



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

**Pirkimų viešinimo proceso automatizavimas su RPA sistema
UiPath**

**Automation of the Procurement Publicity Process With the RPA
System Uipath**

Baigiamasis bakalauro darbas

Atliko: Vytautas Beiga

VU el. p.: vytautas.beiga@mif.stud.vu.lt

Vadovė: asistentė dr. Laura Ringienė

Vilnius
2024

TURINYS

SANTRAUKA.....	5
SUMMARY	6
1. ĮVADAS.....	7
1.1. Temos aktualumas.....	7
1.2. Konfidenciali informacija	7
1.3. Tikslas.....	7
1.4. Uždaviniai.....	7
1.5. Naudojamos sąvokos	7
1.6. Darbo struktūra.....	7
2. RPA ANALIZĖ.....	8
2.1. Robotinis procesų automatizavimas.....	8
2.2. Automatizavimo privalumai.....	8
2.2.1. Laiko taupymas	8
2.2.2. Standartizuojamas procesas	9
2.2.3. Kokybės užtikrinimas.....	9
2.2.4. Darbuotojų kompetencijos gerinimas.....	9
2.3. Automatizavimo trūkumai	10
2.3.1. Informacinių sistemų pokyčiai	10
2.3.2. Informacinių sistemų neveikimai	11
2.4. Procesų automatizavimo programų palyginimas	12
2.4.1. UiPath.....	12
2.4.2. Blue Prism	13
2.4.3. Power Automate	14
2.4.4. Python (su automatizavimo bibliotekomis)	15
2.4.5. Dirbtinio intelekto ir mašininio mokymosi galimybės.....	16
2.4.6. Pasirinkta sistema	17
2.5. „Robotic Enterprise“ karkasas	19
3. ATLIEKAMŲ DARBŲ METODIKA	22
3.1. Pirkimų viešinimo proceso standartizavimas.....	22
3.2. Užduočių susidėjimas iš elektroninių laiškų.....	22
3.3. Duomenų nuskaitymas, apdorojimas ir įvedimas.....	23
3.4. Pirkimų viešinimo procesas	25
3.5. Funkciniai reikalavimai diagramoms.....	27
4. PROCESO IR VIZUALIZACIJOS REALIZAVIMAS.....	29
4.1. Proceso tinkamumo automatizavimui vertinimas.....	29

4.2.	Užduočių gavimas ir susidėjimas	30
4.3.	Sistemų atidarymas ir prisijungimas	32
4.4.	Duomenų nuskaitymas iš sistemų	33
4.4.1.	Duomenų nuskaitymas iš <i>PDS</i> sistemos.....	33
4.4.2.	Duomenų nuskaitymas iš <i>DLX</i> sistemos.....	34
4.4.3.	Duomenų nuskaitymas iš <i>CVPP</i>	35
4.5.	Duomenų sutikrinimas.....	35
4.6.	Duomenų pildymas <i>CVPI</i> ir užduoties uždarymas.....	35
4.6.1.	Informacijos pildymas apie pirkimą	36
4.6.2.	Dalyvių pildymas	36
4.6.3.	Užduoties užbaigimas.....	37
4.7.	Interaktyvių diagramų kūrimas.....	37
4.8.	Proceso kokybės rodiklių palyginimas	40
5.	REZULTATAI IR APIBENDRINIMAS	41
	LITERATŪRA.....	42
	PRIEDAI.....	44

SANTRAUKA

Pagrindinis baigiamojo darbo tikslas – ištirti ir automatizuoti pirkimų viešinimo procesą. Darbo metu buvo sprendžiami šie uždaviniai: nustatyti automatizavimo privalumus ir trūkumus; apžvelgti automatizavimui skirtas programas ir jas palyginti; realizuoti pirkimų viešinimo proceso automatizavimą ir paruošti dinaminę vizualią ataskaitą, kuri leistų stebėti automatizuoto proceso atlikinėjamas užduotis.

Atliekant darbą buvo analizuoti automatizavimo teikiami privalumai ir trūkumai. buvo atliktas keturių skirtingų automatizavimo programų palyginimas: *UiPath*, *Blue Prism*, *Power Automate* ir *Python*. Buvo vertinamos tokios savybės kaip kaina, dokumentacijos ir mokymų, integravimo bei plėtros galimybės. Remiantis surinkta informacija ir poreikiais, buvo pasirinkta tinkamiausia iš lyginamų programų, su kuria buvo tęsiamas darbas. *UiPath* programa buvo toliau tyrinėjama, aptariant jos siūlomas funkcijas ir karkasą, skirtą automatizuoti pasikartojančias verslo operacijas. Baigus analizuoti programą, buvo detalizuotas pirkimų viešinimo procesas ir apibrėžta, ką reikėtų atlikti kiekviename jo etape. Naudojant *UiPath* programą, procesas buvo automatizuotas ir galiausiai sukurtos dinamiškai kintančios vizualizacijos pagal nustatytus funkcinis reikalavimus, padedančios stebėti užduočių būklę, kiekius ir bendrą sutaupytą laiką. Šios vizualizacijos taip pat pasižymi interaktyvumu, nes ant jų paspaudus galima filtruoti norimus duomenis. Darbo pabaigoje buvo įvertinti pasiekti rezultatai bei numatyti ateities veiksmai proceso tobulinimui.

(Raktiniai žodžiai: *RPA#1*, *procesų automatizavimas#2*, *automatizavimo privalumai ir trūkumai#3*)

SUMMARY

The main objective of the thesis is to investigate and automate the procurement publicity process. The objectives of the thesis were: to identify the advantages and disadvantages of automation; to review and compare the applications available for automation; to implement the automation of the procurement publicity process and to create a dynamic visual report that would allow monitoring the tasks performed by the automated process.

Translated with DeepL.com (free version)The work analysed the advantages and disadvantages of automation and compared four different automation programs: *UiPath*, *Blue Prism*, *Power Automate* and *Python*. Features such as cost, documentation and training, integration and expandability were evaluated. Based on the gathered information and the needs, the most suitable of the compared applications was selected and the work continued. The *UiPath* application was further explored by discussing the features it offers and the framework for automating repetitive business operations. Once the analysis of the application was completed, the procurement publicity process was detailed and what should be done at each stage was defined. Using the *UiPath* application, the process was automated and dynamically changing visualizations were created according to the defined functional requirements to monitor the status of tasks, quantities and overall time savings. These visualisations are also interactive, as they can be clicked on to filter the desired data. The work concluded with an evaluation of the results achieved and future ways to improve the process.

(Keywords: RPA#1, process automation#2, the pros and cons of automation#3).

1. ĮVADAS

1.1. Temos aktualumas

Ignitis įmonių grupė turi apie 1,6 milijoną klientų. Kad aptarnautų tokį kiekį klientų, Ignitis Grupė turi apie keturis tūkstančius darbuotojų, kuriems reikia įvairios įrangos bei programų sėkmingam darbo atlikimui. Visiems šiems įrankiams įsigyti yra vykdomi viešieji pirkimai, kad viskas būtų skaidru. Daugelyje pirkimų dalyvauja kelios įmonės, kurios siūlo savo paslaugas ir pagal tam tikrus kriterijus turi būti išrenkamas vienas laimėtojas. Tai taip pat yra viešojo pirkimo dalis. Kadangi šie pirkimai turi būti tinkamai supildyti *Centrinėje Viešųjų Pirkimų Informacinėje Sistemoje (CVPIS)*, juos atlieka viešųjų pirkimų komanda, kuri specializuojasi šioje srityje ir žino kaip duomenys turi būti pateikti įvardintoje sistemoje. Tačiau pildant duomenis, žmogus vis tiek gali padaryti klaidų, t.y. sumoje ne ten padėtas kablelis, įrašytas blogas įmonės pavadinimas, praleistas svarbus informacijos laukelis. Dėl to reikia suieškoti būdų, kaip galima tai išspręsti.

1.2. Konfidenciali informacija

Šio darbo metu aptarinėjamas procesas įtraukia darbuotojų asmeninius duomenis: vardas, pavardė, telefono numeris, elektroninis laiškas. Šie duomenys bus maskuojami, kad būtų apsaugotas šių darbuotojų anonimiškumas.

1.3. Tikslas

Šiuo darbu siekiama ištirti ir automatizuoti pirkimų viešinimo procesą.

1.4. Uždaviniai

1. Nustatyti automatizavimo privalumus ir trūkumus;
2. Apžvelgti automatizavimui skirtas programas ir jas palyginti;
3. Realizuoti pirkimų viešinimo proceso automatizavimą ir dinaminę vizualią ataskaitą;

1.5. Naudojamos sąvokos

- Pirkimų viešinimo procesas – procesas, kurio metu atsakingi asmenys turi supildyti iš pirkėjų gautus duomenis apie pirkimą *CVPIS*.
- Pirkimų viešinimo komanda – darbuotojai ir klientai, kuriems bus automatizuojamas procesas šio darbo metu.

1.6. Darbo struktūra

Baigiamąjį darbą sudaro 6 dalys: įvadas, analitinė dalis, metodinė dalis, įgyvendinimo dalis, rezultatai ir apibendrinimas, literatūros sąrašas.

2. RPA ANALIZĖ

2.1. Robotinis procesų automatizavimas

Robotinis procesų automatizavimas (RPA) yra technologija, naudojanti programinius robotus kasdienių ir rutininių užduočių, kurias tradiciškai atlikdavo žmonės, automatizavimui. Šie robotai gali imituoti daugelį vartotojo sąsajos sąveikų, pavyzdžiui, duomenų įvedimą, operacijų apdorojimą ir įrašų tvarkymą per kelias IT sistemas.

RPA yra ypač efektyvus automatizuojant procesus, kurie yra pagrįsti taisyklėmis, struktūrizuoti ir kartojami. Naudodamos RPA, įmonės gali žymiai padidinti darbo atlikimo efektyvumą, sumažinti klaidų rodiklius ir mažinti operacines išlaidas. Pavyzdžiui, finansų sektoriuje RPA gali automatizuoti tokias užduotis kaip sąskaitų apdorojimas, darbo užmokesčio atnaujinimai ir ataskaitų rengimas.

RPA taikymas neapsiriboja paprastomis užduotimis. Jis taip pat integruojamas su tokiomis technologijomis kaip dirbtinis intelektas ir mašininis mokymasis, kad būtų pagerinti sprendimų priėmimo procesai ir tvarkomos sudėtingesnės užduotys, pavyzdžiui, natūralios kalbos apdorojimas arba prognozuojamoji analitika. Ši integracija leidžia RPA palaikyti platesnį administracinių funkcijų ir pramonės specifikos darbo procesų spektrą, prisidedant prie organizacijų skaitmeninės transformacijos pastangų, kartu išlaisvinant žmogiškuosius darbuotojus, kad jie galėtų skirti daugiau dėmesio strateginėms, kūrybinėms ir orientuotoms į klientus iniciatyvoms [1].

2.2. Automatizavimo privalumai

Automatizavimas gali suteikti daug privalumų priklausomai nuo automatizuojamo proceso tipo ir paskirties. Iš viešinimo proceso yra tikimasi tokių privalumų kaip laiko taupymas, proceso standartizavimas bei kokybės užtikrinimas. Išanalizavus [2]-[5] šaltinius, buvo atrasta keletas privalumų, kuriuos teikia automatizavimas.

2.2.1. Laiko taupymas

Akivaizdžiausia nauda, kurią teikia automatizavimas, yra laiko taupymas. Nors ir gali atrodyti, kad automatizavus procesą daugiau nebereikės jam skirti laiko ir viskas pasidarys savaime, tačiau tai nėra tiesa. Kiekviena užduotis gali turėti tam tikrų išimčių su duomenimis ar jų pildymu, todėl visuomet reikia atsakingo žmogaus, kuris sutikrintų roboto atliktas užduotis ir atliktų tas, kurios turėjo išimčių. Būtent remiantis šia informacija, pirkimų komandos buvo paprašyta į tai atsižvelgti ir skaičiuoti tik tas užduotis, kurios neturi jokių išimčių.

Pagal pateiktus pirkimų komandos duomenis (toliau klientai) (žr. 1 priedą), vieną pirkimo proceso užduotį užtrunka atlikti 40 minučių. Tokių užduočių per metus būna apie 630 (duomenys iš 2022-12-03 – 2023-12-03 laikotarpio). Atlikus skaičiavimus sužinome, kad tai iš viso sudaro 420 darbo valandų arba 1 valandą ir 40 minučių per darbo dieną, monotoniško darbo, kurį būtų galima išnaudoti kitiems, daugiau kūrybiškumo reikalaujantiems darbams.

Panaudojus automatizavimo technologiją, šis laiko taupymas tampa dar įspūdingesnis, nes žmonėms ne tik nebereikės rankomis atlikti viešinimo proceso užduočių, bet ir pačios automatizuoto proceso užduotys bus atliekamos žymiai greičiau. Nesukūrus automatizuoto proceso sunku pasakyti, kiek tiksliai laiko robotas užtruktų atlikti vieną tokio tipo užduotį, tačiau

remiantis jau suprogramuotais panašaus masto ir tipo procesais, viena viešinimo proceso užduotis turėtų būti atliekama per 5 - 10 minučių, priklausomai nuo sistemų greitaveikos. Tai reiškia, kad metinis užduočių skaičius gali būti atliktas žymiai greičiau, sutaupant šimtus darbo valandų.

2.2.2. Standartizuojamas procesas

Norint sėkmingai automatizuoti procesą, neišvengiamas yra vienas žingsnis – proceso standartizavimas. Pagal lietuvių žodyną [6], standartizacija – susitarimų dėl aparatinės ir programinės įrangos optimalios specifikacijos, jų kūrimo ir naudojimo priėmimas; standartų, normų ir taisyklių nustatymas bei taikymas. Procesas laikomas standartizuotu tuomet, kai visi darbuotojai vienareikšmiškai nusprendžia, kaip tas procesas turi būti atliekamas ir kaip turi atrodyti galutinė vykdymo instrukcija. Tai padeda komandoms paprasčiau ir veiksniau bendradarbiauti.

Standartizuotas procesas ne tik palengvina bendradarbiavimą tarp komandų, bet ir užtikrina, kad procesas būtų atliekamas nuosekliai ir efektyviai neatsižvelgiant į asmeninį darbuotojo stilių ar metodą. Tai ypač svarbu didelėse organizacijose, kur skirtingų skyrių ar komandų darbo metodai gali skirtis. Standartizuoti procesai užtikrina, kad visi žingsniai būtų atliekami vienodai, mažinant klaidų tikimybę ir užtikrinant aukštesnę darbo kokybę.

Standartizuotame procese lengviau identifikuoti ir šalinti procesų trūkumus. Kai procesas yra vienodas visose komandose, lengviau pastebėti, kurie jo aspektai yra neefektyvūs arba nereikalingi. Tai leidžia organizacijai greitai reaguoti ir tobulinti procesus, taip pat suteikia galimybę nuolat tobulinti darbo efektyvumą [2].

2.2.3. Kokybės užtikrinimas

Procesuose, kurie yra paremti nuolatinio pasikartojimu, ženkliai išauga žmogiškosios klaidos rizika. Rankinis darbas tokiuose procesuose ne tik sunaudoja daug laiko, bet ir atveria galimybes atsirasti klaidoms dėl darbuotojų dėmesio trūkumo ar nuovargio. Automatizavimas šiuose procesuose leidžia ženkliai sumažinti klaidų tikimybę, kadangi automatizuotos sistemos dirba pagal iš anksto suprogramuotus algoritmus ir yra atsparios žmogiškajai klaidai, nuovargiui bei dėmesio praradimui [3].

Pildant pirkimą per *CVPIS*, atsiranda didelė įvairių klaidų rizika: gali būti neteisingai įvesta pirkimo kaina, klaidingai pagrįstas tiekėjo pasirinkimas, ar netinkamai suvesti tiekėjo rekvizitai. *CVPIS* netikrina, ar pateikti duomenys yra teisingi, ji tik patikrina, ar visi privalomi laukai yra užpildyti. Dėl šios priežasties, pildant naują pirkimą, galima nepastebėti padarytų klaidų. Robotas, skirtingai nei žmogus, tokios klaidos padaryti negali, nes jis tiesiog įrašo duomenis, kuriuos nuskaito iš jam pateikiamos informacijos ir juos apdoroja pagal suprogramuotus algoritmus [4].

Be to, kokybės užtikrinimo procesai gali padėti išlaikyti pirkimų teisingumą ir skaidrumą, kas yra itin svarbu viešųjų pirkimų sektoriuje. Šie procesai leidžia ne tik sumažinti klaidų skaičių, bet ir sutaupyti laiką, kuris anksčiau būdavo skiriamas klaidų taisymui bei duomenų patikrinimui.

2.2.4. Darbuotojų kompetencijos gerinimas

Žiūrint į šiuolaikinius darbo rinkos pokyčius, automatizavimas keičia darbo pobūdį, suteikdamas darbuotojams daugiau galimybių tobulinti savo įgūdžius ir kompetencijas. Kadangi

darbuotojams nereikia kiekvieną dieną virš valandos daryti monotoniško darbo, jie gali daugiau laiko skirti kūrybiškesnėms užduotims, arba tą laiką praleisti saviugdai.

Frontiers straipsnyje [5] aptariamas skaitmeninės transformacijos procesas, kuris darbuotojams suteikia galimybę tobulinti savo įgūdžius atsižvelgiant į technologinius pokyčius. Automatizavimas iškelia naujus reikalavimus darbuotojų kompetencijoms ir mokymuisi, skatindamas darbuotojus tapti lankstesniais ir prisitaikyti prie nuolatos kintančios darbo aplinkos. Šios nuostatos rodo, kad automatizavimas ir skaitmeninė transformacija gali suteikti darbuotojams daugiau galimybių tobulinti savo kūrybiškumą ir profesinius įgūdžius, kartu mažinant rutininio darbo apkrovą.

Gali nutikti taip, kad specifiniuose procesuose tam tikram laikotarpiui atsiranda daug užduočių. Papildomas darbo krūvis tam tikrais laikotarpiais gali sukelti poreikį samdyti daugiau darbuotojų arba versti esamus darbuotojus dirbti viršvalandžius. Automatizavimas leidžia šias problemas išspręsti iš anksto paruošiant procesus, kurie gali būti įjungti reikiamu metu. Tai ne tik padeda išvengti neproporcingai didelio darbo krūvio ir viršvalandžių, bet ir užtikrina, kad darbas būtų atliekamas sklandžiai ir be sutrikimų, net ir esant didesnei darbo apkrovai. Be to, tai leidžia įmonėms lanksčiau valdyti išteklius, optimizuoti darbo procesus ir geriau planuoti darbuotojų poreikius ateityje, taip užtikrinant aukštesnį darbo efektyvumą ir mažesnes eksploataavimo išlaidas.

2.3. Automatizavimo trūkumai

Prieš proceso automatizaciją, reikia nustatyti trūkumus, kurie galėtų kilti automatizavus procesą. Nors trūkumų nėra daug, reikia į juos atsižvelgti, nes jeigu jie dažnai pasikartoja ar sukelia didelius nuostolius kompanijai, tuomet reikia pergaltvoti proceso automatizavimo svarbą. Remiantis [7] ir [8] šaltiniais, pavyko išskirti du trūkumus, dėl kurių gali kilti poreikis pergaltvoti automatizavimo realizavimą.

2.3.1. Informacinių sistemų pokyčiai

Jeigu vyksta dažni informacinės sistemos pokyčiai, neapsimoka automatizuoti proceso, nes dažnas kodo koregavimas gali užimti daug laiko ir neatsipirkti. Todėl, jei įmanoma, reiktų pasirūpinti retesniais sistemos naujinimais arba nesiekti pastovaus naujausios versijos turėjimo. Štai keletas pagrindinių būdų, kaip dažni atnaujinimai gali sutrikdyti šiuos procesus:

- Vartotojo sąsajos pokyčiai - automatizuoti procesai dažnai remiasi konkrečiomis sąsajomis, pavyzdžiui, norint atlikti užduotis, programinės įrangos sąsaja. Kai atnaujinimas pakeičia šias sąsajas – tai gali būti mygtukų, meniu ar kitų sąsajos elementų padėtis – automatizuoti veiksmai gali nesurasti šių elementų, dėl ko gali kilti klaidų ar visiškai sutrikti procesas.
- Funkcionalumo modifikacijos - atnaujinimai gali keisti programinės įrangos funkcijas, nuo kurių priklauso automatizuoti procesai. Jei programinės įrangos atnaujinimas modifikuoja, funkcijos veikimo principą arba ją visiškai pašalina, automatizacijos scenarijus gali neveikti taip, kaip buvo numatyta ir dėl to gali būti reikalingas automatizacijos logikos atnaujinimas.
- Didesnė priežiūros našta - su dažniais atnaujinimais atsakingiems asmenims reikia skirti daugiau išteklių nuolat stebėti ir atnaujinti automatizacijos scenarijus, kad būtų galima

sėkmingai dirbti įvykus pokyčiams. Tai ne tik padidina operacinius kaštus, bet ir nukreipia techninius išteklius nuo kitų vertingų užduočių.

Pirkimų viešinimo proceso atveju atnaujinimai nebus dažni, tačiau kartą į ketvirtį juos gali tekti padaryti. Didžiausi nuostoliai gali įvykti tuomet, kai bus diegiami naujinimai *CVPIS*, nes tai yra išorinė sistema. Kadangi tai yra išorinė sistema, nėra suteikiama jokia testinė aplinka, iš anksto nėra pranešama apie artėjančius naujinimus ir niekas nesiderina su automatizavimo programuotojais apie artėjančių naujinimų laiką. Taigi, atsiranda toks trūkumas, kad atėjus sistemos pokyčiui, priklausomai nuo jo masto, robotas nedirbs tam tikrą laiko dalį, kol nebus perdarytas pagal naują informacinės sistemos versiją.

Taip pat, priklausomai nuo informacinės sistemos veikimo principo ir automatizavimo pobūdžio, gali tekti daryti metinius pakeitimus. Viešinimo procesui kiekvienais metais reiktų nurodyti naujas sutarčių vietos nuorodas dokumentų pildymo sistemoje (DLX), nes keičiasi jų aplankai, o kartu ir nuorodos į juos. Tai nėra didelis trūkumas, tačiau reikia suprasti, kad kiekvienais metais programuotojams teks skirti mažą dalį savo darbo laiko šioms pokyčiams atlikti. Šis laiko kiekis gali skirtis priklausomai nuo to, kiek yra automatizuotų procesų, kuriems reikalingi metiniai pokyčiai.

2.3.2. Informacinių sistemų neveikimai

Kartais gali nutikti taip, kad informacinė sistema neveikia tik dalinai ir ta neveikianti dalis yra suprogramuota robote. Tuomet sistema veikia ne pilnu pajėgumu ir automatizuoti procesai negali atlikti užduočių sėkmingai, nors yra kitas būdas atlikti tokio tipo užduotį. Tokiu atveju, priklausomai nuo neveikimo laiko, gali būti neišnaudojamos mokamos licencijos ir nauda įmonei nebus generuojama, nes darbuotojai turės atlikinėti užduotis rankomis. Tokie sistemos trikdžiai gali kilti dėl įvairių priežasčių: nuo programinės įrangos klaidų iki fizinių serverių gedimų ar netinkamos sistemos priežiūros.

Neveikiančių sistemų pasekmės gali sukelti įvairių problemų. Viena vertus, tai tiesiogiai veikia įmonės veiklos efektyvumą – darbuotojai, turėdami atlikti užduotis rankiniu būdu, vėluoja atlikti kitas svarbias užduotis, o tai gali sukelti vėlavimus tiekimo grandinėje, klientų nepasitenkinimą ar net finansines baidas. Kita vertus, neveikiantys automatizavimo sprendimai didina įmonės išlaidas, nes mokamos licencijos ir kitos investicijos į automatizavimo įrangą bei programinę įrangą nėra efektyviai išnaudojamos.

Be tiesioginių finansinių nuostolių, yra ir kitų neigiamų aspektų. Pavyzdžiui, gali nukentėti darbuotojų moralė, matant, kiek daug pastangų įdėta į automatizavimo sistemas, kurios nėra patikimos ar nesugeba atlaikyti kritinių momentų. Taip pat gali kilti įmonės reputacijos rizika, ypač jei neveikiančios sistemos sukelia klientų nepasitenkinimą ar net praranda svarbius verslo partnerius.

Norint išvengti ar bent jau sumažinti šias problemas, būtina reguliariai atnaujinti ir tikrinti visus informacinių sistemų komponentus, investuoti į patikimus atsarginių sistemų ir duomenų atkūrimo sprendimus bei mokyti darbuotojus, kaip elgtis įvairių trikdžių atveju. Taip pat svarbu turėti aiškų planą, kaip elgtis sistemos neveikimo atveju, kad būtų galima greitai grąžinti operacijas į įprastą vėžes ir užtikrinti, jog įmonės veikla būtų kuo mažiau paveikta.

2.4. Procesų automatizavimo programų palyginimas

Automatizacijos programos tampa vis aktualesnės šiomis dienomis siekiant pagerinti procesų kokybę, sumažinti jų atlikimo kaštus bei pagreitinti užduočių atlikimo laiką. Dėl šios priežasties yra daugybė automatizacijos įrankių, kurie palaiko skirtingus programinės įrangos ir techninės įrangos reikalavimus. Taip pat turi skirtingus išplėtimo, priežiūros, finansinių įsipareigojimų ir kitus parametrus. Dėl šių priežasčių buvo nuspręsta palyginti keturias skirtingas automatizavimui naudojamas programas siekiant išsiaiškinti, kurios iš jų siūlomų galimybių labiausiai atitinka reikalavimus automatizacijai [9], [10], [11]. Šioje analizėje buvo pasirinktos *UiPath*, *Blue Prism*, *Power Automate* ir *Python* (su automatizavimo bibliotekomis) programos.

2.4.1. UiPath

UiPath yra pasaulinė programinės įrangos įmonė, kuri kuria platformą robotizuotai procesų automatizacijai (RPA). Platforma leidžia organizacijoms automatizuoti taisyklėmis pagrįstus procesus, integruojant programas ir sistemas be išsamių programavimo žinių. Šiuo metu *UiPath* yra viena iš populiariausių ir plačiausiai naudojamų automatizavimo programų [12].

UiPath pagrindiniai privalumai [13]:

- Naudojimosi patogumas – *UiPath* turi intuityvų dizainą, kuris yra lengvai suprantamas net ir tų, kurie neturi gilių programavimo žinių. Ši programa naudoja vartotojo sąsają automatizavimo žingsnių kūrimui, kas žymiai sumažina mokymosi kreivę (angl. *learning curve*) ir leidžia greičiau realizuoti automatizavimo projektus.
- Išplėtimas – *UiPath* sukurta taip, kad galėtų būti pritaikoma tiek mažoms operacijoms, tiek įmonių lygio poreikiams. Programa papildomai siūlo įrankį, kuris padeda valdyti turimus procesus ir jų užduotis. Dėl to yra lengva plėsti automatizuotus procesus ir nereikia jaudintis dėl centralizuoto valdymo sprendimų.
- Integravimas – programa siūlo didelį bibliotekų pasirinkimą, kuris leidžia lengvai bendrauti su populiariomis aplikacijomis tokiais kaip: *Microsoft Excel*, *Microsoft Outlook*, *SAP* ir t.t. Galima naudoti aplikacijų programavimo sąsają (angl. *Application Programming Interface*), kuri kartais palengvina atlikti sudėtingus veiksmus. Taip pat yra specialios funkcijos, kurios leidžia prisijungti prie duomenų bazių ir jose atlikti įvairius veiksmus, kurių gali prireikti automatizacijai, o jeigu vis dėl to reikia specifinio funkcionalumo, tuomet galima naudoti integraciją su įvairiomis programavimo kalbomis: *C#*, *VB.NET*, *Power Shell* ir t.t. Visos šios galimybės leidžia atlikti automatizaciją įvairiose informacinėse sistemose.
- Bendruomenė ir priežiūra – šios programos forumai ir dokumentacija yra labai vertinami RPA bendruomenėje dėl savo išsamumo ir patogios pateikimo formos. *UiPath* bendruomenė yra labai aktyvi, dėl to galima rasti daug informacijos oficialiame forume *UiPath Community Forum*. O pati *UiPath* oficiali dokumentacija teikia išsamias ir detalizuotas gaires apie visus *UiPath* programinės įrangos aspektus, nuo diegimo ir pradmenų iki pažangių funkcijų, tokių kaip dirbtinio intelekto integracija. Dokumentacija yra struktūruota taip, kad padėtų tiek naujiems vartotojams, tiek patyrusiems vartotojams, ieškantiems pažangių sprendimų būdų.

Nors programa turi daug teigiamų savybių, yra svarbu paminėti, jog ji turi ir trūkumų, kurie gali būti priežastis nesirinkti *UiPath*.

- Kaina – vienas iš pirmųjų barjerų, kuris gali neleisti pasirinkti *UiPath* programos yra jos kaina. Priklausomai nuo licencijos rūšies, kaina gali atrodyti gan didelė mažoms

įmonėms arba paprastiems asmenims ir dažnai atvejais neatsipirkti. Tai apima ne tik pradinę kainą, bet ir nuolatinės išlaidas, susijusias su atnaujinimais ir priežiūra.

- Sudėtingumas vykdant pažangias automatizacijas – *UiPath* yra draugiškas naudotojui, kai reikia automatizuoti bazinės užduoties, jo pažangesnės funkcijos gali reikalauti gilesnių programavimo ir sistemų integravimo žinių. Sudėtingų automatizavimo procesų programavimas dažnai reikalauja papildomo mokymo ar specializuotų darbuotojų samdos.
- Priklausomybė nuo skaitmeninės infrastruktūros – *UiPath* veikimas stipriai priklauso nuo skaitmeninių aplinkų stabilumo ir konfigūracijos, kuriose jis automatizuojamas. Atnaujinimai integruotose sistemose ar platformose gali sutrikdyti esamas automatizacijas. Ši priklausomybė reiškia, kad įmonėms reikia palaikyti tvirtą informacinių technologijų infrastruktūrą ir būti pasiruošusioms reguliariai ją atnaujinti bei spręsti problemas.

Taigi, nors *UiPath* yra plačiai naudojama programa ir turi daug funkcionalumų, tokių kaip, lengva integracija bei išsami dokumentacija. Kita vertus, ji turi ir savo trūkumų – tokių kaip didelė licencijų kaina, jeigu norima visų funkcijų, bei išaugęs programavimo sudėtingumas norint automatizuoti sudėtingesnius procesus.

2.4.2. Blue Prism

Blue Prism yra kita iš lyderiaujanti programa automatizavimo srityje, kuri padeda automatizuoti užduotis įvairiose informacinėse sistemose. Kaip ir *UiPath*, *Blue Prism* padeda įmonėms automatizuoti pasikartojančias užduotis bei turi savo unikalių savybių ir iššūkių.

Blue Prism pagrindiniai privalumai:

- Stiprios saugumo funkcijos - *Blue Prism* sukurta akcentuojantis į saugumą, įskaitant galutinį (angl. *end to end*) šifravimą ir audito žurnalus, kurie užtikrina, kad visa programinėje įrangoje tvarkoma informacija išliktų saugi ir sekama. Tai ypač tinka pramonės šakoms, kurios tvarko jautrią informaciją.
- Išplėtimas – žinoma dėl savo išplėtimo ir patvarumo galimybių įmonėse, leidžiančio vienu metu valdyti didelį procesų skaičių įvairiuose organizacijos skyriuose be našumo netekimo.
- Integravimas – panašiai kaip ir *UiPath*, *Blue Prism* puikiai integruojasi su įvairiomis duomenų bazėmis, skirtingomis aplikacijų programavimo sąsajomis bei IT sistemomis. Tačiau reikia paminėti, kad integravimas su aplikacijų programavimo sąsajomis per *Blue Prism* yra sudėtingesnis, nes reikia atlikti keletą papildomų žingsnių dėl jos architektūros.
- Bendruomenė ir priežiūra – *Blue Prism* vartotojai yra aktyvūs ir dėl to galima rasti daug informacijos įvairiuose forumuose. Taip pat yra parengtos kokybiškos dokumentacijos įvairiems *Blue Prism* įrankiams, kurios padeda su jais lengviau susipažinti.

Kaip ir analizuojant *UiPath*, nors sistemos privalumai yra labai svarbūs, tačiau renkantis programą ne ką mažiau svarbūs ir jos trūkumai:

- Aukštas mokymosi kreivės lygis – *Blue Prism* gali turėti aukštesnę mokymosi kreivę, lyginant su kitomis RPA programomis, dėl sudėtingesnės sąsajos ir programavimo reikalavimų. Paprastai reikalaujama, kad vartotojai turėtų programavimo patirties, kas gali būti neidealus pasirinkimas įmonėms, ieškančioms labiau prieinamo įrankio.

- Kaina – *Blue Prism* diegimas gali būti santykinai brangesnis dėl licencijų ir reikalingos infrastruktūros, kad būtų palaikytos jos patvarios galimybės. Tai gali būti reikšminga kliūtis mažesnėms organizacijoms arba tiems, kurių biudžetai yra riboti.
- Retesni programos atnaujinimai – palyginus su kitais įrankiais, *Blue Prism* atnaujinimai yra vykdomi rečiau ir dėl to naujausios technologijos ar funkcijos gali pasiekti vartotojus vėliau.

Taigi, nors *Blue Prism* turi daug privalumų, kurie yra naudingi didelėms įmonėms, tačiau mažesnėms įmonėms ar fiziniams asmenims gali būti sunku išnaudoti šios programos siūlomas galimybes, kad atsipirktų prenumeruotos licencijos.

2.4.3. Power Automate

Microsoft Power Automate (toliau *Power Automate*) yra procesų automatizavimo programa sukurta *Microsoft* kompanijos skirta automatizuoti darbo eigas tarp programų ir paslaugų *Microsoft* aplinkoje. Šia programa atliekamos automatizacijos padeda supaprastinti pasikartojančias užduotis ir procesus, automatiškai atliekant veiksmus pagal vartotojo apibrėžtus trigerius (angl. *Triggers*) ir sąlygas. *Power Automate* gali būti naudojamas įvairioms paskirtims, įskaitant duomenų rinkimą, failų sinchronizaciją, pranešimus ir verslo procesų automatizavimą.

Žinoma, norint įvertinti sistemos galimybes, reikia įvertinti jos privalumus bei trūkumus. Taigi, pagrindiniai *Power Automate* privalumai [14]:

- Integravimas *Microsoft* erdvėje – *Power Automate* lengvai integruojasi su *Microsoft Office 365* (toliau *Office 365*) siūlomomis programomis, kas yra didelis privalumas vartotojams jau esantiems šioje ekosistemoje. Ši integracija užtikrina sklandų duomenų perėjimą tarp tokių platformų kaip *SharePoint*, *Excel*, *Outlook* efektyviai valdant jų darbo eigas.
- Naudojimosi patogumas – platforma žinoma dėl savo paprastumo, įskaitant intuityvią sąsają ir nereikalaujančią išplėstinių programavimo įgūdžių. Tai daro ją prieinamą platesniam vartotojų ratui, nuo pradedančiųjų iki pažengusiųjų vartotojų, siekiančių įgyvendinti nuo paprastų iki vidutiniškai sunkių darbo procesų.
- Platus šablonų pasirinkimas – siūlo daugybę iš anksto paruoštų šablonų, kurie supaprastina naujų automatizuotų procesų kūrimą, leidžiant vartotojams greitai nustatyti ir diegti automatizacijas su minimaliomis konfigūracijomis.
- Stiprios saugumo funkcijos – kadangi *Power Automate* priklauso *Microsoft* ekosistemai, ji perima visus aukšto saugumo ir atitikties standartus, todėl yra patikimas pasirinkimas organizacijoms, besirūpinančioms duomenų vientisumu ir saugumu. Tačiau tai, kad programoje daug dėmesio skiriama saugumui, nepaneigia poreikio naudotojams kurti savo darbą atsižvelgiant į saugumą, ypač integruojant išorines sistemas ir tvarkant jautrią informaciją.
- Bendruomenė ir priežiūra – *Power Automate* siūlo išsamią ir nuolat atnaujinamą dokumentaciją, kuri atspindi naujas funkcijas ir geriausias praktikas. Kadangi tai yra *Microsoft* produktas, dokumentacija yra kokybiška ir padaryta taip, kad su *Power Automate* programa galėtų dirbti žmonės, kurie neturi didelių programavimo žinių. Kadangi daug žmonių naudoja *Microsoft* paslaugomis, forumai taip pat yra gausūs informacijos ir juose galima rasti atsakymus į daugelį klausimų [15].

Nors *Power Automate* ir yra draugiška programa naudotojui ir sugeba lengvai integruotis *Office 365* aplinkoje, tačiau ji turi savų trūkumų:

- Didelių duomenų apdorojimas – kai kurie vartotojai praneša, kad tvarkant didelius duomenų kiekius ar sudėtingus procesus, kyla veikimo sutrikimų. Tai apima vykdymo vėlavimus, vykdymo atšaukimus dėl per ilgo apdorojimo (angl. *timeout*), ypač kai procesai reikalauja realaus laiko duomenų apdorojimo. Kartais gali neapdoroti sudėtingesnių procesų dėl resursų trūkumo, nes pagal pradinis nustatymus yra apribotas maksimalus galingumas.
- Integravimas ne *Microsoft* erdvėje – nors *Power Automate* lengvai integruojasi *Microsoft* erdvėje, tačiau su kitomis sistemomis integracija gali būti labai sudėtinga arba iš viso neįmanoma. Tai yra didelė problema įmonėms ar fiziniams asmenims, kurių visa darbo infrastruktūra nepastatyta ant *Microsoft* aplinkos. Tai gali riboti šios programos funkcionalumą ir padidinti realizavimo sudėtingumą.
- Kaina – *Power Automate* priklausomai nuo turimo *Office 365* plano, gali ateiti kartu su juo. Tai yra patogus įmonėms, kurios jau įsitvirtinusios *Microsoft* aplinkoje ir gali papildomai išnaudoti *Office 365* teikiamas galimybes į kurias įeina *Power Automate*, tačiau kad gautų papildomas funkcijas reikia vis tiek pirkti papildomą licenciją, kuri kainuoja papildomai. Dėl šios priežasties ši programa gali būti sunkiai prieinama tiems, kuriems reikalingos pažangios funkcijos, bet negalintiems pateisinti papildomų išlaidų.
- Ribotos be ryšio (angl. *offline*) galimybės – skirtingai nuo kai kurių RPA įrankių, *Power Automate* daugiausia veikia su ryšiu (angl. *online*). Jos priklausomybė nuo debesies ryšio reiškia, kad vartotojai gali susidurti su apribojimais scenarijuose, kur būtinas automatizavimo be ryšio poreikis arba kur interneto ryšys yra nepatikimas.

Taigi, nors *Power Automate* yra paprastas naudoti ir gali būti lengvai prieinamas, jeigu iš anksto yra naudojama *Microsoft* ekosistema, tačiau gali nepatenkinti lūkesčių, kai norima automatizuoti didesnius procesus ar apdoroti didesnius duomenų kiekius bei automatizaciją atlikti ne tik *Microsoft* aplinkoje.

2.4.4. Python (su automatizavimo bibliotekomis)

Python naudojimas robotiniam procesų automatizavimui turi kelis privalumus ir trūkumus, kurie padaro jį geru pasirinkimu vieno tipo projektams, tačiau ne tokiu patogiu kito tipo.

Kad būtų galima suprasti, kuriose vietose *Python* yra patogus naudoti iš pradžių apžvelgsime jo privalumus:

- Integravimas – kadangi *Python* ekosistema remiasi atvirojo kodo bibliotekomis, ji turi daug įvairių funkcijų atliekančių įrankių, kurie gali padėti dirbti su vartotojo sąsaja, duomenų bazėmis, aplikacijų programavimo sąsajomis, įvairaus tipo duomenimis ir t.t. Keletas populiarių bibliotekų pavyzdžių: *Selenium*, *Pandas*, *requests*, *BeautifulSoup*. Šie įrankiai labai palengvina užduotis, kurios dažnai yra automatizuojamos RPA scenarijuose, padarydami *Python* labai naudingą pasirinkimą kūrėjams.
- Kryžminės platformos suderinamumas – *Python* iš prigimties yra tinkama įvairioms operacinėms sistemoms, o tai reiškia, kad šia programavimo kalba parašyti skriptai gali būti diegiami *Windows*, *macOS* ir *Linux* aplinkose su minimaliais pakeitimais. Ši universalumo savybė gali būti svarbi įmonėms, veikiančioms įvairialypėje kompiuterinėje aplinkoje.
- Bendruomenė ir priežiūra – *Python*, turi labai didelę bendruomenę. Ši bendruomenė ne tik padeda kodo klausimais, bet ir įvykus įvairiems sistemoms sutrikimams dėl kurių gali neveikti sukurti skriptai. Taip pat populiarios bibliotekos, kurios palengvina darbą,

dažnai turi kokybišką dokumentaciją, kas pagreitina darbą. Šis bendruomenės pagrindu veikiantis modelis skatina nuolatinį tobulėjimą ir inovacijas.

- Kaina – kadangi tai yra atvirojo kodo kalba, daugelis funkcionalumų yra nemokami, o tai sumažina automatizavimo sprendimų kūrimo ir mastelio didinimo išlaidas. Tai gali būti lemiamas bruožas leidžiantis mažoms įmonėms ir startuoliams įgyvendinti automatizavimo technologijas be reikšmingų investicijų.

Tačiau, kaip ir kiekviena programa, kuri turi savo privalumus, *Python* taip pat turi ir keletą rimtų trūkumų, kurie gali neleisti pasirinkti šio varianto:

- Priežiūros iššūkiai – jeigu su *Python* programa yra sukurta daug automatizavimo skriptų, jie gali tapti sunkiai tvarkomi. Be struktūrizuotų palaikymo ir priežiūros sistemų, kurias siūlo kitos RPA programos, gali būti sudėtinga valdyti ir atnaujinti didelį kiekį skriptų. Tai gali tapti daug resursų reikalaujanti ir į klaidas linkusi užduotis.
- Aukštas mokymosi kreivės lygis – *Python* yra aukšto lygio programavimo kalba, o tai reiškia, kad išnaudoti visas jos automatizavimo galimybes, reikia žmogaus, kuris turi aukštą supratimą apie pačią programavimo kalbą. Išmokti šių įgūdžių užtrunka tikrai ilgiau, nei naudotis kitomis RPA programomis, todėl gali kainuoti daugiau laiko. Kadangi nėra jokios sukurtų skriptų valdymo sistemos, reikia turėti žinių, kaip tai suorganizuoti. Kad tai būtų pasiekta sėkmingai, reikia turėti žinių ne tik apie pačią programavimo kalbą, bet ir apie sistemą ar architektūrą ant kurios bus dirbama.
- Išplėtimas – nors *Python* tinka plačiam automatizavimo užduočių spektrui, jis ne visada gali būti geriausias pasirinkimas dideliems automatizavimo projektams. Šia kalba parašyti skriptai gali tapti sunkiai palaikomi ir sunkiai plečiami, ypač lyginant su specializuotomis RPA programomis, kurios yra skirtos patogiam plėtimui įmonėse.

Taigi, *Python* kalba galima pigiai automatizuoti užduotis įvairiose operacinėse sistemose su patogiu integravimu daugelyje informacinių sistemų. Tačiau taip pat reikalauja gilių programavimo žinių ir laiko, kad būtų sėkmingai kuriami ir valdomi nauji skriptai, ypač, kai jų skaičius didėja.

2.4.5. Dirbtinio intelekto ir mašininio mokymosi galimybės

Pastaruoju metu dirbtinis intelektas (DI) ir mašininis mokymasis (MM) smarkiai pasistūmėjo į priekį, o tai turėjo didelį poveikį automatizavimo programoms, tokioms kaip *UiPath*, *Blue Prism*, *Power Automate* ir *Python*. Šios technologijos taip pat stengiasi integruoti dirbtinį intelektą į procesų automatizavimą. DI sustiprina robotinę procesų automatizaciją, suteikdamas galimybę automatizuoti sudėtingesnes ir kognityvines užduotis, kurios reikalauja supratimo, sprendimų priėmimo ir mokymosi iš nestruktūrizuotų duomenų. Kadangi visų automatizavimo programų paskirtis panaši, tai normalu, kad dirbtinio intelekto ir mašininio mokymosi pritaikymas taip pat ne daug kuo skiriasi nuo vienas kito. Galbūt realizavimo eiga ir skiriasi, tačiau norimas pasiekti panašus rezultatas, plačiau apie kiekvieno iš šių programų galimybes mašininiam mokymui ir DI:

- *UiPath* - yra žinoma dėl savo DI funkcijų, tokių kaip natūralios kalbos apdorojimas ir mašininis mokymasis, kurie leidžia naudotojams įgyvendinti sudėtingas integracijas ir automatizuoti aukšto lygio sprendimų priėmimo procesus. *UiPath* siūlo DI įrankį *Document Understanding*, kuri ištraukia, interpretuoja ir apdoroja duomenis iš įvairių dokumentų, tokių kaip sąskaitos faktūros, formos ir el. laiškai. Ši funkcija naudoja mašininio mokymosi modelius dokumentų turinio supratimui ir kategorizavimui,

leidžiant automatizuoti sudėtingas duomenų tvarkymo užduotis, kurios paprastai reikalauja žmogaus įsikišimo [16].

- *Blue Prism* - savo DI ir MM galimybes skatina per *Decipher* programą, kuri padeda tvarkyti ir struktūrizuoti nestruktūrizuotą informaciją, pavyzdžiui, iš įmonių dokumentų. Panašiai kaip *UiPath Document Understanding*, ši galimybė leidžia greitai ir tiksliai apdoroti didelius dokumentų kiekius, taip padidinant efektyvumą.
- *Power Automate* - naudoja *Microsoft AI Builder* įrankį, kuris integruoja dirbtinį intelektą į automatizavimo procesus, leidžiantį lengvai kurti DI modelius, skirtus formų analizei, veido atpažinimui, ar sentimentų analizei. Praktinis panaudojimo atvejis būtų naudoti *Microsoft AI Builder* formų apdorojimui, kur įrankis gali identifikuoti ir ištraukti informaciją iš formų ir automatiškai įvesti šiuos duomenis į kitas sistemas, pavyzdžiui, klientų santykių valdymo (CRM) įrankį [17].
- *Python* - dėl savo lankstumo, yra viena iš pagrindinių platformų kurti pasirinktinius DI sprendimus. Vartotojai gali naudoti *Python* kartu su bibliotekomis kaip *TensorFlow* ar *PyTorch* kognityvinėms užduotims, tokioms kaip duomenų klasifikacija ar analizė, automatizuoti. Pavyzdžiui, *Python* skriptas, integruotas su *TensorFlow*, galėtų analizuoti klientų atsiliepimų duomenis ir klasifikuoti nuotaikas kaip teigiamas, neigiamas arba neutralias. Šie duomenys vėliau galėtų būti naudojami automatizuotiems atsakymams ar klientų aptarnavimo išteklių prioretizavimui.

Kaip matome, visos šios RPA programos turi savo įrankius mašininio mokymosi ir dirbtinio intelekto panaudojimui. *UiPath* ir *Blue Prism* yra patogios, kai norima nuskaityti dokumentus su nestruktūrizuotais duomenimis ir juos gražiai sudėti į jiems tinkamas vietas. *Power Automate* su savo įrankiais labiau tinkama pagalbai kuriant naujus procesus, o *Python* būtų galima pritaikyti visiems anksčiau minėtiems atvejams, tačiau vienas iš išskirtinių jo bruožų būtų, sentimentų analizė.

2.4.6. Pasirinkta sistema

Kaip ir buvo aptarta anksčiau, kiekviena automatizavimo programa turi savo privalumų ir trūkumų, tad renkantis labiausiai tinkamą variantą gali būti painu. Buvo sukurta lentelė, kurioje *UiPath*, *Blue Prism*, *Power Automate* ir *Python* yra lyginamos pagal tuos pačius parametrus: išplėtimo galimybes, integravimą, bendruomenę ir priežiūrą, kainą bei kam geriausiai yra tinkamos. Šiuos duomenis matome 2.1 lentelėje.

2.1 lentelė Pagrindinių funkcijų palyginimas

	UiPath	Blue Prism	Power Automate	Python
Išplėtimas	Lengvai praplečiama, tinkama naudoti fiziniams asmenims bei didelėms ar mažoms įmonėms.	Įmonėms didelio masto projektams kurti ir jų tvirtumui palaikyti.	Gali būti lengvai plečiamas Microsoft erdvėje.	Gali būti labai gerai plečiamas, jeigu pastatytas ant tinkamos architektūros.
Integravimas	Platus integracijų su įmonių sistemomis pasirinkimas.	Stiprios integravimo galimybės, ypač įmonių vidinėse aplinkose.	Lengvas integravimas su Microsoft produktais, tačiau beveik	Integravimas universalus, priklausomai nuo to kokią automatizavimo

			neįmanomas kitose aplinkose.	biblioteką yra nuspręsta naudoti.
Bendruomenė ir priežiūra	Didelė bendruomenė, priežiūrai išsami dokumentacija bei forumai.	Kokybiškas dokumentavimas ir įsitvirtinę vartotojai, kurie padeda atsakyti į klausimus, tačiau viešai prieinamų mokymų nėra daug.	Didelė bendruomenė, nes yra daug Microsoft vartotojų, plati dokumentacija.	Labai didelė programuotojų bendruomenė su plačiu dokumentacijos pasirinkimu.
Kaina	Prenumeruojamos licencijos, jų kaina priklauso nuo jų tipo, skaičiaus bei tikėjo. Taip pat Laisvai prieinama visuomeninė (angl. <i>community</i>) versija, kuri skirta besimokantiems ir mažoms įmonėms.	Prenumeruojamos licencijos, dėl kainos reikia tartis su licencijų tiektėjais, dažnai būna, kad prenumeruojama mažiausiai 10 licencijų.	Gali būti naudojamas bendrai su Office 365 programa, jeigu pasirinktas tinkamas planas, papildomas mėnesinis mokestis už „Premium“ galimybes.	Naudojimas nemokamas, kaina priklauso nuo naudojamos architektūros ir automatizacijos aplinkos sukūrimo.
Geriausiai tinka	Fiziniam asmeniui, visų dydžių verslams ir įmonėms.	Didelėms įmonėms, kurioms reikia didelių automatizavimo sprendimų.	Verslams, kurie jau naudoja Microsoft produktus.	Kūrėjams ir komandoms su programavimo žiniomis, siekiantiems pritaikyti automatizavimo sprendimus.

Iš anksčiau minėtų automatizavimo įrankių, šiam procesui automatizuoti buvo pasirinkta *UiPath* programa, kuri naudoja VB.NET programavimo kalbą. Ji geriausiai atitinka reikiamus kriterijus ir taip pat yra viena iš populiariausių ir plačiausiai naudojamų automatizavimo programų.

Taip pat *UiPath* siūlo intuityvią vartotojo sąsają ir galingą funkcionalumą, pritaikytą tiek pradedantiesiems, tiek pažengusiems vartotojams. Įvairūs *UiPath* siūlomi įrankiai, tokie kaip *Orchestrator*, *Robot* ir *Studio*, leidžia kurti sudėtingas automatizavimo schemas, kurios gali apimti įvairias sistemas ir aplikacijas. Pagrindiniai du komponentai tai yra *UiPath Studio* ir *UiPath Orchestrator*. *UiPath Studio* yra skirta procesų automatizavimui, kodo rašymui, trikdžių šalinimui (angl. *debugging*) ir daug kitų užduočių, kurios yra susijusios su automatizuoto proceso kodu. O *UiPath Orchestrator*, tai yra įrankis, skirtas kontroliuoti sukurtų procesų užduotis bei sekti jų statusus, kiekius ir gaunamas klaidas. Papildomai *UiPath Orchestrator* gali būti naudojamas, kaip centrinė kintamųjų ir prisijungimo duomenų talpinimo vieta, kurią gali pasiekti visi procesai. Tai palengvina procesų pokyčių atlikimą, jeigu tokie nutinka.

Kad *UiPath* veiktų sklandžiai, reikalinga tinkama infrastruktūra, įskaitant atskirus serverius ir produkcines licencijas. Licencijų kiekis turėtų būti suderintas su automatizuotų procesų skaičiumi, kadangi kiekvienai licencijai vienu metu gali būti priskirtas tik vienas procesas. Tai reiškia, kad didėjant automatizuotų procesų kiekiui, gali prireikti papildomų licencijų.

Be to, svarbu atkreipti dėmesį į nuolatinį mokymą ir vartotojų palaikymą. *UiPath* siūlo išsamius mokymo kursus ir bendruomenės forumus, kur nauji vartotojai gali gauti pagalbą ir patarimus. Tai ypač svarbu, norint užtikrinti, kad visi įmonės darbuotojai galėtų efektyviai naudoti šią programinę įrangą ir pasinaudoti jos teikiamais privalumais.

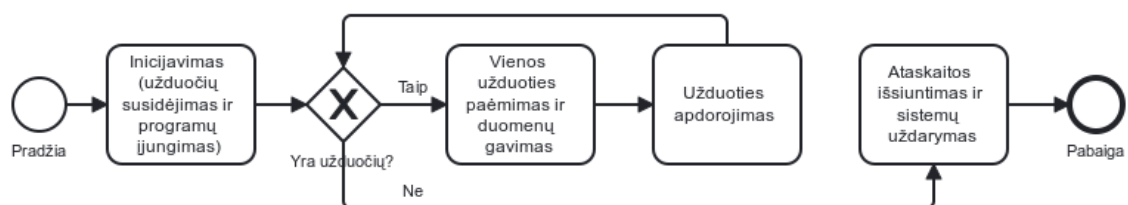
Taip pat *UiPath* siūlo *Robotic Enterprise* karkasą (angl. *framework*), kuris yra specialiai sukurtas verslo procesų automatizavimui, kai procesas turi daug užduočių. Daugiau apie šį karkasą yra papasakota sekančiame skyriuje.

2.5. „Robotic Enterprise“ karkasas

Pirkimų viešinimo proceso automatizavimo realizavimui bus naudojamas *Robotic Enterprise* karkasas (angl. *framework*) (toliau bus vadinamas RE karkasu). Šis RE karkasas yra tinkamas automatizuoti sudėtingus, transakcijomis/užduotimis pagrįstus procesus, kuriuose svarbu patikimumas ir plečiamumas. Žemiau aprašyta keletas vietų, kuriose RE karkasas gali būti veiksmingai naudojamas [18]:

- Transakcijomis pagrįsti procesai – RE karkasas yra sukurtas taip, kad sklandžiai apdorotų didelį kiekį transakcijų, todėl yra idealus procesams, susijusiems su pasikartojančiomis užduotimis, pvz.: duomenų įvedimu iš įvairių šaltinių (eilučių, duomenų bazių, informacinių sistemų ir t.t.) į sistemas ar internetines aplikacijas. RE karkasas tai atlieka kiekvieną transakciją paimdamas atskirai, ją apdorojamas ir tvarkydamas jos išimtis.
- Klaidų valdymas ir proceso atstatymas (angl. *recovery*) – RE karkase yra suprogramuoti klaidų valdymo ir atkūrimo mechanizmai, kurie leidžia efektyviai valdyti išimtis ir tęsti transakcijų apdorojimą įvykus klaidoms. Tai apima paleidimus iš naujo, kai įvyksta inicijavimo arba apdorojimo klaidos, prieš užfiksuojant klaidą ataskaitoje ir nutraukiant procesą arba jį tęsiant.
- Plečiamumas – dėl savo modulinio dizaino RE karkasas yra draugiškas mastelio praplėtimui, leisdamas pridėti papildomų funkcijų ar pakeitimų neveikiant esamam proceso kodui. Tai palengvina didelio masto diegimų valdymą organizuojant procesus į valdomas darbo eigas (angl. *workflows*), kurias galima atskirai ištestuoti ir prižiūrėti.

Supaprastinta RE karkaso diagrama pateikta 2.1 paveiksle.



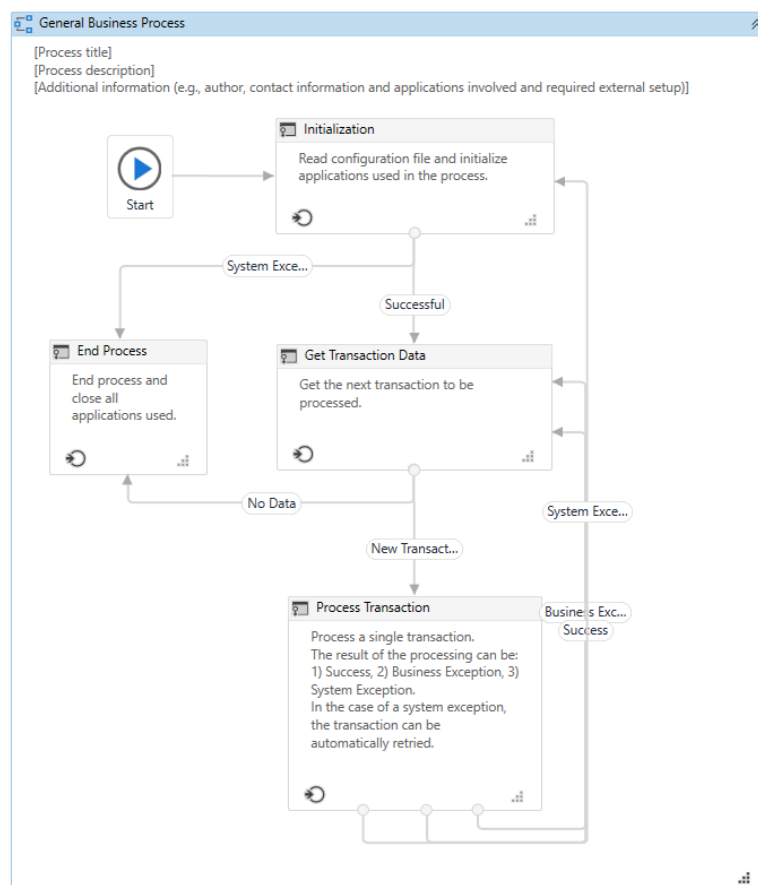
2.1 pav. Supaprastinta RE karkaso diagrama

RE karkasą sudaro keturi pagrindiniai blokai (2.2 pav.):

- Inicijavimas (angl. *Initialization*) – šiame bloke yra nuskaitomas proceso konfigūracijos failas, kur yra pateikiami visi duomenys reikalingi proceso veikimui:

svetainių nuorodos, prisijungimo duomenys, užduočių eilės pavadinimas ir t.t. Taip pat šiame bloke yra sudedamos užduotys į eilę iš kurios paskui jos bus imamos ir atliekamos. Šio bloko pabaigoje yra įjungiamos visos informacinės sistemos, kurios bus naudojamos proceso metu.

- Užduoties duomenų gavimas (angl. *Get Transaction Data*) – šis blokas yra atsakingas už užduoties paėmimą iš eilės ir tos užduoties pradinį duomenų apdorojimą, jeigu tokio reikia. Šis blokas tikrina ar dar yra užduočių eilėje ir jeigu ji yra tuščia, tuomet siunčia signalą proceso užbaigimui.
- Užduoties apdorojimas (angl. *Process Transaction*) – šiame bloke yra atliekamas visas užduoties apdorojimas: duomenų nuskaitymas, skaičiavimai, duomenų surašymas reikalingose sistemose bei išimčių gaudymas / apdorojimas. Jeigu šiame bloke įvyksta nenumatyta klaida (angl. *System Exception*), tuomet yra iš naujo įjungiamos visos sistemos ir pradedama sekanti užduotis. Numatytos klaidos atveju (angl. *Business Exception*) užduočiai yra pritaikoma numatyta išimtis ir po sėkmingo pritaikymo yra pereinama į užduoties duomenų gavimo bloką.
- Proceso užbaigimas (angl. *End Process*) – šis blokas yra vykdomas sėkmės ir klaidos atvejais proceso pabaigoje. Jo pagrindinė užduotis yra uždaryti veikiančias programas ir atlikti veiksmus reikalingus proceso pabaigai.



2.2 pav. RE karkaso diagrama

Apžvelgus RE karkasą, galima matyti, kad jis yra suskirstytas į keturis skirtingus etapus, kurių dėka galima lengviau ir stabiliau automatizuoti pasikartojančius verslo procesus. Svarbu

paminėti, kad jeigu norima procesą automatizuoti su šiuo karkasu, norimas procesas turėtų būti standartizuotas, jo užduotys pasikartojančios ir procese nustatytos jo numanomos klaidos. Kitu atveju procesą automatizuoti bus sudėtinga arba jo automatizavimas iš viso neatsipirks naudojant šį karkasą.

3. ATLIEKAMŲ DARBŲ METODIKA

3.1. Pirkimų viešinimo proceso standartizavimas

IT procesų standartizacija yra būtina strategija šiuolaikinėms organizacijoms. Standartizavimas yra skirtas pagerinti procesų efektyvumą, sumažinti jų klaidas ir padidinti nuoseklumą verslo operacijose. Ji apima vieningo metodo, kaip atlikti įvairias IT susijusias užduotis ir procesus, kūrimą, užtikrinant, kad kiekviena operacija būtų atliekama identiškai, nepriklausomai nuo atliekančio asmens. Taip pat proceso standartizavimas yra privalomas žingsnis norint automatizuoti procesą.

Anksčiau procesas vykdavo taip, kad dalį informacijos pirkimų viešinimo komandai užpildydavo iniciatoriai laisva forma, dažniausiai *Microsoft Word* šablone ir pirkimų viešinimo komanda iš šio užpildyto šablono išsirinkdavo jiems reikiamą informaciją. Kadangi automatizavimo robotai negali patys nuspręsti, kuri informacija yra tinkama, ją reikia pateikti griežtu formatu. Dėl to buvo sudarytas *Microsoft Excel* šablonas, kurį pildys pirkimų viešinimo komanda, su taisyklinga informacija. Pagal šį sudarytą šabloną robotas nuskaitys reikiamą informaciją. Šis šablonas toliau bus vadinamas pirkimų duomenų šablonu.

Iš pirkimų duomenų šablono robotas nuskaitys įvardintas įmones ir pagal jų pavadinimus rekvizitų puslapyje ieškos įmonių kodų. Anksčiau kiekviena darbuotoja įmones rašydavo kokia tik nori tvarka, įmonės būdavo atskirtos kableliais, keliais tarpais, o kartais rašomos iš naujos eilutės. Kad procesas taptų standartizuotas, šių įmonių pavadinimai turės būti išvardinti per kabliataškį, kad robotas juos galėtų teisingai nuskaityti.

Taip pat viešiesiems pirkimams reikia nurodyti pirkimo tipo pasirinkimo pagrindimo straipsnį, kad būtų galima atsekti, pagal kokius teisės aktus buvo pasirinktas pasiūlymą pateikęs tiekėjas. Šiuos straipsnius viešųjų pirkimų komanda turi pasirinkti iš „Lietuvos Respublikos viešųjų pirkimų įstatymo“ [19]. Kadangi straipsnių yra daug ir juos rašant lengva padaryti klaidų, yra atvejų, kai už pirkimus atsakingi asmenys pasirenka ne tą straipsnį, ar pažymi ne tą pirkimo tipą. Šiai problemai išspręsti, kuriamame pirkimų duomenų šablone bus padarytas užrakintas sąrašas su iš anksto įrašytais reikšmėmis. Į šį laukelį nebus galima įrašyti jokios kitos informacijos, apart tos, kuri jau yra paruoštame sąraše.

Šis kuriamas šablonas leis visai pirkimų komandai vienodai supildyti informaciją apie pirkimą ir nepadaryti klaidų pildymo metu.

3.2. Užduočių susidėjimas iš elektroninių laiškų

Norint, kad automatizuotas procesas atlikinėtų užduotis, jas iš pradžių reikia jam pateikti. Šis procesas prasideda nuo užduočių gavimo, kurios yra paaimamos iš elektroninio pašto, nes *Pirkimų Duomenų Sistema (PDS)* šiuo metu turi integruotą funkcionalumą, leidžiantį priskirti pirkimus atitinkamiems atsakingiems asmenims. Ši funkcija yra gyvybiškai svarbi, nes ji užtikrina, kad kiekvienas pirkimas būtų tinkamai identifikuotas ir priskirtas, tačiau ji nepasiūlo lankstumo pasirinkti užduočių paskirstymo būdai.

Dėl sistemos apribojimų, šiuo metu užduotys gali būti gaunamos tik iš *Microsoft Outlook* programos naudojamo elektroninio pašto. Užduočių paėmimo iš elektroninio pašto metodika apima kelis žingsnius.

Pirmiausia reikia patekti į reikiamą elektroninio pašto paskyrą. Tai galima padaryti dviem būdais. Pirmasis būdas – rankinis prisijungimas prie elektroninio pašto per vartotojo grafinę sąsają. Šis metodas reikalauja, kad vartotojas atidarytų pašto programą, naviguotų iki reikiamų laiškų ir tikrindamas kiekvieną laišką nuskaitytų reikiamą informaciją. Tai yra gana daug laiko reikalaujantis procesas, kurio eigoje gali lengvai įsivelti klaidos. Antrasis, žymiai efektyvesnis būdas, yra naudoti specializuotas funkcijas, kurios leidžia programiškai prisijungti prie elektroninio pašto serverio, pasirinkti norimą pašto dėžutę ir automatiškai nuskaityti laiškus pagal pasirinktą būseną, šiuo atveju – neperskaitytus laiškus. Šis metodas sumažina žmogiškąją klaidų atsiradimo galimybę ir pagreitina informacijos apdorojimą, lyginant su laiškų nuskaitymu grafinės vartotojo sąsajos pagalba.

Siekiant optimizuoti laiškų paskirstymą, galima pasinaudoti elektroninio pašto taisyklėmis (angl. *rules*), kurios automatiškai perkelia laiškus į pasirinktą aplanką, jeigu jie atitinka nustatytus parametrus. Šiuo atveju parametrai būtų, kad laiškas turi ateiti iš specifinio siuntėjo ir su konkrečiais raktiniais žodžiais temoje. Tai leidžia efektyviau ir tiksliau valdyti gaunamus duomenis, užtikrinant, kad svarbi informacija būtų greitai ir teisingai perduota automatizuotos sistemos procesams.

3.3. Duomenų nuskaitymas, apdorojimas ir įvedimas

Kiekvienas automatizuotas procesas darbui su duomenimis remiasi pagrindiniais trimis etapais. Pirmasis yra duomenų nuskaitymas, toliau seka duomenų apdorojimas ir galiausiai duomenų suvedimas į reikiamą lauką. Tačiau tai padaryti nėra taip paprasta, kaip atrodo. Kiekvienas iš šių veiksmų turi savo funkcinius ir nefunkcinius reikalavimus, į kuriuos turėtų būti atsižvelgta. Funkciniai ir nefunkciniai reikalavimai apibrėžia sistemų elgseną įvairiose aplinkybėse. Šie reikalavimai gali apimti elementus, tokius kaip skaičiavimai, techninės specifikacijos, duomenų struktūros ir panašiai. Jie gali būti orientuoti tiek į procesus, tiek į informacijos tvarkymą. Reikia paminėti, kad anksčiau minėtuose etapuose šie funkciniai ir nefunkciniai reikalavimai gali būti pritaikomi ne visada, tačiau jeigu yra galimybė, reiktų į juos atsižvelgti.

Duomenų nuskaitymas yra pirmasis ir ypač svarbus etapas, nes jeigu duomenys bus nuskaityti klaidingai tuomet, visi kiti atliekami žingsniai taip pat gali būti atliekami neteisingai. Informacinėse sistemose duomenys gali būti pateikiami įvairiais būdais: teksto laukeliais, lentelėmis ar varnelės pažymėjimu prie tinkamo reikšmės varianto. Dėl to yra svarbu taikyti atitinkamus duomenų surinkimo metodus įvairiems duomenų pateikimo tipams:

1. Gauti tekstą (angl. *Get text*) – tai yra vienas iš patogiausių būdų, kai reikia nuskaityti duomenis iš pavienių teksto laukelių. Ši funkcija gautą reikšmę saugo *string* tipo kintamajame todėl kai kuriais atvejais naudojant šį būdą nereikia daryti papildomo duomenų apdorojimo.
2. Gauti duomenų lentelę (angl. *Extract data table*) – šis metodas yra specialiai pritaikytas informacijos nuskaitymui iš informacinių sistemų ar aplikacijų lentelių. Ši funkcija sistemose ir aplikacijose analizuoja *HTML* kodą [20] ir pagal *<table>* žymėjime esamus duomenis sukuria *DataTable* tipo kintamąjį iš kurio duomenų apdorojimo metu gali būti nesunkiai paimami nuskaityti struktūrizuoti duomenys.
3. Gauti atributą (angl. *Get attribute*) – tai funkcija, kuri leidžia pasirinkti, kokią informaciją norima paimti iš pasirinkto grafinės sąsajos elemento. Kintamojo tipas kinta

priklausomai nuo to, kokį atributą norime gauti. Pavyzdžiui, jeigu pasirenkama gauti ar langelis yra pažymėtas, ar ne, tuomet mūsų kintamojo tipas bus *boolean*. Tai yra lanksti funkcija, skirta įvairių reikšmių, ne tik teksto, gavimui.

Šiuo darbo metu buvo naudojami visi trys duomenų gavimo būdai ir buvo remiamasi 3.1 lentelėje paminėtais funkciniais ir nefunkciniais reikalavimais.

3.1 lentelė Duomenų nuskaitymo funkciniai ir nefunkciniai reikalavimai

Nr.	Reikalavimo aprašymas
DNFR-1	Nuskaityma reikšmė turi būti atpažįstama pagal šalia esantį raktažodį
DNNR-1	Nuskaitymas turi būti vykdomas tik pilnai užsikrovus sistemos puslapiui
DNNR-2	Nuskaitymo pavadinimas turi aiškiai paaiškinti, kokie duomenys yra išgaunami

Duomenų apdorojimas yra antrasis etapas. Kai kuriais atvejais iš gautos informacijos reikia atsirinkti tik specifinius duomenis arba juos paversti į kito tipo kintamąjį, kad būtų galima atlikti reikiamus skaičiavimus. Visa tai yra priskiriama duomenų apdorojimui ir jam dažniausiai yra naudojama funkcija „priskirti“ (angl. *assign*). Tai yra funkcija, kuri gali priskirti parašytą kodą pasirinktam kintamajam, dėl to turi daug panaudojimo galimybių. Dėl skirtingų apdorojimo metodų, nustatyti vienodus reikalavimus gali būti sudėtinga, tačiau 3.2 lentelėje yra pateikti reikalavimai, kurių reiktų stengtis laikytis.

3.2 lentelė Duomenų apdorojimo funkciniai ir nefunkciniai reikalavimai

Nr.	Reikalavimo aprašymas
DAFR-1	Apdorojimo metu turi būti metama klaida, jeigu transformuojami duomenys neatitinka nustatytų reikalavimų
DANR-1	Vienos vertės apdorojimas neturėtų viršyti 1 sekundės

Jeigu duomenų gavimas ir apdorojimas buvo įvykdyti sėkmingai, prieinama prie trečio etapo, kuris yra duomenų pildymas reikiamoje informacinėje sistemoje ar aplikacijoje. Kaip ir duomenų nuskaityme, įvedime taip pat naudojami keli skirtingus įvedimo metodai, kurie leidžia pasiekti norimą rezultatą:

1. Pelės paspaudimas (angl. *Click*) – ši funkcija dažniau naudojama navigavimui per grafinę vartotojo sąsają, tačiau taip pat gali būti naudojama duomenų suvedimui. Kartais reikia nuspausti reikiamą mygtuką priklausomai nuo gautų duomenų, tuomet ir yra naudojamas šis funkcionalumas.
2. Įrašymas į teksto lauką (angl. *Type Into*) – funkcija, kuri yra naudojama, norint užpildyti teksto lauką su norima informacija. Ši funkcija priima *String* tipo kintamąjį ir į teksto lauką įrašo pateiktus duomenis.
3. Pažymėjimas/atžymėjimas (angl. *Check/Uncheck*) – funkcija, kuri yra specialiai sukurta pažymėti ar atžymėti reikiamus langelius. Tai galėtų atlikti ir pelės paspaudimas, tačiau tuomet turėtume tikrinti, ar laukelis yra pažymėtas ar atžymėtas, o pažymėjimo funkcija veikia kiek kitaip. Ji surandą norimą elementą ir jo *HTML* kode pakeičia informaciją prie *isChecked* į *True* arba *False*, priklausomai nuo to, ar mes norime langelį pažymėti ar atžymėti, taip išvengiant papildomo būklės tikrinimo.

Duomenų suvedimui taip pat yra taikomi keli funkciniai reikalavimai (3.3 lentelėje) į kuriuos turėtų būti atsižvelgta.

3.3 lentelė Duomenų pildymo funkciniai reikalavimai

Nr.	Reikalavimo aprašymas
DAFR-1	Į laukelį įvesta reikšmė turi būti patikrinta, ar atitinka norimą įvestį
DAFR-2	Nesėkmingo įvedimo atveju, veiksmas turi būti kartojamas bent 3 kartus

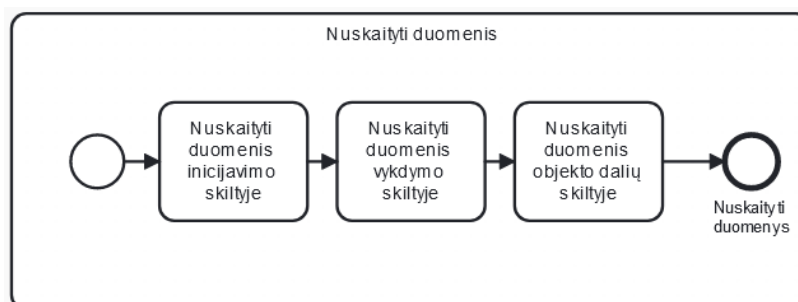
Kiekvienam iš šių etapų yra taikomi tiek funkciniai, tiek nefunkciniai reikalavimai, kurie užtikrina, kad automatizavimo procesai būtų efektyvūs, patikimi ir atitiktų nustatytus kokybės standartus.

3.4. Pirkimų viešinimo procesas

Viešinimo proceso įgyvendinimas yra pagrindinis šios praktikos uždavinys. Pirkimų viešinimo proceso galutinis rezultatas yra pasiekiamas tuomet, kai visa vieša informacija apie pirkimą yra supildyta CVPIS. Visą proceso diagramą galima matyti 2 priede.

Matome, kad pirkimų viešinimo procesas apima 6 skirtingas sistemas, vienos yra įmonės vidinės sistemos: *Outlook*, *Pirkimų Duomenų Sistema (PDS)*, *Doclogix (DLX)* o kitos – išorinės, kurių pokyčių negalime numatyti: *Centrinė Viešųjų Pirkimų Informacinė Sistema (CVPIS)*, *Rekvizitai*, *Centrinis Viešųjų Pirkimų Portalas (CVPP)*. Diagrama prasideda ties *Outlook* sistema, kai procesui paskirtas užduotys įkrenta į bendrą pirkimo užduočių aplanką. Paskui iš eilės yra vykdomas duomenų nuskaitymas iš *PDS*, *DLX*, *CVPP* ir *Rekvizitų* sistemų. Paskutinis etapas yra duomenų pildymas, kuris atvaizduotas CVPIS skiltyje. Jeigu duomenų suvedimo metu įvyko klaida, tuomet robotas eina klaidos keliu ir atmeta užduotį, o sėkmės atveju užduotis yra atliekama ir patvirtinama.

Kadangi šis uždavinys yra gan didelis, šiame punkte bus paaiškinta, kokius veiksmus reikės atlikti viešinimo proceso realizavimui. Kad robotas galėtų paviešinti informaciją CVPIS, už pirkimą atsakingi asmenys turės į nustatytą elektroninį paštą atsiųsti nuorodą į kiekvieno pirkimo kortelę pirkimų duomenų sistemoje. Paskui, atsidaręs atsiųstą nuorodą, robotas turės nuskaityti duomenis iš pirkimų duomenų sistemos (*PDS*), kur viešųjų pirkimų komanda supildys reikiamą informaciją apie pirkimą. Duomenys turės būti nuskaityti iš trijų skirtingų skilčių (3.1 pav.).



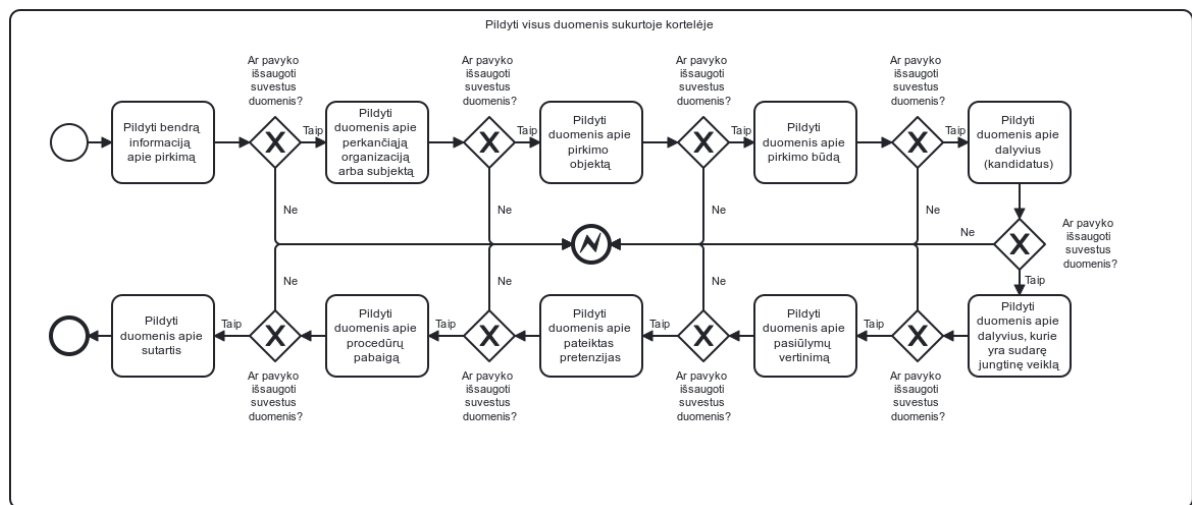
3.1 pav. Pirkimų viešinimo duomenų nuskaitymo iš PDS diagrama

Iš tos pačios sistemos robotas turės atsisiųsti *Microsoft Excel* ataskaitą, kurioje bus pateikta daugiau reikalingų duomenų. Kai visi duomenys bus nuskaityti iš pirkimų duomenų sistemos ir joje esančios ataskaitos, robotas turės patikrinti kokiai įmonei bus vykdomas pirkimas. Jeigu

pirkimas bus vykdomas *ESO* įmonei, tuomet bus atidaroma kortelė dokumentų pildymo sistemoje (*Doclogix*) pagal sutarties numerį, kuris buvo nuskaitytas iš pirkimų duomenų sistemos. Atsidaręs šią kortelę, robotas nuskaitys papildomą informaciją apie pirkimo subrangovus, kuri bus naudojama tolimesniam pirkimo pildymui *CVPIS*. Taip pat robotas keliaus į *Centrinį Viešųjų Pirkimų Portalą (CVPP)* ir ten suvedęs pirkimo sutarties numerį, papildomai surinks informaciją apie pirkimo kainą bei jo pradžios ir pabaigos datas. Toliau robotas atsidarys rekvizitų puslapį ir jame suras visų pirkimo konkurse dalyvavusių įmonių kodus pagal jų pavadinimus.

Jeigu robotas neras būtinų duomenų arba jie bus pateikti neteisingai, jis atmes užduotį ir informuos atsakingus asmenis elektroniniu paštu. Daugiau jokių išimčių procese nėra numatyta. Jeigu robotui sėkmingai pavyko gauti visus duomenis, jis atsidarys *CVPIS* ir supildys informaciją apie pirkimą į atitinkamas pirkimo kortelės dalis. Jeigu, nepavyko išsaugoti tam tikros dalies su užpildytais duomenimis, tuomet yra metama klaida, po kurios yra informuojami atsakingi asmenys ir pati užduotis atmetama *PDS* sistemoje. *CVPIS* kortelės pildymo eigą galima matyti 3.2 pav.

Tolimesnė eiga, kai nepavyko išsaugoti suvestų duomenų, yra pavaizduota 3.2 paveikslėlyje. Tuo atveju turi būti informuojami atsakingi asmenys apie įvykusią klaidą, o pati užduotis atmesta pirkimų duomenų sistemoje. Jeigu visas *CVPIS* supildymas įvyko sklandžiai, tuomet pirkimų duomenų sistemoje užduotis turi būti pažymėta, kaip atlikta.



3.2 pav. Pirkimų viešinimo pirkimų kortelės pildymo diagrama

Tačiau paskutinis žingsnis, pirkimo pavišinimas, vis tiek bus atliekamas viešųjų pirkimų atsakingų asmenų, kai jie patikrins, jog visa supildyta informacija yra teisinga. Kadangi robotas šios vietos neatlieka, ji nėra pavaizduota diagramoje ir realizuojama taip pat nebus.

3.5. Funkciniai reikalavimai diagramoms

Siekiant sukurti interaktyvias diagramas, būtina nustatyti, kokius kriterijus ir informaciją turėtų jos atvaizduoti. Funkciniai reikalavimai sudaromi siekiant užtikrinti, kad sukurtos diagramos atitiktų pradinis lūkesčius. Remiantis šiais duomenimis, buvo suformuluoti reikalavimai kelioms skirtingoms diagramoms.

3.4 lentelė Užduočių kiekio diagramų funkciniai reikalavimai

Nr.	Reikalavimo aprašymas
UKDFR-1	Diagrama turi leisti išfiltruoti užduotis pagal užduoties atlikimo būseną
UKDFR-2	Diagrama turi rodyti pasirinkto periodo duomenis
UKDFR-3	Diagrama turi vizualiai išskirti užduotis pagal užduoties atlikimo būseną

Atsižvelgiant į 3.4 lentelėje išdėstytus reikalavimus, reikės sukurti dvi išsamias diagramas. Pirmoji diagrama bus skirta užduočių sekimui pagal dienas, kuri leis vizualiai įvertinti, kurios dienos yra labiausiai užimtos ir kuriomis dienomis organizacija patiria didžiausią darbo krūvį. Antrąją diagramą reikės parengti užduočių kiekio stebėjimui pa mėnesiui, kas suteiks galimybę atsakingiems asmenims nustatyti, kurie metų laikotarpiai yra ypač intensyvūs.

Abi diagramos turės automatiškai apdoroti ir vaizduoti užduočių būsenas: kiek jų buvo sėkmingai įvykdyta, kiek – neįvykdyta dėl numatytų verslo ir nenumatytų sisteminių klaidų, taip pat atspindėti staigius gedimus ar netinkamą robotų veikimą, kai užduotys atliekamos neteisingai arba visai nebevykdomos. Ši informacija bus labai svarbi, kad atsakingi asmenys galėtų efektyviai stebėti automatizuoto proceso veikimą, išanalizuoti problemų priežastis ir, remiantis gauta informacija, planuoti būsimas veiklos optimizacijas.

3.5 lentelė Klaidingų užduočių diagramos funkciniai reikalavimai

Nr.	Reikalavimo aprašymas
KUDFR-1	Diagrama turi leisti išfiltruoti užduotis pagal klaidos pranešimą
KUDFR-2	Diagrama turi rodyti pasirinkto periodo duomenis
KUDFR-3	Diagrama turi rodyti, kiek kartų kiekviena skirtinga klaida pasikartojo

Remiantis 3.5 lentelėje pateiktais reikalavimais, bus reikalinga sukurti išsamią diagramą, kuri vaizdžiai atspindėtų visas nesėkmingai atliktas užduotis, jų būklę bei dažnumą, kiek kartų šios klaidos pasikartojo. Norint užtikrinti, kad diagrama atitiktų šiuos reikalavimus, būtina jai suteikti informaciją apie neįvykdytas užduotis, jų būsenas bei susijusius klaidų pranešimus. Diagrama automatiškai turės apibendrinti ir vizualiai pateikti vienodo pobūdžio pasikartojančias klaidas, kad atsakingi asmenys ir programuotojai galėtų aiškiai identifikuoti, kokios klaidos

3.6 lentelė Sutaupyto laiko diagramos funkciniai reikalavimai

Nr.	Reikalavimo aprašymas
SLDFR-1	Diagrama turi leisti išfiltruoti sutaupytą laiką pagal mėnesį
SLDFR-2	Diagrama turi rodyti pasirinkto periodo duomenis
SLDFR-3	Diagrama turi rodyti, kiek valandų per kiekvieną mėnesį buvo sutaupyta

pasikartoja dažniausiai. Ši informacija leis ateityje nukreipti dėmesį į dažniausiai pasitaikančių problemų sprendimą ir tobulinimą.

Remiantis 3.6 lentelėje pateiktais duomenimis, būtina sukurti vizualiai informatyvią diagramą, kuri atvaizduotų, kiek valandų per mėnesį sutaupoma dėl automatizuoto proceso įgyvendinimo. Ši diagrama turės rodyti, kiek užduočių per mėnesį buvo atliktos sėkmingai, ir kiekvienai iš šių užduočių turės būti priskirtas laiko taupymo koeficientas, t. y., kiek laiko sutaupo viena sėkmingai atlikta užduotis.

Diagrama automatiškai suskaičiuos ir grafikai pavaizduos sutaupytas valandas kiekvienam mėnesiui, leidžiant analizuoti, kaip taupymo efektyvumas keičiasi metų bėgyje. Tai suteiks galimybę atsakingiems asmenims ne tik matyti bendrą laiko sutaupymą, bet ir identifikuoti laikotarpius, per kuriuos užduočių skaičius yra aukščiausias. Tokia analizė padės suprasti, kada automatizuoti procesai duoda didžiausią naudą ir kaip galima dar geriau išnaudoti automatizacijos teikiamas galimybes efektyvumui didinti.

3.7 lentelė Užduočių vykdymo diagramos funkciniai reikalavimai

Nr.	Reikalavimo aprašymas
UVDJR-1	Diagrama turi rodyti užduočių statusų pasiskirstymą procentais
UVDJR-2	Diagrama turi rodyti pasirinkto periodo duomenis

Atsižvelgiant į 3.7 lentelėje pateiktus reikalavimus, reikės sukurti diagramą, kuri atvaizduos kokia procentinė dalis užduočių buvo atlikta sėkmingai, kuri su numatytomis verslo klaidomis ir kuri su nenumatytomis sisteminėmis klaidomis. Norint, kad diagrama atvaizduotų šiuos duomenis, jai būtina pateikti visas darytas užduotis, nepriklausomai nuo jų statuso. Tuomet diagrama automatiškai paskaičiuoja jai reikiamas reikšmes. Ši diagrama leis matyti bendrą, automatizuoto proceso pasisekimo procentą ir taip pat, koks kiekis užduočių buvo verslo ar sisteminės klaidos.

4. PROCESO IR VIZUALIZACIJOS REALIZAVIMAS

4.1. Proceso tinkamumo automatizavimui vertinimas

Prieš pradedant automatizuoto proceso realizavimą, yra svarbu įsivertinti ar jį tikrai apsimoka automatizuoti naudojantis RE karkasu. Kaip buvo minėta anksčiau, procesas turi būti standartizuotas, pasikartojantis bei turi būti nustatytos numatomos klaidos. Pats proceso standartizavimas vyko su klientų (viešųjų pirkimų komandos) pagalba, kai buvo sudaroma automatizuojamo proceso dokumentacija. Tuo metu taip pat buvo nustatytos pirkimų viešinimo proceso numanomos klaidos. Taip pat pagal klientų pateiktus duomenis, galima matyti, kad procesas turi daug pasikartojančių užduočių. Pagal šią turimą informaciją galima teigti, kad procesą būtų galima automatizuoti naudojant RE karkasą.

Tačiau taip pat reikia įvertinti ar procesas atneš naudą ir atsipirks. Proceso privalumai ir trūkumai buvo aptarti analizės metu ir iš to galima spręsti, kad proceso privalumai persveria trūkumus. Bet privalumų ir trūkumų neužtenka nustatyti galutiniam proceso automatizavimo tinkamumui. Kad būtų galutinai patikrintas tinkamumas automatizavimui, buvo panaudotas *UiPath* verslo automatizavimo analitiko mokymų pabaigoje siūlomas įrankis, kuris buvo išverstas į lietuvių kalbą [21]. Į įrankį suvedus automatizuojamo proceso duomenis tokius kaip: įvedimo tipai, proceso stabilumas, programų stabilumas, priežastys, proceso sudėtingumas ir įvedami duomenys, buvo gautas rezultatas, kad procesą įgyvendinti yra paprasta ir jis atneš didelę naudą, tačiau proceso įgyvendinimui reikia profesionalių žinių (4.1 pav.). Proceso įgyvendinimui reikės nedaug pastangų, nes jis prieš tai buvo standartizuotas ir dėl to veiksmai tapo lengvai suprantami.

REZULTATAI	
ĮGYVENDINIMO PAPRASTUMAS	Mažai pastangų
NAUDA/TINKAMUMAS	Didelė nauda
KŪRIMO PAPRASTUMAS	Šiam procesui automatizuoti reikia profesionalių žinių. Pateikite šią idėją savo automatizavimo komandai
LAISVOS VALANDOS (žmogaus valandos per metus)	
	381
KLAIDŲ SUMAŽINIMAS	88%

4.1 pav. Procesų vertinimo įrankio rezultatai

Proceso duomenų suvedimo metu buvo pasirinkta, kad proceso duomenys yra skaitmeniniai ir struktūrizuoti ir bus atliekami tam tikri pokyčiai metų eigoje. Taip pat buvo įvesta, kad darbuotojai procesui skiria po 100 minučių per darbo dieną ir pats procesas vykdomas per daugiau nei 5 sistemas.

Kadangi procesas yra standartizuotas, jo privalumai persveria trūkumus ir teikiama nauda bus gan didelė, galima spręsti, kad procesas yra tinkamas automatizavimui. Pagal 4.1 pav. galima matyti, kad klaidų turėtų sumažėti 88 procentais ir bus atlaisvinta apie 380 darbo valandų per metus. Nustačius, kad procesas yra tinkamas automatizavimui, galima pradėti realizuoti proceso automatizavimą, o tai yra aprašyta sekančiuose skyriuose.

4.2. Užduočių gavimas ir susidėjimas

Proceso užduočių gavimas yra pradinis veiksmas, kuris prasideda tuo metu, kai užduotys atkeliauja į pagrindinę Microsoft Outlook pašto dėžutę. Atsakingi asmenys patys pasirenka, kad užduotis būtų paskirta vartotojui, kurio pavadinimas yra „robotas_1“. Tuomet įsigalioja specialiai sukurtos Outlook taisyklės (angl. *rules*), kurios automatiškai atpažįsta gautas užduotis ir perkelia jas į tam tikrą aplanką, šiuo atveju – į pirkimų viešinimo aplanką pavadinimu „SP Užduotys (10, 79, 91)“. Sukurta taisyklė yra pavaizduota (4.2 pav.). Šiuo atveju nustatyta, kad ateinantis laiškas temoje turi turėti frazę „Informavimas apie sukurtą užduotį“. Šis funkcionalumas yra labai naudingas siekiant išlaikyti el. pašto dėžutės tvarką ir užtikrinti, kad visos užduotys būtų sistemingai skirstomos pagal jų paskirtį.

The screenshot shows the 'Rules' configuration window in Outlook. At the top, the title bar says 'Rules' with a close button (X). Below the title bar, there is a list of rules. The first rule is selected, indicated by a blue checkmark in a circle. The rule name is 'SP užduotys pirkimams'. Below the rule name, there are two sections: 'Add a condition' and 'Add an action'. Under 'Add a condition', there are two conditions: 'From: Intranet' and 'Subject includes: Informavimas apie sukurtą užduotį'. Each condition has a dropdown arrow and a close button (X). Below the conditions, there is a link 'Add another condition'. Under 'Add an action', there is one action: 'Copy to: SP užduotys (10, 79, 91)'. The action has a dropdown arrow and a close button (X). Below the action, there are links 'Add another action' and 'Add an exception'. At the bottom, there is a checkbox 'Stop processing more rules' with an information icon (i).

4.2 pav. Laiškų perkėlimo taisyklė

Verta paminėti, kad anksčiau aptartas pirkimų viešinimo aplankas nėra naudojamas tik vieno proceso reikmėms. Juo naudojasi trys papildomi procesai, kurie taip pat susiję su pirkimų viešinimu, bet kiekvienas iš jų rengia skirtingo tipo ataskaitas. Tai reiškia, kad iš šio aplanko reikia perkelti tam tikras užduotis į specifiškai sukurtą proceso aplanką. Kad tai atliktų robotas, iš pradžių reikia sužinoti pirkimo tipą, tačiau to nesimato iš pačio laiško. Dėl to reikia atlikti papildomą veiksmą – atsidaryti laiške esančią nuorodą ir nuskaityti pirkimo tipą (4.3 pav.). Kad atsidarytų pirkimas per laiško nuorodą, reikia suvesti prisijungimo duomenis, tuomet pagrindiniame lange galima matyti pirkimo tipą. Jei nurodytas pirkimo tipas yra „Ataskaita“, tuomet robotas perkelia šį laišką į jam specialiai sukurtą pašto dėžutę (4.3 pav.).



4.3 pav. Pirkimo tipo vaizdas

Šį veiksmą būtų galima daryti per vartotojo grafinę sąsają, tačiau buvo nuspręsta, kad būtų patikimiau tai padaryti per aplikacijų programavimo sąsają (angl. *Application Programming Interface*). Jeigu tiksliau, tai pavyko atrasti puslapyje naudojamą nurodą, kuri gražina pirkimo tipą pagal nurodyto pirkimo numerį ir įmonę. Šio funkcionalumo nepavyko įgyvendinti su *UiPath* programos funkcijomis, nes yra naudojamas specializuotas prisijungimas, dėl kurio neveikia paprastas vartotojo duomenų padavimas programai. Dėl šios priežasties, šiam funkcionalumui įgyvendinti buvo naudojama *Python* programavimo kalba su specializuotomis bibliotekomis, kurios padėjo prisijungti prie *PDS* ir gauti pirkimo tipą naudojant specialią nuorodą.

Šiai programai buvo sukurta virtuali aplinka, kad procesas galėtų veikti ant skirtingų serverių ir vartotojų. Kadangi buvo sukurta virtuali aplinka, nepavyko pasinaudoti *UiPath* siūlomomis galimybėmis per ją paleisti šį kodą, dėl šios priežasties buvo daromas šios veikimo apėjimas. Iš pradžių iškviečiamas komandinė eilutės (angl. *command prompt*) laukas, per kurį yra paleidžiamas kodo *.bat* failas. Šis failas yra skirtas, kodo paleidimui naudojantis virtualia aplinka. Šio failo kodas yra pateiktas 4.4 pav.

```
@echo off
call
"...\RPA_config\10._Pirkimu_ataskaitos_pildymas_CVPIS\GetPurchaseType\env\Scripts
activate.bat"
python "...\RPA_config\10._Pirkimu_ataskaitos_pildymas_CVPIS\GetPurchaseType\
getPurchaseType.py" %1
Call
"...\RPA_config\10._Pirkimu_ataskaitos_pildymas_CVPIS\GetPurchaseType\env\Scripts
deactivate.bat"
```

4.4 pav. Pirkimų tipo gavimo programos paleidimo failo kodas

Ši programa priima vieną argumentą, tai yra kelias iki tekstinio dokumento su visais reikiama duomenimis. Šie duomenys yra paduodami per tekstinį dokumentą, nes kitu būdu gali būti, kad dėl lietuviškų raidžių bus iškraipomi pateikiami ir gaunami kodo duomenys. Su visais gautais duomenimis: prisijungimo vardu, slaptažodžiu, pirkimo tipo gavimo nuoroda ir išeities keliu, yra gaunamas pirkimo tipas pagal kurį robotas gali atpažinti pirkimo tipą ir jį paskirti į tinkamą automatizuoto proceso aplanką. Kodas išeities (angl. *output*) duomenis pateikia taip pat tekstiniam dokumente, kad neiškraipytų žodžių su lietuviškomis raidėmis. Pirkimo tipo gavimo kodą galima matyti 3 priede.

Robotas nusiskaito gautą tekstinį failą su pirkimo tipu ir pagal jį priskiria užduotį į tam tikro proceso aplanką. Sukėlus visus tinkamus laiškus į proceso aplanką, yra nuskaityti neperskaityti laišškai ir jų informacija sudedama, kaip užduočių duomenys. Vienos užduoties duomenis eilėje detalčiai galima matyti (4.5 pav.).

```
Specific Data: Object      Output Data: null
url:
  http://pde.corp.rst.lt/pirkimai/
  VKJ2023/202310328/Purchase.a
  spx
purchase: (2023-VKJ-328)
```

4.5 pav. Užduoties duomenys eilėje

Kai laiškas yra nuskaitymas ir jo duomenys perkelti robotui į eilę, tuomet jis yra pažymimas, kaip perskaitytas. Visa tai yra kartojama tol, kol yra nėra perskaityti visi laišškai „SP Užduotys (10, 79, 91)“ aplanke.

4.3. Sistemų atidarymas ir prisijungimas

Sekantis etapas, kuris eina po eilės susidėjimo yra naudojamų sistemų atidarymas. Šiam procesui įvykdyti taip pat reikia atidaryti visas reikalingas informacines sistemas ir prie daugelio iš jų prisijungti. Šių programų atidarymas vyksta iš karto prieš susidėtų užduočių apdorojimą tam, kad šis veiksmas būtų atliekamas kuo mažiau kartų ir sutaupyta licencijų laiko. Tai yra vienas iš žingsnių, kuris bus kartojamas, jeigu apdorojimo metu įvyks nenumatytų sisteminių klaidų. Tokiu atveju visos sistemos bus išjungtos ir įjungtos iš naujo, kad būtų galima iš naujo švariai pradėti dirbti su reikalingomis užduotimis.

Jeigu prie sistemų reikia prisijungti, tokiu atveju turi būti gaunami sistemų prisijungimo duomenys. *UiPath Orchestrator* programa siūlo specialų funkcionalumą, kuris debesų saugykloje saugo visus įrašytus prisijungimo duomenis. Šie duomenys gali būti pasiekiami su specialia funkcija pavadinimui gauti prisijungimo duomenis (angl. *Get App Credentials*). Šiai funkcijai nurodžius, norimos gauti sistemos pavadinimą, yra gražinami vartotojo vardas ar elektroninis paštas ir slaptažodis. Vardas ar elektroninis paštas yra gražinami *String* formatu, tačiau slaptažodis dėl saugumo taisyklių yra gražinamas *SecureString* formatu. Šis slaptažodis yra užkoduotas, kad nebūtų galima jo paprastai išrašyti į konsolę ir taip pat naudojant rašymo funkciją, reikia pasirinkti, kad tekstą rašyti, kaip *SecureString* kintamąjį. Sėkmingai prisijungus prie visų sistemų yra pradedamas eilėje turimų užduočių atlikimas, pradedant nuo duomenų nuskaitymo.

4.4. Duomenų nuskaitymas iš sistemų

4.4.1. Duomenų nuskaitymas iš PDS sistemos

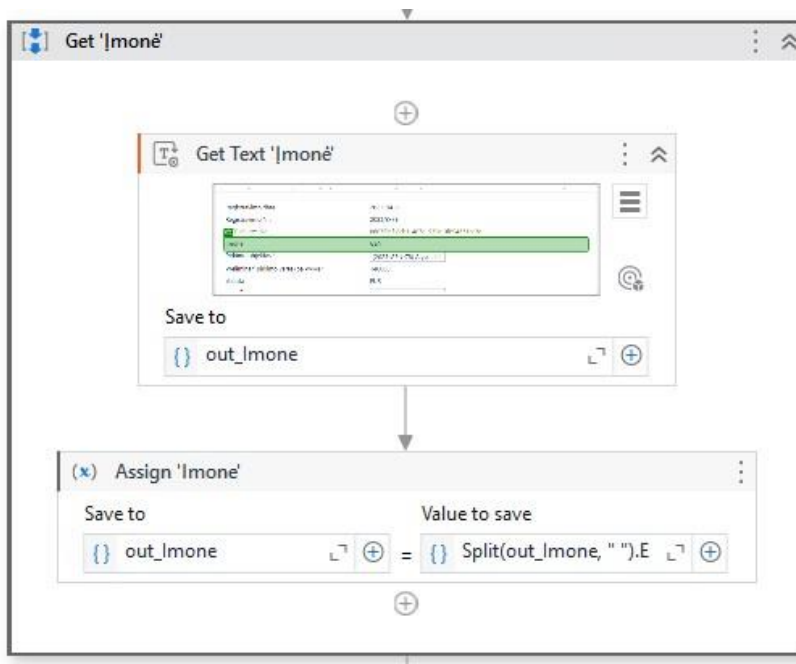
Visas duomenų nuskaitymas prasideda PDS erdvėje, kur yra supildyta didžioji dalis informacijos apie patį pirkimą. Iš čia, taip pat, sužinojome pirkimo tipą, kai rūšiavome tinkamas procesui užduotis. Ši sistema turi keturias skiltis iš kurių mums reikia paimiti duomenis: inicijavimas, vykdymas, objekto dalys ir dokumentai. Pirmiausia duomenų rinkimą pradedam nuo inicijavimo skilties, kur nusiskaitome keletą laukų (4.6 pav.).

inicijavimas	Vykdymas	Objekto dalys	Dokumentai
Registravimo data	2023-04-21		
Registravimo Nr.	2023/3-78		
MPP eilutės Nr.	0687f0af-8cb1-407d-939a-0f894651cc3d		
1 Imonė	GEN		
2 Pirkimo objektas *	(2023-GEN-78) Alyvos filt		
Preliminari pirkimo vertė (be PVM) *	140000		
Valiuta	EUR		
Medžiagų vertė			
3 Pirkimo objektas skaidomas į dalis			

Pildyti tinklo infrastruktūros pirkimams.	
Hidraulinių pavarų alyvos filtravimo elementai	6000
Turbinų tepimo ir hidraulinės valdymo sistemos alyvos filtravimo elementai	87000
Hidraulinės vožtuvų valdymo sistemos alyvos filtravimo elementai	40000
Mobilias alyvos valymo įrenginio alyvos filtravimo elementai	7000

4.6 pav. Nuskaitomi laukai inicijavimo skiltyje

Nusiskaitymo kodas atrodo taip (4.7 pav.). Tekstas yra randamas pagal nurodytą parinkiklį (angl. *selector*) ir raktinį žodį. Tokiu pačiu principu nusiskaitome duomenis iš vykdymo skilties ir nepamirštame į konsolę išvesti gaunamų duomenų, kad pasitikrintume ar tikrai gauname reikiamus duomenis. Duomenų išvedimas į konsolę yra reikalingas, kad įvykus klaidoms, būtų galima matyti visus gautus duomenis ir nuspręsti, kokioje vietoje įvyko klaida neatsidarius proceso kodo. Tai taip pat padeda atskirti kokioje proceso dalyje, kokie duomenys buvo gauti, nes duomenų gali būti labai daug ir jie kartais gali būti panašūs vieni į kitus.



4.7 pav. Nuskaitomo lauko kodo pavyzdys

Kai reikia traukti duomenis iš objekto dalių skilties, situacija yra šiek tiek keblesnė, nes čia gali būti keli objektai, kurio tikslaus skaičiaus nežinome. Taip pat, vien tik esant vienai daliai, reiktų ištraukti vienuolika skirtingų reikšmių, kurios yra visos būtinos sėkmingam užduoties atlikimui. Dėl to gaunamas reikšmės saugome nebe *string* kintamajame bet sąrašas *List<string>*. Taip pat apačioje turime lentelę, kurios duomenis taip pat turime nusiskaityti. Kadangi ji gali turėti kelias eilutes, buvo pasirinktas būdas informaciją saugoti *DataTable* kintamajame. Saugojimo pavyzdį galima matyti 4.1 lentelėje.

4.1 lentelė. Duomenų saugojimo pavyzdys

Objekto Nr.	Eil. Nr.	Tiekėjas	Pasiūlymo kaina	Ekonominio naudingumo balai
0	1	UAB „Įmonė 1“	2933,84	95
0	2	AB „Įmonė 2“	3728	80
0	3	„Įmonė 3“, MB	5110	75
1	1	UAB „Įmonė 4“	6231	63

Paskutinėje skiltyje „Dokumentai“ robotas turi nukeliauti iki ataskaitos dokumento: *Dokumentai -> VAC dokumentai -> Ataskaita.xlsx*. Nukeliavęs iki šio dokumento, jis jį atsisiunčia ir nusiskaito jame esančius duomenis.

Duomenų nuskaitymui iš *.xlsx* tipo dokumentų yra naudojamos specialios UiPath programos funkcijos, kurios gali būti pridėtos prie kiekvieno projekto, kaip biblioteka. Iš pradžių šį dokumentą reikia atidaryti naudojant vienos apimties (angl. *Single Scope*) funkciją, kad esant sulėtėjusiai veiklai ar ištikus trikdžiams, robotas nebandytų atidaryti naujo *Microsoft Excel* dokumento, neuždarius tuo metu naudojamo. Kad galėtume nuskaityti duomenis, reikia nurodyti ataskaitos puslapį, kuris šiuo metu vadinasi „Sheet1“ ir taip pat duomenų regioną, kurį norime nuskaityti. Šio dokumento atveju, turėsime nuskaityti visą įrašytą tekstą nuo pačio pirmo langelio. Tai yra atliekamas pagal nutylėjimą, jeigu nenurodomos ribos tarp kurių reikia ištraukti informaciją. Šie duomenys yra pateikiami *DataTable* kintamajame. Nuskaitytus duomenis iš šio dokumento, su šia sistema darbas bus atliekamas tik pačioje užduoties pabaigoje.

4.4.2. Duomenų nuskaitymas iš DLX sistemos

Nuskaitus duomenis iš *PDS* sistemos, reikia įvertinti, kokia įmonė vykdo viešąjį pirkimą. Jeigu perkančioji įmonė yra *ESO*, tuomet reikia eiti į dokumentų valdymo sistemą *DLX* ir ten nuskaityti informaciją apie subrangovus. Subrangovų gavimas vyksta taip:

1. Atidaroma *DLX* sistema;
2. Prie jos prisijungiama su specialiai šiam procesui sukurtu vartotoju;
3. Atsidarius reikiamą aplanką (naudojama rodinio nuoroda) yra surandama sutartis pagal sutarties numerį, kuris taip pat buvo nuskaitytas iš *PDS*;
4. Atidaroma sutartis ir surandami jos subrangovai;
5. Atsidarius kiekvieną subrangovą atskirai yra nuskaitymas jų įmonės kodas ir pilnas pavadinimas.

Ši surinkta informacija apie subrangovus taip pat yra laikoma *DataTable* kintamajame, nes kiekvienai pirkimo daliai yra skirtinga sutartis ir kiekviena skirtinga sutartis gali turėti keletą skirtingų subrangovų su jų įmonės kodais.

4.4.3. Duomenų nuskaitymas iš CVPP

Galiausiai reikia nueiti į *Centrinį Viešųjų Pirkimų Portalą* ir pasižiūrėti, pirkimo datą bei užregistruotą pirkimo kainą. Kad tai padarytų robotas, jis iš pradžių turi nueiti į CVPP sistemą, kuriai nereikia jokių prisijungimo duomenų. Tuomet suvesti gautą sutarties numerį į paiešką ir iš gautos lentelės nuskaityti reikalingus duomenis. Naudojamos funkcijos yra panašios į praeitus duomenų nuskaitymo punktus, dėl to kiekvienas žingsnis nėra detalizuojamas. Galutinė informacija apie pirkimą taip pat yra laikoma sąrašė *List<String>* kadangi kiekvienas pirkimo objektas turi skirtingą sutarties numerį.

4.5. Duomenų sutikrinimas

Sekantis etapas po duomenų rinkimo yra atidus duomenų tikrinimas, kurio tikslas – įsitikinti, jog visi reikalingi duomenys ne tik buvo gauti iš atitinkamų sistemų, bet ir kad jie yra pateikti teisingu ir tinkamu formatu. Ši veikla yra gyvybiškai svarbi, nes net ir menkiausias informacijos trūkumas ar netikslumas gali turėti didelių pasekmių tolesnių veiksmų efektyvumui ir teisingumui.

Procesas prasideda nuo tuščio tekstinio (*String* tipo) kintamojo, kuris paprastai pavadintas „klaidos“, sukūrimo. Šis kintamasis vėliau bus naudojamas kaip talpykla klaidų, susijusių su trūkstama ar neteisingai pateikta informacija, fiksavimui. Jeigu tikrinimo metu aptinkama, kad trūksta būtinos reikšmės ar ji pateikta netinkamu formatu, į „klaidos“ kintamąjį įrašoma klaidos žinutė, nurodant, kokios tiksliai informacijos trūksta.

Proceso pabaigoje atliekamas svarbus žingsnis – galutinis „klaidos“ kintamojo turinio patikrinimas. Jeigu šiame kintamajame yra užfiksuota bent viena tuščia reikšmė (t.y., jis nėra tuščias), tai reiškia, kad rasta bent viena problema, dėl kurios negalima tęsti tolesnių veiksmų be klaidų ištaisymo. Tuo atveju, sistema automatiškai generuoja klaidos pranešimą ir siunčia jį atsakingiems asmenims arba komandoms, informuodama apie trūkstamus arba neteisingai pateiktus duomenis. Šis pranešimas apima ir konkretų klaidų aprašymą iš „klaidos“ kintamojo, suteikiant gavėjams aiškų supratimą apie problemos esmę ir reikalingus taisymo veiksmus.

4.6. Duomenų pildymas CVPIS ir užduoties uždarymas

Jeigu visi reikalingi duomenys buvo sėkmingai nuskaityti, pereinama prie pirkimo pildymo CVPIS. Prie sistemos buvo prisijungta inicijavimo dalyje ir dabartinis žingsnis yra reikiamo pirkimo suradimas ir jo kortelės sukūrimas CVPIS. Šis etapas yra vykdomas taip:

1. Pagrindiniame lange paspaudžiamas mygtukas „Vykstantys“;
2. Atsidariusiame lange yra suvedamas pirkimo pavadinimas ir paspaudžiamas mygtukas filtruoti;
3. Radus pirkimą, jį reikia atsidaryti;
4. Atsidariusiame lange spaudžiamas mygtukas „Procedūrų ataskaitos“ ir sukuriamą pirkimo kortelė, kuri turės būti pildoma su visais nuskaitytais duomenimis;

Sukurtoje kortelėje reikia užpildyti 11 skirtingų dalių, kad roboto darbas užsiskaitytų kaip sėkmingas. Kadangi tai labai ilgas procesas, aprašytos bus tik kelios dalys, kad būtų suprantamas pildymo principas.

4.6.1. Informacijos pildymas apie pirkimą

Pirmiausia yra supildoma informacija apie pirkimą. Kiekvienas įvedimas į tekstinį lauką yra vykdomas keliais žingsniais:

1. Ištrinama viskas, kas yra įrašyta laukelyje;
2. Įrašomas norimas tekstas;
3. Patikrinama ar įrašytas tekstas yra toks, kokio tikimasi;
4. Jeigu tekstas ne toks, kokio tikimasi, tuomet yra kartojami 1 – 3 žingsniai;

Tokiu būdu yra suvedami duomenys pirmame laukelyje (4.8 pav.). Kiti trys punktai turi būti supildomi paspaudžiant „TAIP“ arba „NE“ priklausomai nuo gautų duomenų. Taip pat trečias punktas yra visuomet žymimas „TAIP“ ir jį nuspaudus atsiranda dar trys laukai, kuriuos reikia supildyti. Jie yra supildomi tokiu pačiu principu, kaip ir praeiti laukeliai.

Duomenų pildymo forma

I dalis: Bendra informacija apie pirkimą

1. Pirkimo Numeris 14

1.1. Pirkimo Numeris *

2. Pirkimo pavadinimas

Pirkimo objekto pavadinimas *

(2023-G5C-473) AV - Belaidžio kompiuterių tinklo prieigos

3. Pirkimo vertė * 15

☐ Tarptautinis pirkimas

☐ Supaprastintas pirkimas

4. Ar pirkimas yra susijęs su projektu ir/arba programa, finansuojama Europos Sąjungos ar kitų šalių fondų lėšomis? * 16

☐ TAIP ☐ NE

5. Ar pirkimas atliekamas centrinės viešųjų pirkimų informacinės sistemos priemonėmis (elektroninis pirkimas). *

☐ TAIP ☐ NE

17

Išsaugoti Išsaugoti ir toliau 18

4.8 pav. Pildomi laukai apie pirkimo bendrą informaciją

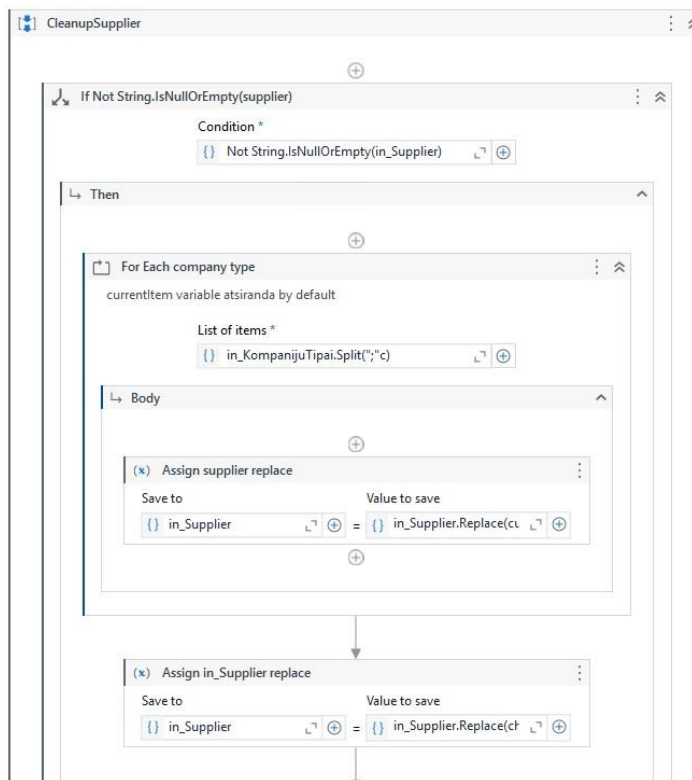
Šių laukų pildymas turi būti griežtas ir atliktas teisingai, nes kiekvienas ne toks paspaudimas gali pakeisti kitų pirkimo pildymo dalių duomenų pildymo laukelius ir užduotis nebus įvykdoma sėkmingai.

4.6.2. Dalyvių pildymas

Šioje dalyje reikia supildyti visas pirkime dalyvavusias įmones ir fizinius asmenis. Kaip buvo minėta anksčiau, sukūrus kortelę, jau gali būti automatiškai supildyta dalis dalyvių. Todėl iš pradžių reikia nusiskaityti, kokie dalyviai jau yra supildyti. Kitas žingsnis yra pasižiūrėti ar visi dalyviai yra įmonės. Tai atliekama pagal specifinius įmonių raktinius žodžius: UAB, AB, MB, VĮ ir t.t. Po atpažinimo einama į rekvizitų puslapį ir pagal pavadinimus yra surandami įmonių kodai. Sekantis žingsnis, iš visų skirtingų laukų gautus dalyvius sudėti į vieną sąrašą ir panaikinti dublius.

Kad tai būtų atlikta sėkmingai, reikia apvalyti įmonių pavadinimus. Tai yra atliekama keliais žingsniais (4.9 pav.):

1. Iš pradžių nuimami visi žodžiai, kurie diktuoja, kad tai įmonė, pavyzdžiui: UAB, AB, MB, VĮ ir t.t.
2. Toliau nuimamos visos kabutės, kableliai, taškai ir tarpai;
3. Galiausiai yra paimami tik žodžiai, be jokių kitų atsitiktinių ženklų.



4.9 pav. Tiekėjo apvalymo kodo pavyzdys

Panaikinus įmonių dublius ir suradus jų įmonių kodus, galima pradėti pildyti informaciją *CVPIS*. Pirmiausia reikia pažymėti ar rašomas dalyvis yra įmonė ar fizinis asmuo. Jeigu tai yra įmonė, papildomai, šalia pavadinimo, reikia įrašyti įmonės kodą. Jeigu yra daugiau nei vienas dalyvis, reikia spausti mygtuką „Pridėti naują objektą“ ir kartoti ką tik paminėtus veiksmus.

4.6.3. Užduoties užbaigimas

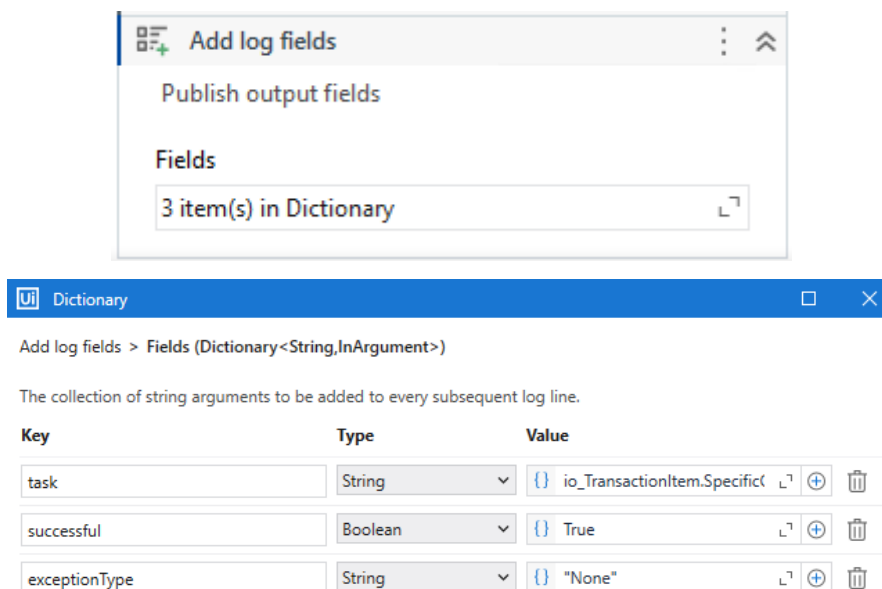
Kai viskas yra sėkmingai supildyta *CVPIS*, tuomet *PDS* sistemoje reikia pažymėti, kad užduotis yra atlikta. Tai padaroma šoninėje juostoje paspaudžiant „Patvirtinti“. O jeigu užduoties nepavyko atlikti, tuomet toje pačioje vietoje reikia įrašyti komentarą, kad nepavyko atlikti užduoties ir paspausti mygtuką „Atmesti“ (4.2 pav.). Galiausiai, jeigu nepavyko sėkmingai apdoroti užduoties, papildomai turi būti išsiunčiamas laiškas atsakingiems asmenims su visa trūkstama informacija. Atsakingi asmenys turės papildyti ar pakoreguoti nurodytas klaidas ir užduotį paskirti robotui iš naujo.

4.7. Interaktyvių diagramų kūrimas

Kai jau procesas veikė sėkmingai, paskutinis žingsnis buvo interaktyvių diagramų sukūrimas, kad klientai galėtų matyti, kaip robotas atlikinėja užduotis. Diagramos buvo sukurtos

pagal iš anksto nustatytus funkcinis reikalavimus (3.5 Funkciniai reikalavimai diagramoms). Tačiau iš pradžių reikėjo užduočių duomenis perkelti į diagramų aplinką *Kibana*. Šioje aplinkoje buvo kuriamos interaktyvios diagramos, kuriomis galima pasidalinti su kitais asmenimis.

Duomenų perdavimas į *Kibana* vyko keliais žingsniais. Iš pradžių užduočių informacija buvo patalpinta *UiPath* suteikiamoje duomenų bazėje, nes yra padarytos specialios funkcijos, kurios leidžia be jokių papildomų argumentų patalpinti norimą informaciją apie atliekamas užduotis (4.10 pav).



4.10 pav. Duomenų įrašymas į *UiPath* duomenų bazę

Tuomet iš šios duomenų bazės *Kibana* aplinka automatiškai pasiima visus duomenis ir patalpina savo nereleacinėje duomenų bazėje *Elastic*. Kai duomenys atsidūrė *Kibana* aplinkos vidinėje duomenų bazėje, buvo nesunku sukurti reikiamas diagramas nurodant specifinius filtrus ir stulpelių pavadinimus.

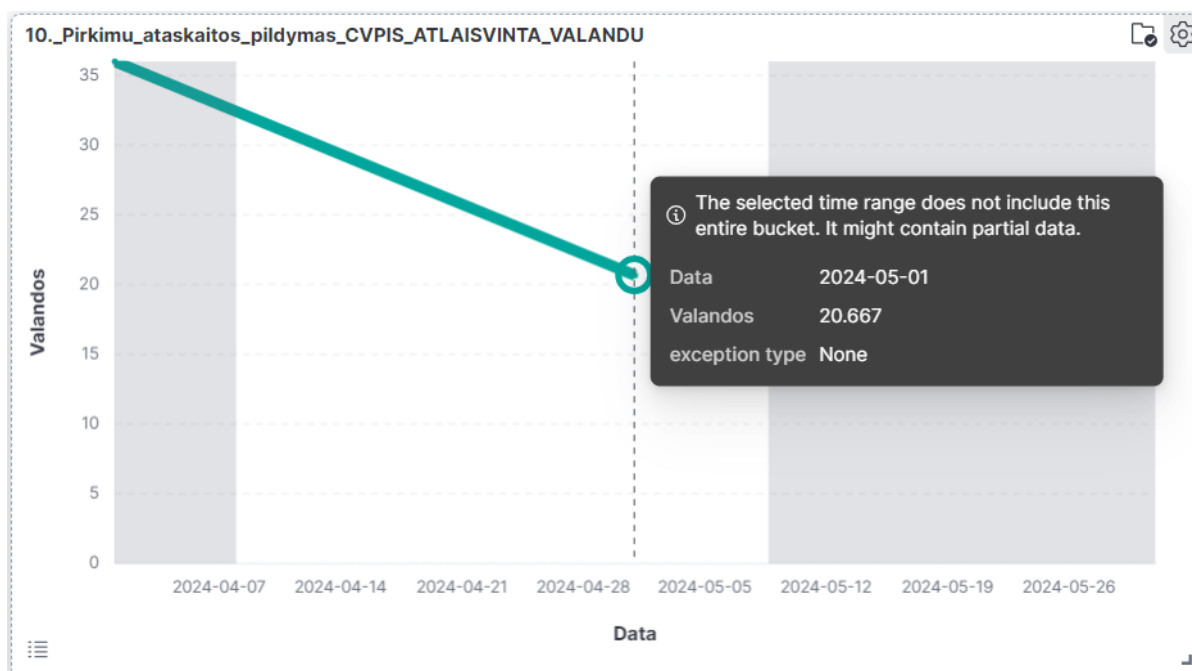
Pirmiausia buvo sukurtos užduočių kiekį padieniui ir pamėnesiui rodančios diagramos. Vizualizacijoje buvo nurodyta, kad informacija būtų imama iš *message* stulpelio su raktine fraze „Main|ProcessTransaction execution ended with“ ir rodytų skirtingus išimties tipus priklausomai nuo gautos žinutės, jeigu klaidos nebuvo, išimties tipas – *none*, jeigu verslo numatyta klaida – *Bussines* ir jeigu nenumatyta sisteminė klaida – *Application*. Abi šias diagramas galima matyti 4 priede ir jose yra pasirinktas laiko periodas nuo 2024-04-08 iki 2024-05-08. Šiose diagramose užvedus pelę ant kurios nors stulpelio spalvos, yra parodoma, koks tiksliai užduočių kiekis yra tame stulpelyje su specifiniu tipu. Jeigu paspaustume ant kurio nors stulpelyje esančio tipo, tuomet programa automatiškai rodytų tik to pasirinkto tipo duomenis.

Kita diagrama buvo sukurta užduočių klaidų ir jų tipų stebėjimui. Ji buvo kuriama panašiai, kaip užduočių kiekio diagramos, tačiau papildomai reikėjo nurodyti klaidos žinutę. Dėl tikslesnio atvaizdavimo, buvo pasirinkta lentelės formos diagrama, kuri atvaizduoja užduoties statuso tipą, klaidos pranešimą ir kiek kartų ji pasikartojo pasirinktame laikotarpyje (4.11 pav.).

KLAIDU_SARASAS		
Export		
Klaidos tip...	Klaidos pranešimas	Klaidų kiekis
Business	Ataskaita užpildyta anksčiau.	5
Application	Nerastas pirkimas CVP IS.	4
Application	Nepatvirtinta	4
Application	Object reference not set to an inst...	3
Business	Nepavyko pasirinkti pasirinkimo pa...	3
Application	Nepavyko rasti ESO DLX sutarties .	2
Application	Activity timeout exceeded	1
Business	Nėra paviešinta sutartis/ Neteising...	1
		23

4.11 pav. Klaidų sąrašo diagrama

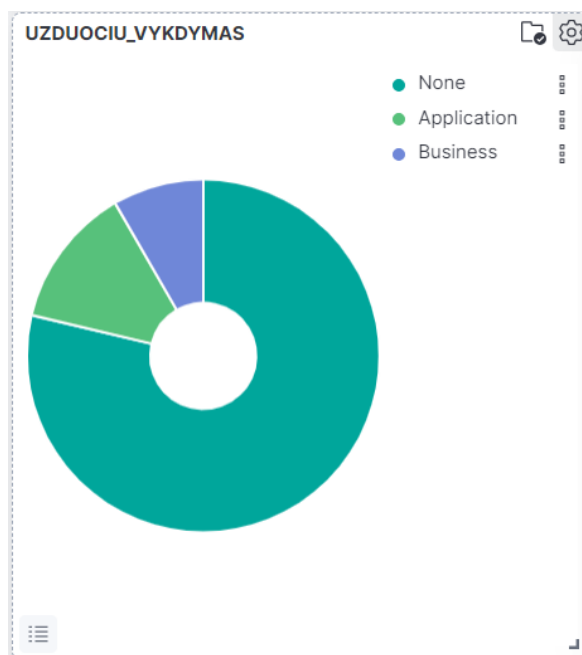
Taip pat reikėjo sukurti diagramą, kuri leistų stebėti automatizuoto proceso sutaupytą laiką. Tam pasiekti buvo sukurta kreivės diagrama, kuri žiūri laiko taupymo kitimą kas mėnesį. Svarbu paminėti, kad šiam sutaupytam laikui yra skaičiuojamos tiksliai sėkmingai atliktos užduotys su iš anksto nustatyta taupymo konstanta. Diagramą galima matyti 4.12 pav.



4.12 pav. Laiko taupymo diagrama

Paskutinė diagrama turi parodyti bendrą užduočių būklių procentinę dalį. Tam atvaizduoti buvo pasirinktas skritulinės diagramos tipas. Šiai diagramai, kaip ir užduočių kiekio sekimui buvo paduodamos užduotys su jų būsenomis, tačiau skaičiavimai atlikti nesumuojant visų užduočių, o ištraukiant jų būsenų dalį. Jeigu užvestume ant kurios nors spalvos, tiksliai matytume kokią

procentinę dalį sudaro kiekviena užduočių būseną, o paspaudus ant pasirinktos spalvos, programa automatiškai rodytų tik to pasirinkto tipo duomenis (4.13 pav).



4.13 pav. Užduočių vykdymo diagrama

Šios diagramos sukurtos pagal anksčiau nustatytus funkcinis reikalavimus ir leidžia visiems asmenims su prieigos teisėmis matyti, kaip robotas atlikinėja užduotis bei kokias klaidas jis daro.

4.8. Proceso kokybės rodiklių palyginimas

Baigus automatizuoto proceso realizavimą ir interaktyvių programų kūrimą buvo įmanoma pasižiūrėti, kaip pakito proceso rodikliai. Deja, nėra kai kurių tikslų duomenų, kai procesas buvo atlikinėjamas žmogaus, nes niekas šių matmenų nematavo. Tačiau galima matyti dabartinį rezultatą po proceso automatizavimo 4.2 lentelėje.

4.2 lentelė. Proceso kokybės rodikliai

Rodiklis	Parametrai prieš automatizavimą	Parametrai po automatizavimo
Užduoties atlikimo laikas	40 minučių	5 minutės
Užduočių pildymo kokybė	Sunku pamatuoti, nes kai kurios vietos gali būti nesupildytos arba supildytos blogai, tačiau tikrindamas žmogus jas vis tiek gali praleisti.	Procesas nėra pilnai paleistas, tačiau sėkmingai atliekama 77 % užduočių ir 5 % neatlikta dėl numatytų galimų žmogaus klaidų. Taip pat likę 18 % buvo nenumatytos sisteminės klaidos.

Užduoties atlikimo laikas buvo gautas iš *UiPath Orchestrator* pateikiamos informacijos, kuri parodo, kiek vidutiniškai laiko robotas užtrunka atlikti vieną užduotį. O užduočių pildymo kokybė yra matoma sukurtų diagramų dėka. Tačiau reikia paminėti, kad proceso užduočių pildymo kokybė turėtų pagerėti, kai bus procese išgaudytos visos smulkios klaidos poros mėnesių laikotarpyje.

5. REZULTATAI IR APIBENDRINIMAS

Baigiamojo bakalaurinio darbo metu buvo analizuojami automatizavimo privalumai, trūkumai bei realizavimo programos. Darbo tikslas buvo ištirti ir automatizuoti pirkimų viešinimo procesą. Šiam tikslui pasiekti buvo išsikelti trys uždaviniai:

1. Darbo pradžioje buvo aptarti automatizavimo privalumai ir trūkumai, kurie galėtų kilti automatizavus viešinimo procesą. Nes jeigu įdėtas laikas automatizavimui neatsipirktų arba dar blogiau pakenktų dabartiniai proceso situacijai, tuomet būtų veltui prarasti resursai. Apsvarsčius privalumus ir trūkumus paaiškėjo, kad automatizuotas viešinimo procesas atneštų ne tik laiko taupymo naudos, tačiau papildomai standartizuotų procesą, kas pagerintų bendradarbiavimą tarp komandų ir padėtų lengviau atrasti iškilusias problemas. Anksčiau šio darbo metu gauti rezultatai leidžia matyti, kad automatizavimas procese sumažintų klaidų kiekį 88 procentais ir taip pat sutaupytų apie 380 valandų per metus. Pagal nustatytus trūkumus, galima matyti, kad programuotojams net ir paleidus procesą, vis tiek reikės jam skirti priežiūros tam tikrais laiko periodais. Tačiau kiekvienam procesui gali reikėti minimalios priežiūros, todėl tai nėra priežastis automatizavimo atšaukimui.
2. Atlikus *UiPath*, *Blue Prism*, *Power Automate* ir *Python* programų analizę, buvo pasirinkta *UiPath* programa kaip pagrindinė programa su kuria bus automatizuojamas procesas. Pagrindiniai bruožai dėl ko buvo pasirinkta ši programa, o ne kitos, yra platus pritaikymas įvairiose sistemose, lengvas naudojimas, gerai paruošta dokumentacija ir greitai suteikiama pagalba bei prieinama kaina, kurią lengva kontroliuoti dėl licencijų tipų. Tačiau, kad procesas būtų vykdomas stabiliau, viešinamų pirkimų tipo nustatymas buvo daromas su *Python* programavimo kalba ir jos siūlomomis bibliotekomis. Nusprendus, kokie įrankiai bus naudojami automatizavimui, buvo nustatyti reikalingi atlikti veiksmai tokie kaip užduočių sudėjimas procesui bei pačio proceso automatizavimas.
3. Procesas turi būti pasikartojantis, standartizuotas bei nustatytos jo numanomos klaidos, kad jis būtų sėkmingai automatizuotas. Darbo metu buvo išanalizuoti šie kriterijai ir nustatyta, kad procesą galima automatizuoti. Toliau automatizuojant procesą buvo sudedamos užduotys robotui, nuskaitomi užduočių duomenys, patikrinami visi gauti duomenys, pildomas *CVPIS*, patvirtinamas sėkmingas užduoties atlikimas arba informuojami atsakingi asmenys apie atmetimą. O kad klientai ir programuotojai galėtų matyti informaciją apie proceso atliekamas užduotis tokią kaip: sutaupyta laikas, užduočių kiekis, atlikimo procentas, buvo padarytos dinaminės vizualizacijos. Šios vizualizacijos buvo sukurtos su *Kibana* programa, kuri leidžia lengvai atvaizduoti jai pateikiamus duomenis pagal sukurtas diagramas.

Šio bakalaurinio darbo išsikelti uždaviniai buvo pasiekti. Realizuotas pirkimų viešinimo automatizuotas proceso pagerins viešųjų pirkimų pildymo kokybę ir sutaupys atsakingų darbuotojų darbo laiką, kuris galės būti išnaudotas atliekant kitas užduotis. Šio proceso automatizavimo metu išbandžiau apjungti dvi skirtingas programas (*UiPath* ir *Python*), kurios dirbdamos kartu padidina tikimybę sėkmingai atlikti užduotį.

Ateityje viešųjų pirkimų automatizavimo procesą galima patobulinti. Pavyzdžiui, pagerinti proceso atlikimo kokybę, kad būtų didesnis procentas sėkmingai atliekamų užduočių ir kelti užduočių informaciją į pirkimų bendrą ataskaitą, kad klientai galėtų matyti visų jų automatizuotų procesų atliekamas užduotis vienoje ataskaitoje.

LITERATŪRA

- [1] González, José, and JG Enríquez. "(PDF) Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study." ResearchGate, 5 February 2020, https://www.researchgate.net/publication/339341848_Robotic_Process_Automation_A_Scientific_and_Industrial_Systematic_Mapping_Study.
- [2] Brandall, Benjamin. "Why Process Standardization Improves Quality, Productivity, and Morale." Process Street, 30 May 2018, <https://www.process.st/process-standardization/>.
- [3] Winkler, Dietmar, and Stefan Biffl. "Improving Quality Assurance in Automation Systems Development Projects." IntechOpen, 23 March 2012, <https://www.intechopen.com/chapters/33278>.
- [4] Schwabe, Henrik, and Fulvio Castellacci. "Automation, workers' skills and job satisfaction." PLOS, 30 November 2020, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0242929>
- [5] Wang, Tiandong, and Xiaoyue Lin. "Preparing Workplaces for Digital Transformation: An Integrative Review and Framework of Multi-Level Factors." Frontiers, 24 February 2021, <https://www.frontiersin.org/articles/10.3389/fpsyg.2021.620766/full>.
- [6] *Lietuvių žodynas*, https://www.lietuviuzodynas.lt/terminai/Standartizacija#google_vignette.
- [7] Badkar, Akshay. "Advantages and Disadvantages of Automation." Simplilearn.com, 4 July 2023, <https://www.simplilearn.com/advantages-and-disadvantages-of-automation-article>.
- [8] Olmstead, Levi. "What Is Process Automation? Examples, Enterprise Benefits." Whatfix, 3 November 2023, <https://whatfix.com/blog/process-automation/>.
- [9] "Blue Prism." SS&C Blue Prism | Docs, 11 April 2024, <https://bpdocs.blueprism.com/en-us/home.htm>.
- [10] "Official Microsoft Power Automate documentation - Power Automate | Microsoft Learn." Learn Microsoft, <https://learn.microsoft.com/en-us/power-automate/>.
- [11] "The Selenium Browser Automation Project." Selenium, 17 November 2023, <https://www.selenium.dev/documentation/>.
- [12] "Top 10 Best RPA Tools in 2024." The Georgia Straight, 12 January 2024, <https://www.straight.com/guides/software/best-rpa-tools/>.
- [13] Ostlick, Nick. "The Benefits and Challenges of RPA Implementation." UiPath, 3 November 2016, <https://www.uipath.com/blog/rpa/the-benefits-and-challenges-of-rpa-implementation>.
- [14] Patel, Parth. "What is Power Automate? Advantages & Disadvantages of Power Automate" Sapphire Software Solutions, 26 May 2022, <https://www.sapphiresolutions.net/blog/what-is-power-automate-advantages-and-disadvantages-of-power-automate>.

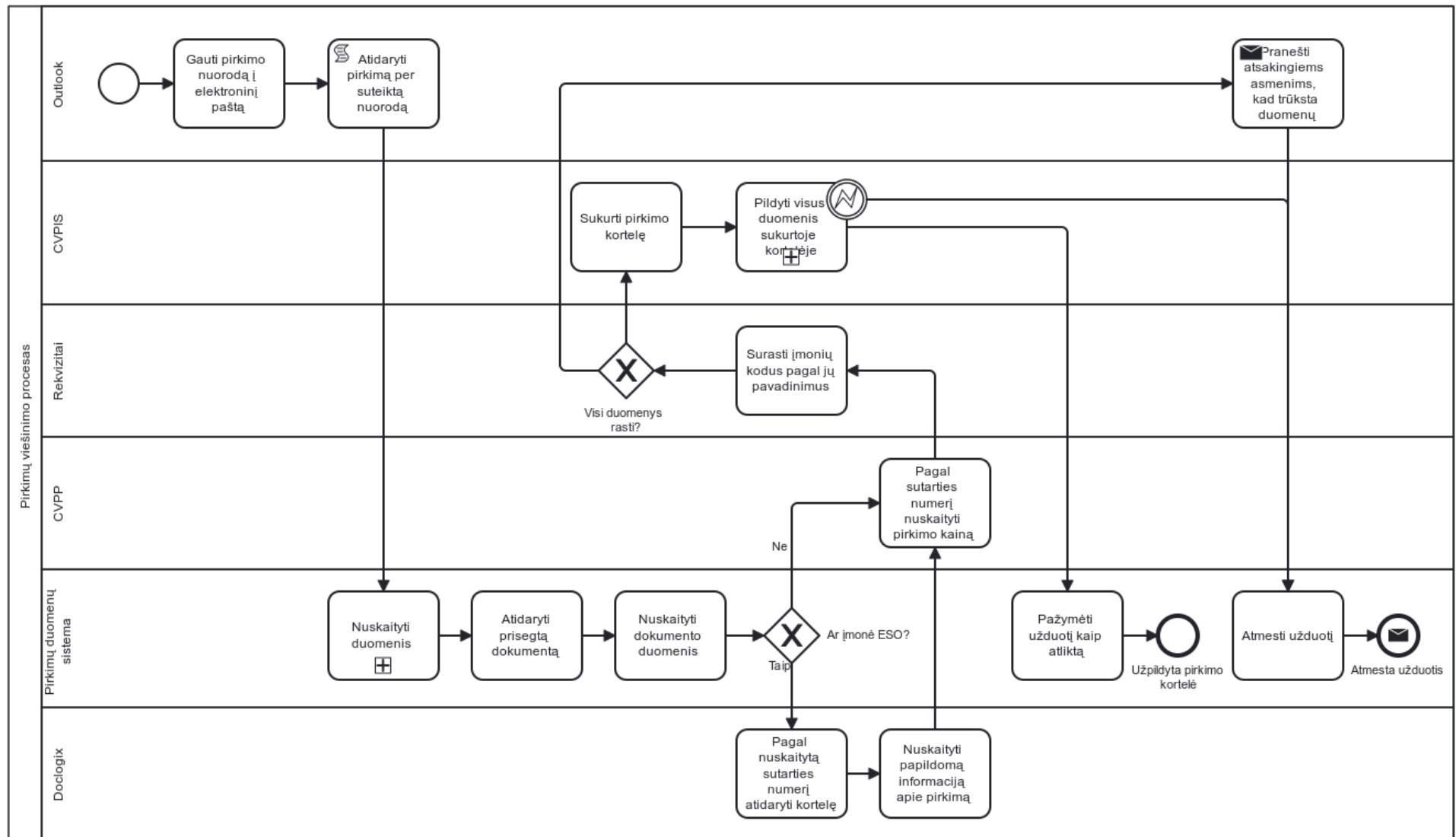
- [15] “Official Microsoft Power Automate documentation - Power Automate | Microsoft Learn.” Learn Microsoft, <https://learn.microsoft.com/en-us/power-automate/>.
- [16] “AI Center - About AI Center.” UiPath Documentation, 18 April 2024, <https://docs.uipath.com/ai-center/automation-cloud/latest/user-guide/about-ai-center>.
- [17] “AI Builder documentation | Microsoft Learn.” Learn Microsoft, <https://learn.microsoft.com/en-us/ai-builder/>.
- [18] UiPath. “Robotic Enterprise Framework.” 26 April 2024, <https://docs.uipath.com/studio/standalone/2023.10/user-guide/robotic-enterprise-framework>.
- [19] Lietuvos Respublikos Seimas. “Lietuvos Respublikos viešųjų pirkimų įstatymas.” 9 November 2017, <https://www.e-tar.lt/portal/lt/legalAct/TAR.C54AFFAA7622/asr>.
- [20] “full HTML Standard.” *HTML Standard*, <https://html.spec.whatwg.org/multipage/>.
- [21] “Automation Business Analyst Foundation.” *UiPath Academy*, 11 September 2023, <https://academy.uipath.com/learning-plans/automation-business-analyst-foundation>.

PRIEDAI

1 PRIEDAS

Bendra informacija apie procesą	
1.Proceso pavadinimas	10._Pirkimu_ataskaitos_pildymas_CVPIS
2.Padaliny, kuriame atliekamas procesas	Padalinio pavadinimas
3.Proceso savininkas	Vardenis Pavardenis
4.Proceso kliento įmonė	Įmonės pavadinimas
5.Procesą kuriojantis asmuo kliento įmonėje	Vardūnas Pavardūnas
6.Kada pradedamas procesas?	Kai supildoma kortelė pirkimų duomenų sistemoje ir priskiriama užduotis robotui
7.Proceso rezultatas	Supildyta pirkimo ataskaita Atn-1 CVP IS`e
8.Vidutinė proceso užduoties atlikimo trukmė	40 min.
9.Proceso užduočių kiekis per metus	630
10. Procesui atlikti reikalingos IT sistemos	MS Outlook; Sharepoint (PDE); MS Excel; MS Word; ESO Doclogix; CVPIS; CVPP; RPA_DB

2 PRIEDAS



3 PRIEDAS

```
import sys
from requests_ntlm import HttpNtlmAuth
import requests
import re

def get_args_from_txt(file_path: str):
    try:
        with open(file_path, 'r') as file:
            first_line = file.readline().strip()
            args = first_line.split()
            return args[0], args[1], args[2], args[3]
    except Exception as e:
        raise Exception("Could not open args txt file: " + str(e))

def get_purchase_type(username, password, url):
    auth = HttpNtlmAuth(username.strip(), password.strip())
    response = requests.get(url, auth=auth)
    purchase_type = re.findall('(<="title:") [A-Ža-ž ]*', response.text)
    print(purchase_type)

    if len(purchase_type) == 1:
        return str(purchase_type[0])
    else:
        return ""

def write_data_to_txt(purchase_type, url, output_path):
    # Open output.txt in write mode
    with open(output_path, 'w') as file:
        file.write(f"{purchase_type}; {url}") # Write the purchase type to file

if __name__ == "__main__":
    if len(sys.argv) == 2:
        temp_args_path = str(sys.argv[1])

        username, password, url, output_path = get_args_from_txt(temp_args_path)

        purchase_type = get_purchase_type(username, password, url)

        write_data_to_txt(purchase_type, url, output_path)
    else:
        print("Invalid number of arguments.")
```

4 PRIEDAS

