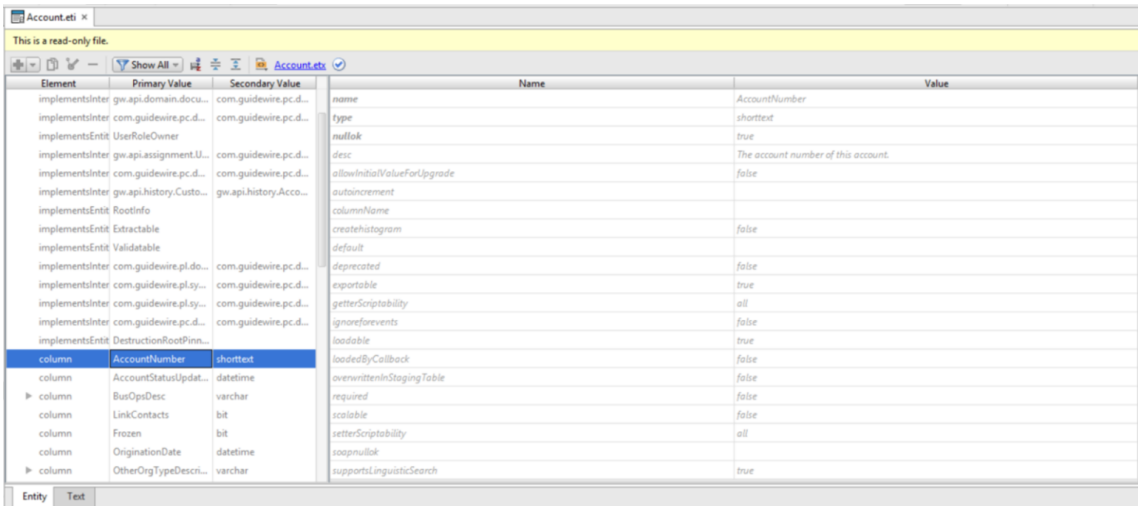
AddressShortInputSet sisältää täytettävät tiedot osoitteen täyttämiselle. Esimerkkinä on kaupunki. Kaupunkia tai Citya klikkaamalla näemme, mitä kaikkea se sisältää (kuva 11), esimerkiksi ID:n, labelin ja valuen. Kaupungin label eli kentän nimi on ’kaupunki’. Se haetaan display keystä. Sen arvoksi annetaan osoitteenomistajan osoite. Tällä syötekentällä Basic Propertiesit, jotka nähdään myös kuvassa (kuva 11) ja Advanced Propertiesit. Advancer Propertiesista voisi esimerkiksi määrittää, kuinka monta merkkiä syötteeseen otetaan tai mitä merkkejä siihen hyväksytään.

4.4 PolicyCenterin datamalli

Guidewiren datamalli oikein yksinkertaisimmillaan on määritelmiä typelisteista ja entityista xml-formaatissa.

Entity koostuu joukosta määriteltyjä kenttiä:

- Column
- Foreign key
- Type key
- Array
- Edge foreign key.



Kuva 1. Kuvakaappaus Guidewire-studiosta Account.eti-tiedostosta

Seuraavana on muutama esimerkki Account entityn kentistä eli Account.eti. Accountilla on column kuten AccountNumber, joka on tilin yksilöidä käyttäjätilin numerokoodi. Accountilla on typekey PreferredCoverageCurrency, joka kertoo käyttäjätilin suosiman valuutan. Arrayna Accountilla voi olla esimerkiksi Notes, joka pitää sisällään listauksen muistioista liittyen käyttäjään.

Typelist määrittelee joukon koodi- ja arvopareja nimeltään typecode. Nämä typecodet sen sijaan määrittelevät entityiden type key-kentät. Typelistejä on useita:

- Custom Typelists on täysin kustomoitava typelist. Tällaisellä voi luoda kenttiä täysin uusiin entityihin tai luoda uusia kenttiä jo olemassa oleviin entityihin. Mutta halutessaan luoda uusia kenttiä uusiin tai vanhoihin entityihin on käytettävä custom typelistiä.
- Extendable Typelist näitä voi muokata ainoastaan sen mukaan, miten ne on määritelty schemassa.
- Internal Typelist on sisäisiä typelisteja. Näitä typelisteja ei voi muokata ollenkaan, sillä ne sisältävät tärkeää Guidewiren sisäistä logiikkaa.

Käynnistyessään Guidewire lataa datamallin metadatan. Ladattu metadata toteuttaa datamallin kokoelmana tietokanta tauluja sovelluksen tietokannassa. Metadata myös syöttää sovelluksen palvelimelle Java- ja Gosu-luokat luodakseen ohjelmallisen käyttöliittymän entityille ja typelisteille tietokannassa.

Guidewirearkkitehtuuri pohjautuu paljon xml-tiedostoihin. Nämä tiedostot antavat sovelluksille niiden oletuskäyttäytymisen, objektit pohjatoteutukselle ja tarvittavat käyttöoikeudet. Guidewiressa muutetaan lopulta vain xml-tiedostoja luomalla Gosu business ruleja, luokkia, enhancementteja ja muita objekteja. Muihin samantapaisiin ohjelmistoihin verrattuna Guidewiressa ei tarvitse niinkään muuttaa itse koodia vaan riittää muutokset edellä mainittuihin ominaisuuksiin.

Data entity on korkean tason objekti, joita käytetään PolicyCenterissä. Tällaisiä ovat esimerkiksi:

- Policy eli vakuutus
- PolicyLine eli vakuutuslinja esimerkiksi venevakuutus
- Coverage eli turva esimerkiksi tuulilasin rikkoutumista varten.

Data entityitä ei voi muokata suoraan, vaan niihin tehdään muutokset aina extensioneiden kautta. Näitä extensioneita on useita erilaisia riippuen siitä, mihin ne liittyvät. PolicyCenteriä konfiguroidessa tarvitaan usein entityiden kenttiä ja arvoja. Guidewiressa tätä helpottamaan on käytössä pistenotaatio. Kun halutaan viitata esimerkiksi accountnumber-kenttään account entityllä tämä löytyisi account.AccountNumber tai toisena esimerkkinä, jos kyseisen accountin vakuutuksen päättämispäivän tarvitsee löytyisi se account.Policy.ExpirationDate.

Workflow on useasti suoritettava toistuva osuus prosessista. Tällainen voi olla esimerkiksi vakuutuksen muuttaminen tai vakuutuksen uudistaminen. Nämä prosessit sitten noudattavat valmista kaavaa, jonka mukaan prosessi toteutetaan. Workflow sisältävät itsessään etappeja, joiden mukaan prosessi etenee. Näistä etappeja voi olla yksi, useampi tai ei yhtään.

Aktiviteetit eli Acitivityt on tehtäviä PolicyCenterissä. Esimerkiksi asiakkaan tapaaminen sopimuksen tietojen selvityksen myötä voi olla aktiviteetti tai tarjouksen tarkastaminen ennen vakuutuksen voimaan saattamista. PolicyCenter seuraa sitten näitä aktiviteetteja helpottaakseen käyttäjien tekemiä vakuutukseen liittyviä toimenpiteitä. Activiteetit antavat esimerkiksi managerien seurata määrättyjä töitä tai tarkastajien tarkistaa vakuutuksen tietoja tarvittaessa. Esimerkiksi vanha vene, jolle merkitään kuitenkin korkea rahallinen arvo voisi nostaa activityn. Tällaisessä activityssä työntekijä tarkistaisi tilanteen,

kuinka vanha vene voi olla arvokas. Aktiviteeteillä on myös niin kutsutut kaavat eli Activity Patternit, joita ne seuraavat. Nämä kaavat standardisoivat aktiviteettien luomisen. Jokainen kaava määrittelee yhden aktiviteetin käsittelyn vakuutukselle tai käyttäjälle.

4.5 Gosu

Gosu on yleiskäyttöinen staattisesti tyypitetty oliopohjainen ohjelmointikieli. Se on suunniteltu helposti lähestyttäväksi. Se on täysin yhteensopiva Javan kanssa ja pohjautuu siihen osittain. Gosussa on kuitenkin otettu Javan hyvät ominaisuudet kuten laajat kirjastot käyttöön ja sen lisäksi laajennettu ominaisuuksia lukuisilla tavoilla. Tässä on listattu muutamia Gosulle ominaisia piirteitä.

- Type inferense yksinkertaistaa koodia ja edesauttaa staattista tyypittämistä. Se tekee koodista myös helpommin luettavaa.
- Lohkot eli lambda-lausekkeet tekevät yksinkertaisten funktioiden kirjoittamisen helpommaksi.

```
14      public static void main(String[] args) throws Exception {
15          List<String> stringLista = Arrays.asList("Abc", "ef", "g", "hijk");
16
17          List<String> pitkaString = new ArrayList<String>();
18
19          for( String s : stringLista ) {
20            if( s.length() > 2 ) {
21              pitkaString.add( s );
22            }
23          }
24
25          Collections.sort(pitkaString, new Comparator<String>() {
26            public int compare( String s, String s2 ) {
27              return s.compareTo( s2 );
28            }
29          });
30
31          StringBuilder sb = new StringBuilder();
32          for( String s : pitkaString ) {
33            if(sb.length() != 0) {
34              sb.append(", ");
35            }
36            sb.append(s);
37          }
38
39          System.out.println(sb.toString()); // tulostaa Abc, hijk
40      };
41  }
```

Kuva 1. Kuvakaappaus Java-koodiesimerkistä

Sama osa koodia kuin kuvassa 1 Gosulla hyödyntäen lambda-lausekkeita:

```
var stringLista = {"Abc", "ef", "g", "hijk"}
var pitkaString = stringLista.where(\s -> s.length > 2)
print( pitkaString.join(", ") ) // tulostaa Abc, hijk
```

- Enhancementsin avulla voidaan lisätä ominaisuuksia ja metodeja jo olemassa oleviin tyyppeihin. Esimerkiksi metodin ’printWarning’ lisääminen string-tyyppiin.

```
package example

enhancement TestiStringEnhancement : String {
    function tulostaVaroitus() {
    print ( "VAROITUS: " + this );
    }
    }
```

- Staattinen tyypitys
- Tuki XML- ja XSD-tiedostoille

Lopuksi vielä yksi koodi esimerkki ehkä se kaikista tunnetuin eli Hello World.

Hello World Javalla:

```
class HelloWorld{
    public static void main(String args[]){
    System.out.println(“Hello World”);
    }
}
```

Gosulla sama koodi mahtuu yhdelle riville:

```
print(“Hello World”)
```

5 Asiakasprojekti

5.1 Työn idea

Asiakas ei halunnut nimeään käytettävän, joten työssä käytän jatkossa nimeä Vakuutusyhtiö AB, Poistuvayhtiö, Jätettäväyhtiö ja uusi Alueyhtiö Fuusioitu yhtiö.

Asiakkaan yhtiö on jaettu useampaan pienempään yhtiöön alueellisesti. Esimerkiksi isolla kaupungilla voisi olla oma yhtiönsä ja pienemmällä kunnalla Pohjois-Suomessa ihan omansa. Joskus kuntamuutosten ja muiden syiden myötä kuitenkin voi joutua yhdistämään tällaiset aiemmin itsenäisesti toimineet alueyhtiöt keskenään tätä kutsun fuusioitumiseksi. Asiakkaalla oli tiedossa, että Poistuvayhtiö ja Jätettäväyhtiö tulisivat fuusioitumaan tulevaisuudessa uudeksi alueyhtiöksi Alueyhtiö Fuusioitu yhtiö.

Aikaisemmassa toteutuksessa oli jo tehty tarvittavat muutokset estämään uusien vakuutusten luonti vanhalle Poistuvayhtiölle, mutta vanha yhtiö pystyi vielä toimimaan homecompanyna tilille. Varsinainen työ ei mahdollista fuusiota vaan ennemminkin tukee tätä prosessia. Työ ei sisällä oikeaa sulautumista tai tietokannan siirtoa. Sen sijaan työssä keskityttiin enemmän UI-muutoksiin PolicyCenterissä ja muutamaan muuhun myöhemmin ongelmaiseksi koituneisiin kohtiin.

Työ itsessään alkoi siitä, että selvitettiin jokaisen Centerin kohdalta, mitä mahdollisesti tällainen fuusio vaatisi, jotta se onnistuisi sujuvasti. Koska työni liittyy PolicyCenteriin, en käy tässä läpi muiden Centerien muutoksia. Työn lopussa kuitenkin palaan aiheeseen uudestaan ongelmien kohdalla.

Työn kuvaus oli tutkia PolicyCenterissä seuraavia asioita ja tarvittaessa tehdä niihin muutoksia:

- Assignment Queues
- Producer Codes
- Groups
- Affinity Groups
- UW Authority profiles
- Sub-groups
- AD mapping

Assignment Queue on jono. Tähän jonoon voidaan asettaa aktiviteetteja ilman, että niitä täytyy suoraan osoittaa yksittäiselle henkilölle. Esimerkiksi vakuutusta tehdessä kohteena voi olla hyvin arvokas vene. Tällainen tilanne voisi luoda aktiviteetin, jossa esimies tai korkeammassa asemassa oleva työntekijä voisi tarkistaa vakuutetun kohteen tai vakuutuksen ennen kuin se hyväksytään. Tämä loisi automaattisesti aktiviteetin, joka menisi jonoon. Tästä jonosta oikeat henkilöt osaisivat sitten ottaa tuon aktiviteetin työn alle.

Jokaisella alueyhtiöllä on oma koodinsa Producer Code, jonka avulla tämä voidaan liittää transaktioihin asiakkaan ja yhtiön välillä. Producer Coden muuttaminen tässä tapauksessa tarkoitti Fuusioituneelle yhtiölle nimen muutosta. Poistuvalle yhtiölle muutos oli tämän tilan muuttaminen aktiivisesta terminated tilaan, jolloin Poistuvalle yhtiölle ei voinut luoda enää vakuutuksia.

Groups ja Sub-groups ovat alueyhtiön ryhmiä ja aliryhmiä. Uudella fuusioituneella alueyhtiöllä nämä muutokset ovat vain muutoksia nimeen ja kuvauksiin. Poistuvalla yhtiöllä nämä ryhmät pitää saada poistettua käyttöliittymältä.

Affinity Groups ovat ryhmiä, joihin liitetty asiakas on oikeutettu esimerkiksi alennuksiin. Tällainen ryhmä voisi olla esimerkiksi etukorttijäsenyys. Näihin ei kuitenkaan tarvinnut lopulta tehdä muutoksia ollenkaan.

UW Authority profilet ovat profiilien nimiä, joilla on oikeuksia esimerkiksi hyväksyttää ongelmallisia vakuutuksia. Tällainen voisi olla esimerkiksi UW_PET_Uusi_Fuusioitu yhtiö. Nämä muutokset olivat vain pelkkiä nimimuutoksia.

AD mapping eli admin-käyttäjien uudelleenkartoitus vanhalta Poistuvalta yhtiöltä Uudelle Fuusioidulle yhtiölle. Tämä työ ei myöskään jäänyt meidän toteutettavaksi, vaan nämä muutokset toteuttivat lopulta integraatiotiimi.

Hyväksymiskriteereissä kriteerinä oli päivittää kaikkialla käyttöliittymässä näkyvissä paikoissa alueyhtiöiden nimet vastaamaan uutta nimeä. Tämän lisäksi piti katsoa, ettei vanhalla yhtiöllä olisi enää näkyvissä aiemmin mainittuja muutettavia kohteita. Myöskään Poistuvayhtiö ei voisi olla enää käytettävissä niin kutsuttuna Homecompanyna tai Underwriter companyna eli Poistuvayhtiö ei voisi enää luoda vakuutuksia uusille asiakkaille ja tätä yhtiötä ei voisi valita uusille asiakkaille käytettäväksi. Käyttäjänä minun ei pitäisi nähdä Poistuvayhtiö enää käyttöliittymässä. Poistuvayhtiö pitäisi laittaa ei toiminnassa olevaksi ja estää uusien aktiviteettien luontia tälle yhtiölle.

5.2 Itse työ

Ensimmäinen asia projektissa oli tehtävänä selvittää, missä kaikkialla käytetään vanhaa käyttöön jäävää yhtiötä. Guidewire-studiossa on useita hyödyllisiä pikakomentoja esimerkiksi ctrl+shift+f pystyi hakemaan missä kaikkialla studion koodissa esiintyy, vaikka tietty sana.

Tätä käyttäen alkoi muutettavan koodin kartoitus. Ensimmäinen muutettava paikka oli UWCompanyCode.ttx. Tämä ttx-tiedosto on UWCompanyCode.tti-laajennus. Sen avulla PolicyCenteriin lisätä uusia underwritereita, mutta tässä tapauksessa muutettaisiin olemassa olevaa.

```
typelistextension | typecode
1  <?xml version="1.0"?>
2  <typelistextension
3    xmlns="http://guidewire.com/typelists"
4    desc="A code for an underwriting company"
5    name="UWCompanyCode">
6    <typecode
7      code="1111_11111"
8      desc=""
9      name="Alueyhtiö Fuusioitu yhtiö"/>
10   <typecode
11     code="2111_11111"
12     desc=""
13     name="Acme Medium Hazard Insurance"/>
14   <typecode
15     code="3111_33333"
16     desc=""
17     name="Acme High Hazard Insurance"/>
18   <typecode
19     code="4111_44444"
20     desc=""
21     name="Four Corners Low Hazard Casualty"/>
22   <typecode
23     code="5666_55555"
24     desc=""
25     name="Four Corners Medium Hazard Casualty"/>
26   <typecode
27     code="6666_66666"
28     desc=""
29     name="Four Corners High Hazard Casualty"/>
30   <typecode
31     code="7777_12345"
32     desc=""
```

Kuva 1. Kuvakaappaus uwcompanycode.ttx-tiedostosta

Tällaisessa muutoksessa riittäisi pelkän nimen muuttaminen ja niin myös tehtiin.

Seuraavat muutettavat kohteet löytyivät lokalisaatiotiedostoista typelisteista, joissa oli määritelty typekey fuusioidulle yhtiölle. Tämä muutettiin kolmessa paikassa:

- typelist_en_US.properties
- typelist_sv_SE.properties
- typelist_fi_FI.properties.

Tämä täytyi muuttaa jokaisessa, jotta kieltä vaihtaessa käyttäjälle näkyisi aina yhtiön nimi oikein kirjoitettuna. Alla on kuvankaappaus typelist_fi_FI.properties tehdystä muutoksesta.

```
TypeKeyDescription.Months.june=Kesäkuu
TypeKeyDescription.Months.july=Heinäkuu
TypeKeyDescription.Months.august=Elokuu
TypeKeyDescription.Months.september=Syyskuu
TypeKeyDescription.Months.october=Lokakuu
TypeKeyDescription.Months.november=Marraskuu
TypeKeyDescription.Months.december=Joulukuu
TypeKey.Homecompany.Fuusioituyhtiö=Alueyhtiö Fuusioitu yhtiö
TypeKeyDescription.Homecompany.Fuusioituyhtiö=Alueyhtiö Fuusioitu yhtiö
```

Kuva 1. Kuva typelist_fi_FI.properties-tiedostossa näkyvistä muutoksista

Seuraava muutos löytyi undewriting_companies.xml-tiedostosta. Tällaiseen tiedostoon on koostettu jokainen alueyhtiö ja sen producer code, status ja muuta tarpeellista. Tiedostossa piti tehdä jälleen nimimuutos Alueyhtiö Fuusioidulle yhtiölle. Tämän lisäksi täällä asetettiin vanhan poistuvan yhtiön status tilaan retired eli asetettiin tämä pois käytöstä. Tekemällä tämän muutoksen kyseistä poistuvaa alueyhtiötä ei saanut enää lisättyä uudelle tilille homecompanyksi.

```
import | UWCompany | ParentName
<?xml version="1.0"?>
<import>
  <UWCompany public-id="uwc:fuus">
    <Code>950</Code>
    <NAICCode/>
    <Name>Fuusioitu Alueyhtiö</Name>
    <ParentName>Vakuutusyhtiö AB</ParentName>
    <State/>
    <Status>Active</Status>
  </UWCompany>
  <UWCompany public-id="uwc:poist">
    <Code>902</Code>
    <NAICCode/>
    <Name>Poistuva yhtiö</Name>
    <ParentName>Vakuutusyhtiö AB</ParentName>
    <State/>
    <Status>Retired</Status>
  </UWCompany>
```

Kuva 2. Kuvakaappaus underwriting_companies.xml-tiedostossa näkyvistä muutoksista

Seuraava muutettavat löytyivät admindataloader-tiedostoista. Tiedostot nämä tiedostot luetaan sovelluksen käynnistyessä. Ensimmäisessä versiossa eli 01_adl-tiedosto sisälsi lähemmäs kuusituhatta riviä koodia, jossa viitattiin vanhaan Poistuvaan yhtiöön sekä jäljelle jäävään yhtiöön, millä oli vielä väärä nimi. Ensimmäisellä kerralla teimme suoraan uuden adl-tiedoston järjestysnumero 05. Tiedosto nimettiin tämän mukaan ja kopioitiin kaikki 01 tiedostossa olleet rivit ja tehtiin niihin tarvittavat muutokset.

```
<ProducerCode public-id="impprdcd:fuusioituyht">
    <Address/>
    <AddressPublicID/>
    <AppointmentDate/>
    <Branch/>
    <Code>fuusioitu</Code>
    <CommissionPlans>
      <CommissionPlan public-id="impcommplan:15">
        <CommissionPlanID/>
        <Currency>eur</Currency>
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
      </CommissionPlan>
    </CommissionPlans>
    <Description>fuusioitu alueyhtiö</Description>
    <Description_L10N_ARRAY>
      <ProducerCode_Description_L10N public-id="prdcode_desc:13">
        <Language>en_US</Language>
        <Owner public-id="impprdcd:fuusioituyht"/>
        <Value>fuusioitu alueyhtiö</Value>
      </ProducerCode_Description_L10N>
      <ProducerCode_Description_L10N public-id="prdcode_desc:35">
        <Language>fi_FI</Language>
        <Owner public-id="impprdcd:fuusioituyht"/>
        <Value>fuusioitu alueyhtiö</Value>
      </ProducerCode_Description_L10N>
      <ProducerCode_Description_L10N public-id="prdcode_desc:57">
        <Language>sv_SE</Language>
        <Owner public-id="impprdcd:fuusioituyht"/>
        <Value>Nya företag</Value>
      </ProducerCode_Description_L10N>
    </Description_L10N_ARRAY>
    <Organization public-id="systemTables:1"/>
    <Parent/>
    <PreferredUnderwriter/>
    <ProducerCodeRoles>
      <ProducerCodeRole public-id="impprodrole:133">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_submission"/>
      </ProducerCodeRole>
      <ProducerCodeRole public-id="impprodrole:134">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_renewals"/>
      </ProducerCodeRole>
      <ProducerCodeRole public-id="impprodrole:135">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_policychanges"/>
      </ProducerCodeRole>
      <ProducerCodeRole public-id="impprodrole:136">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_cancellations"/>
      </ProducerCodeRole>
```

```
      <ProducerCodeRole public-id="impprodrole:137">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_basics"/>
      </ProducerCodeRole>
      <ProducerCodeRole public-id="impprodrole:138">
        <ProducerCode public-id="impprdcd:fuusioituyht"/>
        <Role public-id="producercode_reinstate"/>
      </ProducerCodeRole>
    </ProducerCodeRoles>
    <ProducerStatus>Active</ProducerStatus>
    <TerminationDate/>
  </ProducerCode>
```

Esimerkkikoodi 1. producer code uuden fuusioidun yhtiön muutoksista admin dataloaderissa

Ajaessa automaattisia jenkins-testejä tämä viimeinen tiedosto kuitenkin kaatoi testit. Syyksi selvisivät tiedoston suuri koko ja rivien määrä. Kaikkia kuutta tuhatta riviä ei tarvitsisi siirtää tähän uuteen tiedostoon. Sen sijaan riittäisivät vain ne rivit, joihin muutokset liittyivät. Edellä ollut esimerkki näyttäisi näiden muutosten jälkeen huomattavasti siistimmältä.

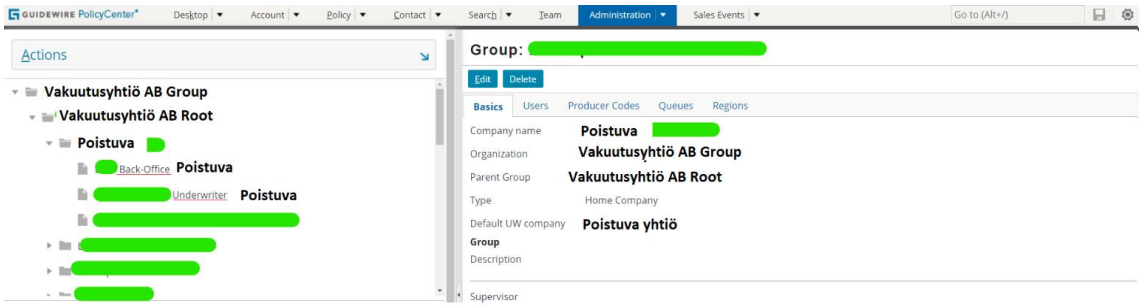
```
<ProducerCode public-id="impprdcd:fuusioituyht">
    <Code>fuusioitu</Code>
    <CommissionPlans>
    <Description>Fuusioitu Alueyhtiö</Description>
    <Description_L10N_ARRAY>
      <ProducerCode_Description_L10N public-id="prdcode_desc:13">
        <Value>Fuusioitu Alueyhtiö</Value>
      </ProducerCode_Description_L10N>
      <ProducerCode_Description_L10N public-id="prdcode_desc:35">
        <Value>Fuusioitu Alueyhtiö</Value>
      </ProducerCode_Description_L10N>
      <ProducerCode_Description_L10N public-id="prdcode_desc:57">
        <Value>Nya företag</Value>
      </ProducerCode_Description_L10N>
    </Description_L10N_ARRAY>
    <Organization public-id="systemTables:1"/>
  </ProducerCode>
```

Tässä vain yksittäinen esimerkki yhdestä muutetusta producer codesta. Näiden lisäksi admindataloaderissa piti päivittää:

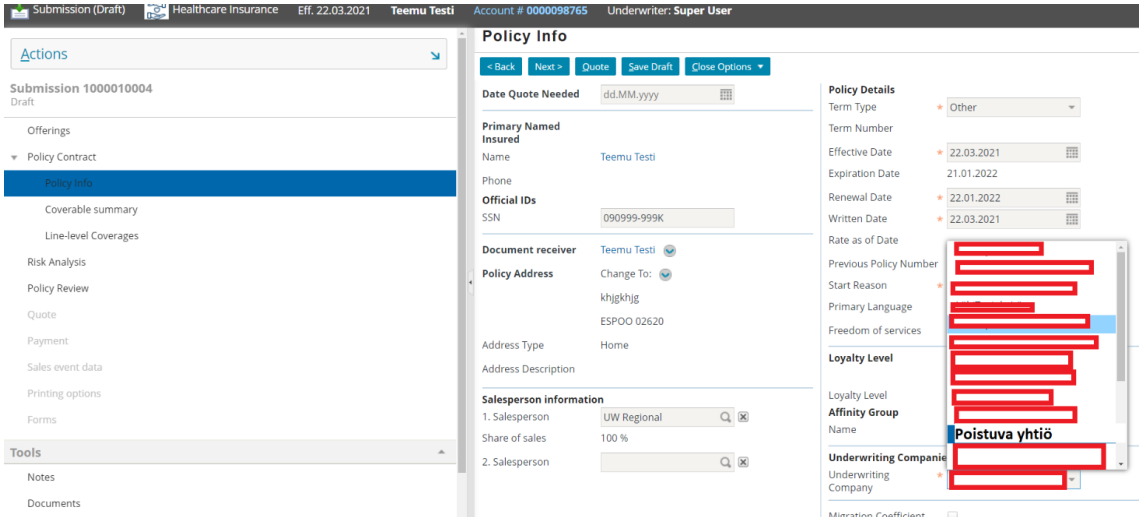
- Producer coden tietojen päivitys uudelle yhtiölle
- uwauthorityprofileiden kuvauksen ja nimen muuttaminen jokaiselle vakuutuslinjalle
- regionin päivitys uudelle yhtiölle
- groupin päivitys uudelle ryhmälle
- groupin päivitys uuden ryhmän alla olevalle Underwriter-ryhmälle
- groupin päivitys uuden ryhmän alla olevalle back officelle

- poistuvan yhtiön producercoden asettaminen tilaan terminated, jotta sille ei tehtäisi enää uusia vakuutuksia
- jonojen nimi muutokset uudelle yhtiölle
- poistuvan yhtiön auw profiilien nimien perään liite _Not_In_Use, jotta kukaan ei niitä vahingossa käyttäisi.

Muutosten jälkeen alkoi projektin testaaminen. Testatessa kuitenkin selvisi, että Poistuva yhtiö ei ollut vielä täysin piilotettu käyttöliittymästä, vaikka se oli asetettu Underwriting_companies.xml-tiedostossa jo tilaan retired. Käyttöliittymässä näkyi vielä admin-sivuilla ryhmä, underwriter ja back office. Nämä näkyivät myös uutta vakuutusta tehdessä.



Kuva 1. Kuvakaappaus käyttöliittymän Group-näkymästä, jossa näkyi vielä Poistuva yhtiö



Kuva 2. Kuvakaappaus new submission-näkymästä, jossa näkyi vielä poistuva yhtiö pudotuslistassa.

Ongelmaksi näytti muodostuvan, ettei Poistuvaa yhtiötä voinut asettaa suoraan pois käytöstä. Tämä olisi käynyt kaikista helpoten asettamalla UWCompanyCode.ttx typelist extensionissa kyseisen yhtiön retired-tilan arvoksi true. Tämä aiheutti kuitenkin ongelmia esimerkiksi sen, että palvein ei suostunut käynnistymään tarkoittaen, ettei koko PolicyCenteriä voinut ajaa paikallisesti omalla koneella.

Ratkaisu löytyi lopulta admindataloader-tiedostoista. Tiedostossa Poistuvan yhtiön ryhmällä oli retire- arvo. Tämä arvo vaikutti suoraan siihen, kuinka tämä yhtiö näkyi käyttöliittymässä. Arvo oli oletuksena 0 ja oletusarvollaan Poistuva yhtiö näkyi yhä käyttöliittymässä väärässä paikassa. Arvon muututtua 1:een eli true Poistuva yhtiö ei näkynyt enää käyttöliittymässä eikä UWCompanyCode.ttx-tiedostossa olleet ongelmat toistuneet.

Muutosten jälkeen koodimuutokset kävivät läpi koodin tarkastamisen ennen kuin muutokset propagoitiin func-testiympäristöön.

5.3 Demo

Asiakkaalle pidettiin Demo, jossa esiteltiin jokaisen Centerin toiminnallisuus muutoksien jälkeen ja todennettiin, että nämä muutokset toimivat. PolicyCenterin kohdalla käytiin läpi aiemmin listatut vaatimukset ja huomattiin, että kaikki nämä täytettiin. Sama tehtiin myös ClaimCenterin kohdalla, mutta siellä oli vielä muutamia asioita korjattavana demon jälkeen.

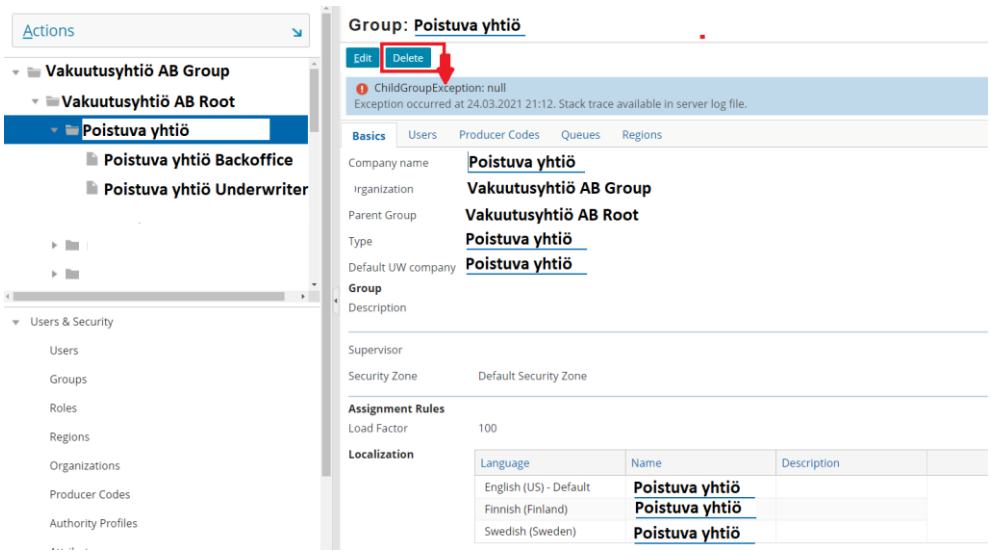
5.4 Ongelmat

Funktionaalisessa testaamisessa ei huomattu muita ongelmia kuin aiemmin mainittu pudotusvalikossa nähtävä väärä alueyhtiö sekä uuden vakuutuksen luontivaiheessa pystyi valitsemaan Poistuvan yhtiön. Muut testit menivät läpi samoin myös nämä korjausten jälkeen. Tämä oli sama ympäristö, jossa ratkaisu myös esiteltiin asiakkaalle. Seuraava testaamisen vaihe oli integraatio testaaminen. Integraatio testeissä selvitettiin, ettei rat-

kaisu tuottanut ongelmia muiden integroitujen ohjelmistojen ja järjestelmien kanssa. Tässäkään vaiheessa ei vielä huomattu virheitä, joten ratkaisu päätyi seuraavaksi asiakkaalle niin kutsuttuun hyväksymistestaukseen eli user acceptance testingiin myöhemmin nimellä UAT-testaus.

UAT-testauksessa huomattiin virheitä. Ensimmäinen virhe, joka tuli vastaan, oli että tietyt Admin-käyttäjät eivät kyenneet kirjautumaan sisään. Näitä käyttäjiä yhdisti se, että he olivat aiemmin kuuluneet Poistuvan yhtiön ryhmään. Tätä tutkiessa huomattiin, että PolicyCenterin, BillingCenterin ja ClaimCenterin lähestyminen Poistuvan yhtiön muutoksiin poikkesivat toisistaan. Varsinkin ClaimCenterin ja PolicyCenterin välinen ero koitui ongelmaksi. Koska työ käsittelee enimmäkseen PolicyCenteriä, en kerro näiden ratkaisujen eroavaisuuksista.

PolicyCenterissä viimeinen muutos, mikä koodiin tehtiin, oli Poistuvan yhtiön groupin asettaminen tilaan retired = 1 eli true admindataloader-tiedostossa. Tämä tehtiin sen takia, että sai Poistuvan yhtiön pois UI-erinäkymistä, joissa se oli näkynyt vielä funktionaalisessa testaamisessa. Tässä oli kuitenkin jäänyt huomioimatta, että ryhmällä oli kaksi aliryhmää Underwriter ja Back Office. Kun tällaisen retired-muutoksen tekisi UI:n kautta estäisi PolicyCenterin tämän. Tällöin järjestelmä ilmoittaisi, ettei kyseistä ryhmää voi laittaa retired-tilaan, koska kyseisellä ryhmällä on kaksi aliryhmää, jotka ovat yhä tilassa active.



Kuva 3. Kuvakaappauksessa error-viestistä, joka tulee Poistuvaa yhtiötä poistettaessa.

Oikea tapa olisi UI:n kautta asettaa ensin aliryhmät retired ja sitten vasta itse pääryhmä (Kuva 17.). Kuvassa näkyvä error-viesti on eri kuin func-ympäristössä tuleva, sillä kuva on otettu lokaalista ympäristöstäni. Mutta virhe viesti tuli samaan kohtaan ja herjasi vielä aktiivisena olevista aliryhmistä. Kun muutoksen tekee koodin kautta, ei PolicyCenter tarkista, mihin tilaan aliryhmät ovat jääneet. Tämän myötä PolicyCenteriin muodostui omituinen tilanne, jossa itse pääryhmää ei ole, mutta sen alla olevat aliryhmät ovat vielä olemassa. Tämä tilanne aiheutti muun muassa seuraavaksi lueteltavia ongelmia:

- Voimassa oleville Poistuvan yhtiön aliryhmille pystyi muodostamaan activityja. Activityt poistetulle yhtiölle taas vaikeuttivat ClaimCenterin toimintaa. ClaimCenter ei voinut laittaa Poistuvaa yhtiötä pois käytöstä voimassa olevien activityjen takia.
- Poistuvan yhtiön admin käyttäjät olivat vielä kytköksissä Poistuneeseen yhtiöön. Tämän myötä heidät lukittiin järjestelmästä ulos.
- Ongelmassa huomattiin myös, että Poistuvalle yhtiölle oli onnistuttu luomaan uusia tarjouksia tilassa new, tarjouksia tilassa quoted, tarjouksia tilassa draft. Järjestelmään oli myös eksynyt yksittäinen voimassa oleva vakuutus Poistuvalle Yhtiölle.
- Useita käyttäjätilejä, joilla oli useita homecompanyita. Tällainen tilanne voisi tapahtua esimerkiksi luomalla vakuutuksen, kun tilin homecompany on Poistuva yhtiö. Tämän jälkeen kontaktin tietoja muutettaisiin ja vaihdettaisiin mikä tahansa uusi yhtiö.

5.5 Ratkaisut

Ratkaisun keksiminen tähän ongelmaan oli hyvin hankalaa, koska ongelma huomattiin niin myöhään. Ratkaisua lähdettiin tutkimaan usealta kannalta.

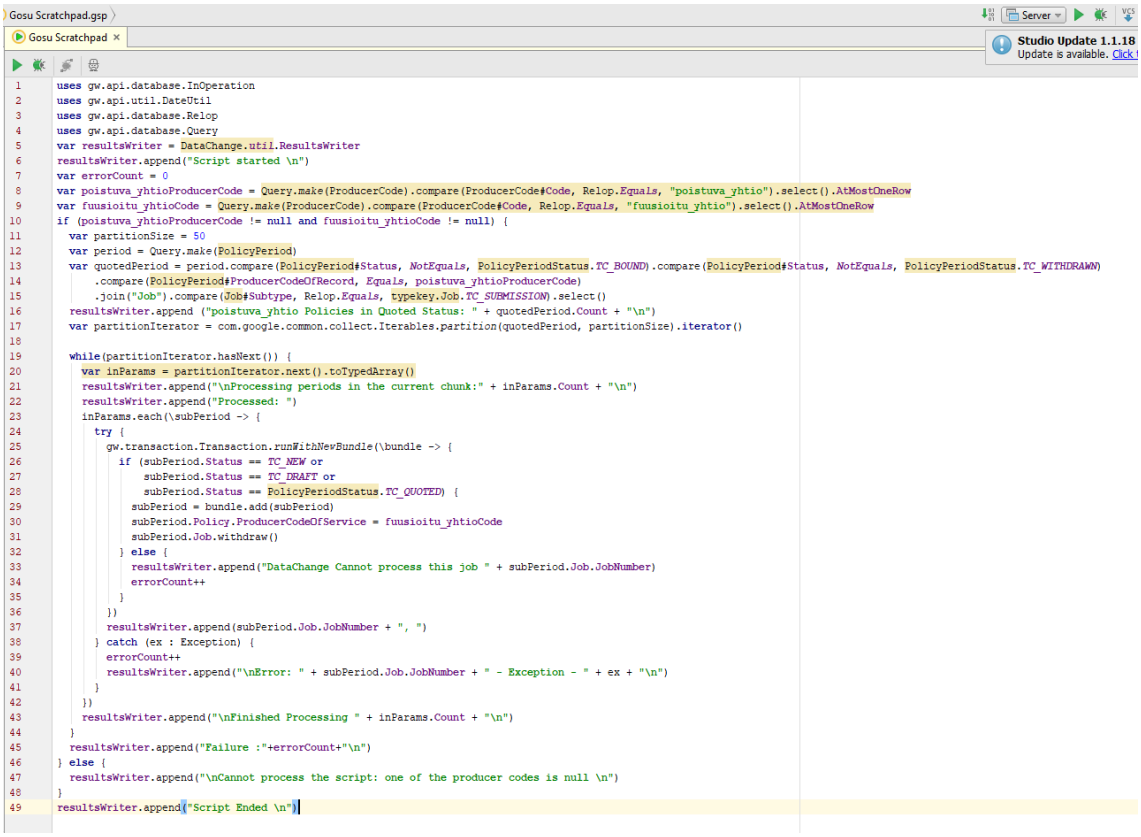
Ensimmäinen vaihtoehto oli muuttaa olemassa olevaa admindataloader-tiedostoa ja päivittää sinne ryhmän lisäksi myös aliryhmät tilaan retired. Tässä koitui kuitenkin ongelmaksi, mitä kävisi olemassa oleville tarjouksille ja sille yhdelle voimassa olevalle vakuutukselle. Toinen ongelma olisi vanhaa admindataloader-tiedostoa muokatessa pitäisi regressio testata kaikki sen jälkeen tulleet admindataloader-tiedostot. Tämä toisi paljon ylimääräistä työtä, mikä ei ole toivottu lopputulos.

Toinen vaihtoehto oli luoda täysin uusi adimindataloader-tiedosto. Tämä pitäisi sitten käydä kaikki testiympäristöt läpi ja testata myös perusteellisesti. Aina uusia admindataloader-tiedostoja lisätessä ympäristöön oli nollattava tietokanta. Tämän myötä pitäisi luoda kaikki uudet testikäyttäjät ja muu testaukseen tarpeellinen. Tämä olisi myös hyvin aikaa vievää, sillä tämä pitäisi tehdä ensin FUNC-ympäristössä, sen jälkeen SIT-ympäristössä ja viimeiseksi vielä UAT-ympäristössä.

Kolmas vaihtoehto olisi skriptien avulla muokata tietokannassa olevia vakuutuksia, tarjouksia ja käyttäjiä. Tämä ratkaisu ei vaatisi tietokannan nollausta tai regressio testaamista. Kolmas vaihtoehto oli tehokkain ratkaisu ja tämän avulla lähdettiin korjaamaan ongelmaa. Koska projekti on vielä työn kirjoitushetkellä kesken, ei välttämättä kaikkia ongelmia ole vielä löydetty. Kerron kuitenkin niistä korjaustoimenpiteistä, jotka työn kirjoitushetkellä oli jo tehtynä.

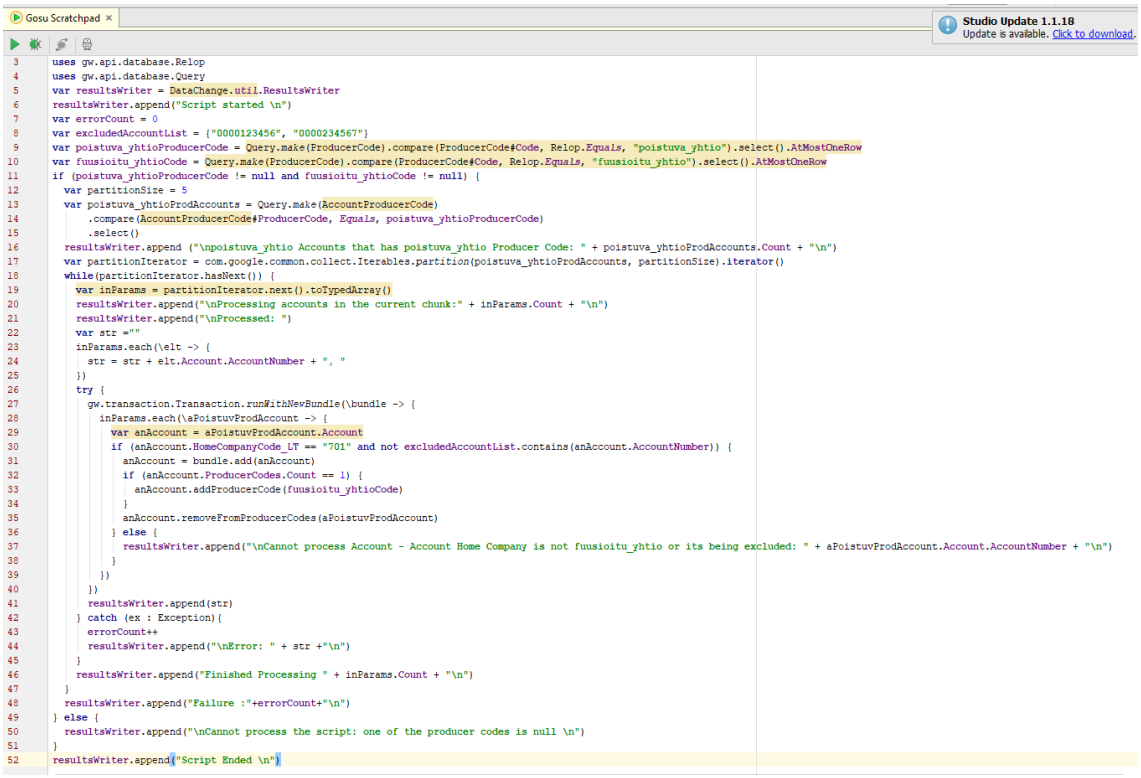
Skriptit piti ensin rekisteröidä ja sen jälkeen suorittaa tuotannossa. Ennen skriptien suorittamista tuotannossa ne kuitenkin testattiin HotFix-korjausympäristössä. Tässä ympäristössä ne testattiin poimimalla ongelmallisia käyttäjiä tuotantoympäristöstä HotFixiin ja todentamalla tietokannasta SQL-kyselyillä, että tiedot olivat muuttuneet oikeiksi.

Ensimmäinen skripti keräsi kaikki Policy Transactionit, jotka olivat tilassa New, Draft tai Quoted. Skripti katsoi nämä läpi ja sen jälkeen asetti niille Producer Code Poistuva Yhtiö tilalle Fuusioitu alueyhtiö. Asiakkaan kanssa sovimme, että kaikki nämä transaktiot voisi asettaa tilaan withdrawn, koska näitä ei tultaisi käyttämään, mutta niihin pitäisi päästä käsiksi. Tämän skriptin jälkeen kaikki vanhalla yhtiöllä olleet transaktiot olivat jälleen avattavissa ja tilassa withdrawn.



Kuva 1. Kuvakaappaus ensimmäisestä skriptistä

Toinen skripti keräsi kaikki käyttäjätilit, jotka olivat producer code Poistuva yhtiö. Tämä skripti kuitenkin jätti pois listauksesta muutaman hieman ongelmallisemman käyttäjätilin. Kun lista tileistä oli kerätty, skripti asetti niille uuden Fuusioidun alueyhtiön producer code-deksi.



Kuva 2. Kuvakaappaus toisesta skriptistä

Kolmannessa skriptissä käytiin läpi käyttäjät, joilla oli useita producer codeja. Näistä yhtiöistä vähintään yksi oli Poistuva yhtiö. Mikäli producer codeja olisi useampi poistettaisiin Poistuva yhtiö. Jos olisi vain yksi Producer Code poistettaisiin, se ja sen tilalle asetettaisiin kyseisen accountin default underwriting company esimerkiksi alueyhtiö ABC.

6 Yhteenveto

Työn oli alun perin hahmoteltu pienkehitysprojektiksi. Ideana oli edesauttaa kahden olemassa olevan alueyhtiön Poistuva yhtiö ja Fuusioitu alueyhtiö yhdistymistä. Itse työn tarkoitus oli tukea tätä toimenpidettä.

Työssä tehtiin enimmäkseen muutoksia, joilla saatiin käyttöliittymästä Poistuva yhtiö pois käytettävistä ja pois näkyvistä. Tämä tehtiin muun muassa tekemällä muutoksia pcf-, admindataloader- ja muutamiin xml-tiedostoihin.

Työssä havaittiin ongelmalliseksi ei yhtenäinen tapa lähestyä ongelmaa eri järjestelmien välillä, joiden pitäisi kommunikoida ja toimia saumattomasti keskenään. Tämä ei yhtenäinen tapa toteuttaa fuusiota tukevat työt tuotti paljon ylimääräistä työtä. Toinen havainto oli se, että työssä olevan Poistuvan yhtiön kaikkia kytköksiä ei tutkittu tarpeeksi tarkasti. Esimerkiksi tilanne missä on Poistuvaan yhtiöön kytköksissä oleva pankki. Poistuvan yhtiön lakkauttamisen jälkeen ei tälle pankkia kyetty käyttämään uusissa vakuutuksissa. Myös vakuutukset, joissa pankki oli ollut kiinnittäjänä, oli nyt suljettu pois käyttäjiltä muutoksen jälkeen.

Työn tavoitteet saada Poistuva yhtiö pois näkyvistä käyttöliittymästä sekä se, ettei tälle voisi luoda enää uusia vakuutuksia toteutettiin onnistuneesti. Kuitenkin tällä oli valitettavasti odottamattomia seurauksia.

Tämän työn myötä osaamme olla jatkossa varovaisempia ja lähestyä yhtenäisemmin muutoksia, jotka koskettavat useampaa järjestelmää ja niiden välisiä kytköksiä. Ilman tätä työtä ei esimerkiksi olisi selvinnyt, että Alueyhtiön ryhmän voi asettaa tilaan retired-koodista ilman, että PolicyCenter tarkistaa, onko sen alla olevat aliryhmät tilassa active. Insinöörityössäni käsittelemäni osuus on valmistunut onnistuneesti. Mutta projekti jatkuu yhä tarvittavilla korjauksilla, jotta voidaan varmistaa kaikkien käyttäjien ja vakuutusten toimita nyt ja jatkossa.

Työn aikana opin, kuinka tärkeää on selvittää tarkasti asiat ennen kuin niitä toteutetaan. Tämä tehdään sen takia, ettei tulisi yllätyksiä yhden ryhmän asettamisella eri tilaan useampi käyttäjä ei pääsisi käsiksi vakuutuksiinsa. Sen lisäksi sen, kuinka tärkeää on varmistaa usean järjestelmän ratkaisuissa, että kaikki ovat sopineet yhteisestä linjasta, kuinka ratkaisu toteutetaan.

7 Lähteet

1 Rantala, Jukka & Kivisaari Esko. 2016. Vakuutusoppi.12.m uudistettu painos. Turenki: Hansa Print Oy.

2 Rantala, Jukka & Kivisaari Esko. 2016. Vakuutusoppi.12.m uudistettu painos. Turenki: Hansa Print Oy.sivu 70.

3 Blomberg. 2021. Verkkoaineisto <https://www.bloomberg.com/profile/person/16948296>. Luettu 17.2.2021.

4 Guidewire PolicyCenter Docs. Guidewire sisäinen dokumentointi. Guidewire Software Inc.

5 Guidewire ClaimCenter Docs. Guidewire sisäinen dokumentointi. Guidewire Software Inc.

6 Guidewire BillingCenter Docs. Guidewire sisäinen dokumentointi. Guidewire Software Inc.