



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Kamilė Mikalauskaitė

PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ TAIKYMAS
PROGRAMINĖS ĮRANGOS APIMČIAI ĮVERTINTI

Baigiamasis magistro projektas

Vadovas
doc. dr. L. Čeponienė

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ TAIKYMAS
PROGRAMINĖS ĮRANGOS APIMČIAI ĮVERTINTI

Baigiamasis magistro projektas
Informacinių sistemų inžinerijos studijų programa (kodas 621E15001)

Vadovas

doc. dr. L. Čėponienė
2017–05–22

Recenzentas

doc. dr. K. Kapočius
2017–05–22

Projektą atliko

Kamilė Mikalauskaitė
2017–05–22

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

Kamilė Mikalauskaitė

(Studento vardas, pavardė)

Informacinių sistemų inžinerijos studijų programa, 621E15001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ TAIKYMAS
PROGRAMINĖS ĮRANGOS APIMČIAI ĮVERTINTI“
AKADEMINIO SAŽININGUMO DEKLARACIJA

_____ m. _____ d.
Kaunas

Patvirtinu, kad mano, **Kamilės Mikalauskaitės**, baigiamasis projektas tema „PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ TAIKYMAS PROGRAMINĖS ĮRANGOS APIMČIAI ĮVERTINTI“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Mikalauskaitė, Kamilė. PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ TAIKYMAS PROGRAMINĖS ĮRANGOS APIMČIAI VERTINTI. Magistro baigiamasis projektas / vadovas doc. dr. Lina Čeponienė; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatikos inžinerija, technologijos mokslai

Reikšminiai žodžiai: *Panaudojimo atvejai, programos apimtis, panaudojimo atvejų taškų metodas, UCP, REUCP, EUCP.*

Kaunas, 2017. 81 p.

SANTRAUKA

Įmonėse, kuriančiose programinę įrangą, labai svarbu yra teisingai nustatyti projekto kūrimo apimtį ir kainą. Augant konkurencijai ir norint turėti gerą vardą įmonės, kuriančios programinę įrangą, turėtų atkreipti dėmesį į visus veiksnius: kainą, laiką, kokybę, todėl atsiranda poreikis kurti aukštos kokybės programinę įrangą, turinčią protingą kainos ir kokybės santykį. Norint apskaičiuoti programinės įrangos kūrimo apimtį gali būti taikomas ne vienas metodas: LOC, funkciniai taškai, objektų taškai, panaudojimo atvejų taškai ir pan. Vienos įmonės taiko tam tikrus metodus, kitos netaiko jokių skaičiavimo metodų.

Pagrindinis šio darbo tikslas išanalizuoti panaudojimo atvejų taškų metodikas, pritaikyti apimties skaičiavimo metodikas naudojant *MagicDraw* įrankį, ir pateikti metodikų naudojimo rekomendacijas.

Šiame darbe apžvelgti esami mokslininkų sprendimai, paremti UCP metodu, palyginti esami įrankiai, skaičiuojantys programinės įrangos apimtį, sukurtas ir pritaikytas sprendimas *MagicDraw* programinėje įrangoje, apskaičiuojantis projekto apimtį pagal tradicinį UCP metodą, išplėstąjį UCP metodą ir patikslintąjį UCP metodą. Taip pat atlikta eksperimentinė dalis su keturiais skirtingos apimties ir sudėtingumo projektais.

Remiantis analizės ir eksperimento rezultatais, galima teigti, kad metodikų naudojimas vis dar nėra aiškus, aktorių, panaudojimo atvejų, techninių ir aplinkos faktorių sudėtingumo vertinimas nėra aiškiai apibrėžtas, norint teisingai nustatyti sudėtingumą reikia remtis asmenine metodikų naudojimo patirtimi. Remiantis atliktu eksperimentu, pastebėta, kad metodikų skaičiavimo rezultatai priklauso nuo projekto apimties ir sudėtingumo.

Atliktų eksperimentų patirtis leidžia tikėti, kad PA taškų skaičiavimo variacijas galima pritaikyti projekto planavimo srityse kuo tikslesniam projekto apimties nustatymui, taip pat tai gali prisidėti prie projekto planavimo – paruošus preliminarias panaudojimo atvejų diagramas ir pritaikius metodus galima matyti kokį jie poveikį turės galutinei projekto apimčiai.

Mikalauskaitė, Kamilė. APPLYING USE CASE POINTS TECHNIQUE FOR SOFTWARE EFFORT ESTIMATION: Master's thesis in Information Systems Engineering / supervisor assoc. prof. Lina Čeponienė. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics Engineering, Technology Science

Key words: *use case, software cost estimation, use case point method, UCP, REUCP, EUCP.*

Kaunas, 2017. 81 p.

SUMMARY

It is very important to correctly set the project scope and cost of development in companies, which focuses on creating software. With the growth of competition and in order to have a good company name, the software company should pay attention to all the factors, such as time, quality, cost, and, therefore, the need to create high-quality software with reasonable price-quality ratio. In order to calculate the estimation of software development, there may be applied more than one method: LOC, Functional Points, Object Points, Use Case Points Methods for software costs estimation, and etc. Some of the companies are using certain methods, the others are not applying any methods of calculation.

The main objective of this work is to analyze the use case points methods, adapt the software cost estimation calculation methodologies by using *MagicDraw* tool and to provide recommendations for the use of the methodologies.

In this thesis, there is an overview of existing solutions made by scientists, supported by UCP method; the comparison of the existing tools, that calculates the software costs estimation; created and adapted solution with *MagicDraw* software, which calculates the software cost estimation of the project by traditional UCP method, expanded UCP method and revised UCP method. Also performed an experimental part with four different scope and complexity projects.

On the basis of the analysis and experiment results, it could be argued that the use of the methodologies are still not clear, actors, use cases, technical and environmental complexity factors evaluation are not clearly defined, in order to correctly identify the complexity, there is a need to rely on the personal experience with the use of the methodologies. On the basis of an experiment, it is noted that the results of the methodologies calculations depend on the scope and the complexity of the project.

The experience of the experiments allows to expect that the Use Case Points methods variations may be tailored as far as possible in the project planning in assessing the software cost estimation of the project, as well as to contribute to the planning of the project – by preparing the provisional application of the methods of the Use Case diagrams, you may see how they will impact on the final scope of the project.

TURINYS

Lentelių sąrašas	3
Paveikslų sąrašas	5
Terminų ir santrumpų žodynas	7
1. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO PROBLEMINĖS SRITIES ANALIZĖ	10
1.1. Analizės tikslas	10
1.2. Tyrimo objektas, sritis ir problema	10
1.3. Panaudojimo atvejų taškų metodo analizė	10
1.4. Projekto apimties vertinimo, paremto UCP, žingsniai	12
1.4.1. Aktorių svorio skaičiavimas	12
1.4.2. Panaudojimo atvejų svorio skaičiavimas	12
1.4.3. UUCP (nekeguotų panaudojimo atvejų taškų) skaičiavimas	12
1.4.4. Techninių ir aplinkos faktorių skaičiavimas	13
1.4.5. Panaudojimo atvejų taškų skaičiavimas	14
1.4.6. Apimties nustatymas	14
1.5. Panaudojimo atvejų taškų metodo vartotojai	15
1.6. Esamų panaudojimo atvejų taškų metodikų analizė	16
1.6.1. Mokslininkų sukurtas U–EST įrankis	16
1.6.2. <i>Sparx Enterprise Architect</i> (EA) projekto apimties vertinimas	17
1.6.3. <i>MagicDraw</i> – projekto apimties vertinimo ataskaita	18
1.6.4. Panaudojimo atvejų taškų skaičiuotuvas	20
1.6.5. <i>Excel</i> skaičiuotuvas (verslo analitiko – Jono Hanseno)	21
1.6.6. Kitų mokslininkų tyrimų apžvalga	22
1.7. Siekiamo sprendimo apibrėžimas	24
1.8. Analizės išvados	25
2. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO SPRENDIMO REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS	26
2.1. Reikalavimų specifikacija	26
2.2. Dalykinės srities modelis	31
2.3. Vartotojo sąsajos modelis	32
2.4. Nefunkciniai reikalavimai	33
2.5. UCP skaičiavimo įrankio projektas	33
2.5.1. Projekto tikslas	33
2.5.2. Metodų taikymo etapai	33
3. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO REALIZACIJOS PROJEKTAS	38
3.1. Projekto įgyvendinimas <i>MagicDraw</i> programinėje įrangoje	38
3.1.1. Panaudojimo atvejų taškų metodo profilis realizuotas <i>MagicDraw</i> programinėje įrangoje	38

3.1.2. Sąrašai (angl. <i>enumerations</i>).....	38
3.1.3. Stereotipai (angl. <i>stereotype</i>)	40
3.1.4. Rezultatų ataskaitos	40
4. PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ REALIZACIJA IR TESTAVIMAS	42
4.1. Sprendimo realizacijos ir veikimo aprašas	42
5. EKSPERIMENTINIS PANAUDOJIMO ATVEJŲ TAŠKŲ METODŲ TAIKYMO TYRIMAS	50
5.1. Eksperimento planas	50
5.1.1. Projekto „Pardavimai“ eksperimento planas	50
5.1.2. Projekto „Pranešimai“ eksperimento planas.....	51
5.1.3. Projekto „Banko importas“ eksperimento planas	51
5.1.4. Projekto „Valiutos“ eksperimento planas	52
5.2. Eksperimento rezultatai	52
5.2.1. Projekto „Pardavimai“ eksperimento rezultatai.....	53
5.2.2. Projekto „Pranešimai“ eksperimento rezultatai	55
5.2.3. Projekto „Banko importas“ eksperimento rezultatai	58
5.2.4. Projekto „Valiutos“ eksperimento rezultatai	61
5.3. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas	63
5.4. Sprendimo taikymo rekomendacijos.....	64
6. REZULTATŲ APIBENDRINIMAS IR IŠVADOS.....	66
7. LITERATŪRA.....	67
8. PRIEDAI.....	68
8.1. Priedas. Sprendimo eksperimento PA aprašymai ir veiklos diagramos	68
8.1.1. Projektas „Pardavimai“	68
8.1.2. Projektas „Pranešimai“.....	71
8.1.3. Projektas „Banko importas“	72
8.1.4. Projektas „Valiutos“	75

LENTELIŲ SĄRAŠAS

1.1 lentelė UCP aktorių svoriai	12
1.2 lentelė Panaudojimo atvejų svorio faktoriai.....	12
1.3 lentelė Techniniai faktoriai	13
1.4 lentelė Aplinkos faktoriai.....	14
1.5 lentelė Palyginimo lentelė (Įrankiai).....	22
1.6 lentelė Mokslininkų tyrimo rezultatų lentelė	24
2.1 lentelė. PA „Apskaičiuoti UCP“ formalus aprašymas	27
2.2 lentelė PA „Pasirinkti UCP metoda“ formalus aprašymas	28
2.3 lentelė PA „Pasirinkti UCP metoda“ formalus aprašymas	28
2.4 lentelė PA „Apskaičiuoti panaudojimo atvejų sudėtingumą“ formalus aprašymas.....	29
2.5 lentelė PA „Apskaičiuoti techninius faktorius“ formalus aprašymas	30
2.6 lentelė PA “ Apskaičiuoti aplinkos faktorius“ formalus aprašymas	31
2.7 lentelė Nefunkciniai reikalavimai	33
2.8 lentelė Aktorių svorio koeficientai pagal skirtingus metodus.....	33
2.9 lentelė Panaudojimo atvejų svorio koeficientai pagal skirtingus metodus	34
2.10 lentelė Techninių faktorių svorio koeficientai pagal skirtingus metodus	35
2.11 lentelė Aplinkos faktorių svorio koeficientai pagal skirtingus metodus.....	36
5.1 lentelė Projekto „Pardavimai“ aktorių klasifikavimas	53
5.2 lentelė Projekto „Pardavimai“ aktorių svoriai	53
5.3 lentelė Projekto „Pardavimai“ panaudojimo atvejų klasifikavimas.....	53
5.4 lentelė Projekto „Pardavimai“ panaudojimo atvejų svoriai	54
5.5 lentelė Projekto „Pardavimai“ panaudojimo atvejų taškai.....	54
5.6 lentelė Projekto „Pardavimai“ techninių faktorių klasifikavimas	54
5.7 lentelė Projekto „Pardavimai“ techninių faktorių svoriai	54
5.8 lentelė Projekto „Pardavimai“ techninių faktorių sudėtingumas	54
5.9 lentelė Projekto „Pardavimai“ aplinkos faktorių klasifikavimas	54
5.10 lentelė Projekto „Pardavimai“ aplinkos faktorių svoriai.....	55
5.11 lentelė Projekto „Pardavimai“ aplinkos faktorių sudėtingumas	55
5.12 lentelė Projekto „Pardavimai“ galutiniai skaičiavimai	55
5.13 lentelė Projekto „Pranešimai“ aktorių klasifikavimas	55
5.14 lentelė Projekto „Pranešimai“ aktorių svoriai	56
5.15 lentelė Projekto „Pranešimai“ panaudojimo atvejų sudėtingumo klasifikacija	56
5.16 lentelė Projekto „Pranešimai“ panaudojimo atvejų svoris.....	56
5.17 lentelė Projekto „Pranešimai“ panaudojimo atvejų sudėtingumas	56
5.18 lentelė Projekto „Pranešimai“ techninių faktorių klasifikacija	56
5.19 lentelė Projekto „Pranešimai“ techninių faktorių svoris	57
5.20 lentelė Projekto „Pranešimai“ techninių faktorių sudėtingumas.....	57
5.21 lentelė Projekto „Pranešimai“ aplinkos faktorių klasifikavimas.....	57
5.22 lentelė Projekto „Pranešimai“ aplinkos faktorių svoris	57
5.23 lentelė Projekto „Pranešimai“ aplinkos faktorių sudėtingumas	57
5.24 lentelė Galutiniai projekto „Pranešimai“ rezultatai	58
5.25 lentelė Projekto „Banko importas“ aktorių klasifikacija.....	58
5.26 lentelė Projekto „Banko importas“ aktorių svoriai	58
5.27 lentelė Projekto „Banko importas“ panaudojimo atvejų klasifikacija	58
5.28 lentelė Projekto „Banko importas“ panaudojimo atvejų svoriai.....	59
5.29 lentelė Projekto „Banko importas“ panaudojimo atvejų sudėtingumas.....	59
5.30 lentelė Projekto „Banko importas“ techninių faktorių klasifikavimas.....	59
5.31 lentelė Projekto „Banko importas“ techninių faktorių svoriai	59
5.32 lentelė Projekto „Banko importas“ techninių faktorių sudėtingumas.....	59

5.33 lentelė Projekto „Banko importas“ aplinkos faktorių klasifikavimas.....	60
5.34 lentelė Projekto „Banko importas“ techninių faktorių svoriai.....	60
5.35 lentelė Projekto „Banko importas“ techninių faktorių sudėtingumas.....	60
5.36 lentelė Galutiniai projekto „Banko importas“ rezultatai.....	60
5.37 lentelė Projekto „Valiutos“ aktorių klasifikavimas.....	61
5.38 lentelė Projekto „Valiutos“ aktorių svoriai.....	61
5.39 lentelė Projekto „Valiutos“ panaudojimo atvejų sudėtingumo klasifikavimas.....	61
5.40 lentelė Projekto „Valiutos“ panaudojimo atvejų svoris.....	61
5.41 lentelė Projekto „Valiutos“ panaudojimo atvejų taškai.....	61
5.42 lentelė Projekto „Valiutos“ techninių faktorių reikšmės.....	62
5.43 lentelė Projekto „Valiutos“ techninių faktorių svoris.....	62
5.44 lentelė Projekto „Valiutos“ techninių faktorių sudėtingumas.....	62
5.45 lentelė Projekto „Valiutos“ aplinkos faktorių reikšmės.....	62
5.46 lentelė Projekto „Valiutos“ aplinkos faktorių svoriai.....	62
5.47 lentelė Projekto „Valiutos“ aplinkos faktorių sudėtingumas.....	63
5.48 lentelė Projekto „Valiutos“ galutiniai skaičiavimai.....	63
5.49 lentelė Rezultatų palyginimas pagal metodus.....	63
5.50 lentelė Rezultatų palyginimas, pagal metodus.....	64
8.1 lentelė PA „Sukurti pardavimą“, „Redaguoti pardavimą“ formalus aprašymas.....	68
8.2 lentelė PA „Ištrinti pardavimą“ formalus aprašymas.....	69
8.3 lentelė PA „Registruoti iVaz“ formalus aprašymas.....	69
8.4 lentelė PA „Panaikinti iVaz registraciją“ formalus aprašymas.....	70
8.5 lentelė PA „Importuoti pardavimą“ formalus aprašymas.....	70
8.6 lentelė PA „Įvesti apmokėjimą“ formalus aprašymas.....	70
8.7 lentelė Panaudojimo atvejo „Sukurti pranešimą“ ir „Redaguoti pranešimą“ formalus aprašymas.....	71
8.8 lentelė PA „Filtruoti pranešimus“ formalus aprašymas.....	71
8.9 lentelė PA „Sukonfigūruoti elektroninį paštą“ formalus aprašymas.....	72
8.10 lentelė PA „Importuoti banką“ formalus aprašymas.....	72
8.11 lentelė PA „Importuoti PayPal“ formalus aprašymas.....	73
8.12 lentelė PA „Sudengti“ formalus aprašymas.....	73
8.13 lentelė PA „Filtruoti“ formalus aprašymas.....	74
8.14 lentelė PA „Įkelti komisinius“ formalus aprašymas.....	74
8.15 lentelė PA „Priskirti A/S“ formalus aprašymas.....	75
8.16 lentelė PA „Sukurti valiutą“ formalus aprašymas.....	75
8.17 lentelė PA „Trinti valiutą“ formalus aprašymas.....	76
8.18 lentelė PA „Importuoti valiutą“ formalus aprašymas.....	76

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Panaudojimo atvejų diagrama, iliustruojanti projekto vadovą, vertinantį projekto apimtį....	16
1.2 pav. <i>Sparx Enterprise Architect</i> programos sąsaja	18
1.3 pav. <i>MagicDraw</i> metodo taikymas	18
1.4 pav. Aktorius <i>MagicDraw</i> sugeneruotoje ataskaitoje	19
1.5 pav. Aktorių svoriai sugeneruotoje <i>MagicDraw</i> ataskaitoje	19
1.6 pav. Panaudojimo atvejai sugeneruotoje <i>MagicDraw</i> ataskaitoje	19
1.7 pav. Panaudojimo atvejų klasifikavimas sugeneruotoje <i>MagicDraw</i> ataskaitoje.....	19
1.8 pav. Techninių faktorių svoriai	20
1.9 pav. Aplinkos faktorių svoriai.....	20
1.10 pav. Projekto apimties vertinimas	20
1.11 pav. Panaudojimo atvejų taškų skaičiuotuvą interneto naršyklėje	21
1.12 pav. Aktorių ir panaudojimo atvejų vertinimas	21
1.13 pav. Techninių ir aplinkos faktorių vertinimas	21
1.14 pav. Skaičiavimai ir rezultatai.....	22
2.1 pav. UCP skaičiavimo metodo atvejų diagrama	26
2.2 pav. „Apskaičiuoti UCP“ veiklos diagrama.....	26
2.3 pav. „Pasirinkti UCP metodą“ veiklos diagrama	27
2.4 pav. Veiklos diagrama „Apskaičiuoti aktorių sudėtingumą“	28
2.5 pav. „Apskaičiuoti panaudojimo atvejų sudėtingumą“ veiklos diagrama	29
2.6 pav. „Apskaičiuoti techninius faktorius“ veiklos diagrama.....	30
2.7 pav. „Aplinkos faktorių apskaičiavimas“	31
2.8 pav. UCP dalykinės srities modelis.....	32
2.9 pav. <i>MagicDraw</i> vartotojo sąsaja.....	32
2.10 pav. Sugeneruotos rezultatų ataskaitos fragmentas.....	33
3.1 pav. Panaudojimo atvejų taškų metodo profilis	38
3.2 pav. Aktorių ir panaudojimo atvejų sudėtingumo sąrašai.....	39
3.3 pav. Techninių ir aplinkos faktorių sąrašai	40
3.4 pav. Stereotipai skirti darbui su aktoriais ir panaudojimo atvejais.	40
3.5 pav. Metodų ataskaitos fragmentas	41
4.1 pav. Metodo naudojimo <i>MagicDraw</i> aplinkoje vedlys	42
4.2 pav. Metodo naudojimo <i>MagicDraw</i> aplinkoje vedlys Nr. 2.....	43
4.3 pav. Metodo naudojimo <i>MagicDraw</i> aplinkoje vedlys Nr. 3.....	43
4.4 pav. Aktorių sudėtingumo pasirinkimas	44
4.5 pav. Panaudojimo atvejų sudėtingumo pasirinkimas.....	44
4.6 pav. Aplinkos ir techniniai faktoriai	45
4.7 pav. Techninių ir aplinkos faktorių reikšmių priskyrimas	45
4.8 pav. Projekto ataskaitos duomenų saugojimas.....	46
4.9 pav. Ataskaitos duomenų priskyrimas prie šablono.....	46
4.10 pav. Ataskaitos generavimo žingsnis nr. 1	46
4.11 pav. Ataskaitos generavimo žingsnis nr. 2.....	47
4.12 pav. Ataskaitos generavimo žingsnis nr. 3.....	47
4.13 pav. Ataskaitos generavimo žingsnis nr. 4.....	48
4.14 pav. Ataskaitos generavimo žingsnis nr. 5.....	48
4.15 pav. Panaudojimo atvejų taškų metodo bendros ataskaitos fragmentas, rodantis aktorių ir panaudojimo atvejų tarpinius rezultatus	49
4.16 pav. Panaudojimo atvejų taškų metodo bendros ataskaitos fragmentas, rodantis galutinius rezultatus	49
5.1 pav. Projekto „Pardavimai“ panaudojimo atvejų diagrama	50
5.2 pav. Projekto „Pranešimai“ panaudojimo atvejų diagrama	51

5.3 pav. Projekto „Banko importas“ panaudojimo atvejų diagrama.....	52
5.4 pav. Projekto „Valiutos“ panaudojimo atvejų diagrama.....	52
5.5 pav. Projekto „Pardavimai“ eksperimento rezultatų grafikas	55
5.6 pav. Projekto „Pranešimai“ eksperimento rezultatai.....	58
5.7 pav. Projekto „Banko importas“ eksperimento rezultatai.....	60
5.8 pav. Projekto „Valiutos“ eksperimento rezultatų grafikas.....	63
8.8.1 pav. Panaudojimo atvejo „Sukurti pardavimą“ ir „Redaguoti pardavimą“ veiklos diagrama	68
8.2 pav. Panaudojimo atvejo „Ištrinti pardavimą“ veiklos diagrama.	69
8.3 pav. PA „Registruoti iVaz“ ir „Panaikinti iVaz registraciją“ veiklos diagramos	69
8.4 pav. PA „Importuoti pardavimą“ veiklos diagrama.....	70
8.5 pav. PA „Sukurti apmokėjimą“ veiklos diagrama	70
8.6 pav. Panaudojimo atvejo „Sukurti pranešimą“ veiklos diagrama.....	71
8.7 pav. Panaudojimo atvejo „Filtruoti pranešimus“ veiklos diagrama.....	72
8.8 pav. Panaudojimo atvejo „Sukonfigūruoti elektroninį pašta“ veiklos diagrama	72
8.9 pav. Panaudojimo atvejo „Importuoti banką“ veiklos diagrama	73
8.10 pav. Panaudojimo atvejo „Importuoti PayPal“ veiklos diagrama.....	73
8.11 pav. Panaudojimo atvejo „Sudengti“ veiklos diagrama.....	74
8.12 pav. Panaudojimo atvejo „Filtruoti“ veiklos diagrama.....	74
8.13 pav. Panaudojimo atvejo „Įkelti komisinius“ veiklos diagrama	75
8.14 pav. Panaudojimo atvejo „Priskirti A/S“ veiklos diagrama.....	75
8.15 pav. Panaudojimo atvejo „Sukurti valiutą“ veiklos diagrama	76
8.16 pav. Panaudojimo atvejo „Trinti“ veiklos diagrama	76
8.17 pav. Panaudojimo atvejo „Importuoti “ veiklos diagrama.....	77

TERMINŲ IR SANTRUMPŲ ŽODYNAS

1. UPC – Panaudojimo atvejų taškų metodas.
2. EUCP – Išplėstas panaudojimo atvejų taškų metodas.
3. Re-UCP – Patikslintas panaudojimo atvejų taškų metodas.
4. Aktorius – vienas iš pagrindinių panaudojimo atvejų diagramos elementų. Tai gali būti tiek žmogus, tiek sistema. Aktorius gali vaizduoti: sistemos vartotoją, įmonę, organizacinį vienetą, išorinę programų sistemą, automatinį įrenginį ir t.t.
5. UUCP – nekoreguotas panaudos atvejo taškas.
6. TCF – techninis kompleksiskumo faktorius.
7. ECF – aplinkos kompleksiskumo faktorius.
8. UUCW – nekoreguotas panaudos atvejo SVORIS.
9. UAW – nekoreguotas aktoriaus svoris.
10. PF – produktyvumo aktorius.
11. GUI – grafinė vartotojo sąsaja.
12. PA – Panaudojimo atvejis.
13. LOC – kodo eilučių kiekis.

IVADAS

Šiomis dienomis programinės įrangos atsiradimas nebėra toks stebinantis ir sunkus procesas, koks buvo prieš 10 metų. Yra nemažai tiek stambių, tiek mažesnių įmonių, kuriančių programinę įrangą, todėl, nenuostabu, kad šioje srityje yra išaugusi konkurencija, klientai turi didelį pasirinkimą, koki programinės įrangos tiekėją jiems pasirinkti, jie gali rinktis pagal kainą, kokybę, greitį ar kitus veiksnius, tačiau nenuostabu, kad pigiai ir greitai, tai nebus kokybiškai, o ganėtinai greitai ir kokybiškai, nebus pigiai. Augant konkurencijai, bei norint turėti gerą vardą, įmonės, kuriančios programinę įrangą, turėtų atkreipti dėmesį į visus veiksnius: kainą, laiką, kokybę, todėl atsiranda poreikis kurti aukštos kokybės programinę įrangą, turinčią protingą kainos ir kokybės santykį.

Įmonėms, kuriančioms programinę įrangą, programos apimties vertinimas dažnai vis dar yra iššūkis. Programinės įrangos apimtį gali būti sunku vertinti dėl didelio kiekio detalizuotos informacijos pradinėje projekto kūrimo stadijoje. Norint apskaičiuoti programinės įrangos kūrimo apimtį gali būti taikomas ne vienas metodas: LOC, funkciniai taškai, objektų taškai, panaudojimo atvejų taškai (UCP) ir pan. LOC metodas remiasi programos kodo eilučių skaičiavimu. Funkcinių taškų metodas yra paremtas programos charakteristikų kombinacijomis: įėjimų, išėjimų kiekiu, vartotojo sąveikų kiekiu, išorinių sąsajų kiekiu ir sistemoje naudojamų failų kiekiu. Objektų taškų metodas skaičiuoja objektus (langus, ataskaitas, modulius) ir įvertina jų sudėtingumą: paprastu, vidutiniu ar sudėtingu. Panaudojimo atvejų taškų metodas projekto apimtį apskaičiuoja remiantis UML panaudojimo atvejų diagramomis, vertinant panaudojimo atvejų ir aktorių sudėtingumą, paprastu, vidutiniu ir sudėtingu, techninius faktorius, kurie nusako vartotojo lūkesčius programinei įrangai ir aplinkos faktorius, kurie nusako komandos galimybes ir patirtį. Pagrindinis UCP metodo privalumas yra tas, kad jis yra paremtas panaudojimo atvejų diagramomis, aprašytomis UML kalba, kurios yra priimtinos ir plačiai naudojamos kuriant programinę įrangą. Panaudojimo atvejų taškų metodas yra nuolat tobulinamas, todėl yra ne viena metodo variacija. Panaudojimo atvejų diagramos yra kuriamos ankstyvoje projekto stadijoje, todėl ir metodą galima taikyti pirmame projekto kūrimo etape.

Vienos įmonės taiko tam tikrus metodus, kitos netaiko jokių skaičiavimo metodų, dėl ko dažnai projektai vėluoja, išauga jų kaina ar krenta įmonės reputacija. Šiame darbe bus analizuojamos panaudojimo atvejų taškų metodų variacijos, siekiama sukurti skirtingų metodikų naudojimo rekomendacijas, taikant metodus skirtingo sudėtingumo ir apimties projektams, ir automatiniams skaičiavimams reikalingos informacijos surinkimą iš UML specifikacijomis aprašytų panaudojimo atvejų diagramų.

Šio tyrimo tikslas yra palengvinti panaudojimo atvejų taškų tradicinio, išplėsto ir patikslinto metodikų taikymą, sukuriant metodikų naudojimo rekomendacijas ir pritaikant įrankius.

Šio tyrimo uždaviniai yra:

1. išanalizuoti panaudojimo atvejų taškų taikymo metodikas;
2. išanalizuoti esamų UCP metodų sprendimus;
3. sukurti UCP metodikų sprendimą;
4. automatizuoti metodikų taikymui reikalingos informacijos surinkimą iš panaudojimo atvejų diagramų, aprašytų UML kalba;
5. atlikti UCP metodikų eksperimentinį tyrimą;
6. apibendrinti tyrimo rezultatus;

Tyrimo rezultatai – panaudojimo atvejų taškų metodikų rekomendacijų pateikimas ir *MagicDraw* įrankio galimybių pritaikymas, siekiant palengvinti panaudojimo atvejų taškų metodikų taikymą programavimo darbų apimties vertinimui. Dabar viena iš pagrindinių problemų yra ta, kad nėra vieningų rekomendacijų, nusakančių, kaip reikėtų naudoti panaudojimo atvejų taškų skaičiavimo metodus, todėl rekomendacijų surinkimas ir jų tyrimas turės didelę svarbą, kaip projekto apimties vertinimo stadijoje reikėtų teisingai vertinti aktorių ir panaudojimo atvejų svorius, kaip teisingai nustatyti techninių ir aplinkos faktorių reikšmes, kokį metodą tiksliausiai pasirinkti skirtingos apimties ir sudėtingumo projektams.

Darbas susideda iš įvado, penkių skyrių, išvadų, literatūros sąrašo ir priedų. Darbo apimtis – 68 psl., priedas – 7psl. Pirmajame skyriuje aprašomas tyrimo tikslas, problematika, tradicinis panaudojimo atvejų taškų metodas, esamų mokslininkų sprendimų analizė – išplėstas ir patikslintas UCP metodas. Antrame skyriuje aprašoma panaudojimo atvejų taškų metodo sprendimo reikalavimų specifikacija, dalykinės srities modelis, vartotojo sąsajos modelis bei panaudojimo atvejų taškų skaičiavimo įrankio projektas. Trečiame skyriuje yra aprašomas panaudojimo atvejų taškų metodo realizacijos projektas, projekto įgyvendinimas *MagicDraw* programinėje įrangoje. Ketvirtas skyrius susideda iš projekto apimties skaičiavimo sprendimo aprašymo, naudojant panaudojimo atvejų taškų metodikas. Penktame skyriuje yra aprašoma panaudojimo atvejų taškų metodikos eksperimentinė dalis, kurioje buvo pasirinkti keturi skirtingos apimties ir sudėtingumo projektai, eksperimentinė dalis padėjo sukurti metodikų naudojimo rekomendacijas.

1. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO PROBLEMINĖS SRITIES ANALIZĖ

1.1. Analizės tikslas

Šio darbo analizės tikslas yra ištirti panaudojimo atvejų taškų metodikų taikymo galimybes. Išanalizuoti panaudojimo atvejų metodus taikant gautus mokslininkų rezultatus, juos apžvelgti, palyginti ir pateikti išvadas.

1.2. Tyrimo objektas, sritis ir problema

Šio tyrimo objektas – panaudojimo atvejų taškų metodo taikymas programinės įrangos kūrimo projektams, kurie naudoja panaudojimo atvejų diagramą.

Tyrimo sritis – programinės įrangos apimties matavimo priemonės ir metodai.

Tyrimo problema yra ta, kad vis dar nėra vieningų rekomendacijų, kaip naudoti panaudojimo atvejų metodikas, programinės įrangos apimčiai apskaičiuoti, dalis skaičiavimams reikalingos informacijos yra suvedama rankiniu būdu, nors galėtų būti surenkama iš UML specifikacijos.

1.3. Panaudojimo atvejų taškų metodo analizė

Šiais laikais programinė įranga yra laikoma brangiausia kompiuterių sistemos dalimi. Sėkmingas programinės įrangos projektas yra tas, kuris neperžengia tam tikro iš anksto nustatyto plano kainai ir projekto trukmei. Perdėtai didelė programos kaina gali būti žalinga gamintojui. Kainos apskaičiavimas dešimtmečiais buvo didelė problema sistemos analitikams, projektų vadovams ir programinės įrangos inžinieriams. Gebėjimas nustatyti tikrąją programinės įrangos projekto kainą padeda vadybininkams tiksliai nustatyti realią programinės įrangos kainą.

Kuriant programinės įrangos projektus pirmiausia yra analizuojami ankstesni projektai, kurie yra parduodami kaip produktas. Programinės įrangos kainos apskaičiavimas dažniausiai yra sudėtingas. Šie projektai anksčiau nebūdavo suprantami, kadangi dažnai idėjos tarp kliento įgeidžių ir vadybininko suvokimo skirdavosi. Palaipsniui augant taikomųjų programų dydžiui ir svarbai, lygiagrečiai augo ir programinės įrangos kūrimo kaina. Dėl šios priežasties programinės įrangos planavimo kaina būtų perdėtai didelė. Ankstesniais metais buvo pristatyti įvairūs metodai, kurių paskirtis įvertinti galimą programinės įrangos projekto kainą. Kai kurie programinės įrangos vystymo projektai žlunga dėl prasto projektų valdymo. Pagrindiniai projektų valdymo tikslai yra paskaičiuoti, kas apskaičiuojama, pamatuoti, kas pamatuojama. Geras projekto vertinimo metodas leidžia optimaliai panaudoti resursus ir pakelia galutinio projekto kokybę, dėl ko klientas lieka labiau patenkintas [5].

Didėjant projekto apimčiai – didėja ir projekto trukmė, žmogiškųjų išteklių ir finansinės išlaidos. Klientai turi vienintelį tikslą – lengvai ir tiksliai su programa atlikti jiems rūpimas užduotis. Dauguma būsimų vartotojų projekto pradžioje nepajėgūs apibrėžti visų savo lūkesčių, susijusių su programinės įrangos realizacija. Iš kitos pusės, preliminarus kainos įvertinimas yra svarbus faktorius, kuris paveikia programinės įrangos kainą. Programos dydžiui įvertinti yra naudojami keli metodai. Vienas svarbiausių yra funkcijų taškas (angl. *Function Point, FP*). [2]

Sudėtingumas yra vienas iš svarbiausių veiksnių, kuris turėtų būti apsvarstytas skaičiuojant programinės įrangos kainą (angl. *Software Cost Estimation, SCE*). Projekto sudėtingumas gali būti apskaičiuotas remiantis projekto faktoriais. Svarbu paminėti, jog sudėtingumo faktorius nepriklauso nuo projekto dydžio. Nedidelis projektas gali būti sudėtingesnis negu didelės apimties projektas. Pavyzdžiui, 3D modelio programavimas yra sudėtingesnis negu teksto redaktoriaus kūrimas. Sudėtingumas taip pat gali būti reliatyvus veiksnys, pavyzdžiui, komercinis internetinis projekto realizavimas gali būti sudėtingas procesas tam, kas nėra to programavęs anksčiau. Šis faktorius yra pagrįstas vartotojų paklausa. Tai reiškia, jog norint patenkinti kliento norus pagaminti produktą laiku, tenka padidinti darbuotojų skaičių ir tai reikalauja daugiau pastangų. Dėl šios priežasties,

projektų vadovai privalės skirti daugiau laiko komunikacijai ir realizuotų projekto dalių sąryšiui įgyvendinti.

Yra ne vienas kainos vertinimo metodas, pvz., COCOMO ar FPA (angl. *function point analysis*), kurie yra naudojami programinės įrangos kūrimo industrijoje. FPA metodo trūkumas yra tas, kad norint apskaičiuoti funkcinius taškus yra reikalingi patyrę specialistai.

1993 metais mokslininkas Gustavas Karneris sukūrė objektyvų apimties ir vertinimo metodą, kurį pavadino panaudojimo atvejų taškų (angl. *Use Case Point*) metodu, skirtu vertinti programinės įrangos kūrimo apimtį anksčiausiose programinės įrangos kūrimo stadijose. Norint naudoti panaudojimo taškų metodą turi būti supratimas apie problemos sritį, sistemos dydį ir architektūrą.

Norint gauti informaciją anksčiausiose programinės įrangos kūrimo stadijose daugiausia yra naudojamos dvi UML tipo diagramos: panaudojimo atvejų diagrama ir klasių diagrama.

Populiariausi IT projektų apimties vertinimo metodai [7]:

1. „ekspertų sprendimas“ (angl. *Expert Judgement*);
2. kodo eilučių skaičius (*LOC*);
3. funkcinių taškų vertinimas (*FP*);
4. panaudojimo atvejų taškų vertinimas (*UCP*);

„Ekspertų sprendimo“ metodas yra vykdomas bendraujant su dalykinės srities ekspertais. Šis metodas yra taikomas, kai jau buvo vystyti panašūs projektai, atsižvelgiant į juos. Problemos iškylančios naudojant šį metodą, yra tai, kad sudėtinga rasti patyrusių profesionalų, galinčių teisingai vertinti savo lygį, todėl šis metodas yra laikomas subjektyviu.

Kodo eilučių skaičiavimo metodą naudoja didelę patirtį turinčios komandos. Šis metodas turi taisyklę, kad Y eilučių kiekį galima parašyti per X valandų. Pavyzdžiui, 100 valandų 1000 kodo eilučių. Vienas iš pagrindinių metodo trūkumų yra tas, kad turime iš anksto žinoti apytikslį eilučių kiekį kuriant produktą, jei labai panaši sistema jau buvo kurta, tai apskaičiuoti gali nebūti labai sunku, tačiau kitais atvejais tai yra ganėtinai sudėtinga. Kitas trūkumas yra tas, kad, norint apskaičiuoti kodo eilučių kiekį, taip pat palyginti su kitomis panašiomis sistemomis, reikia turėti didelę patirtį turinčią komandą.

Funkcinių taškų skaičiavimo metodas yra sukurtas IBM kompanijos. Jis skaičiuoja funkcinių reikalavimų elementus. Šis metodas yra naudojamas ganėtinai plačiai ir yra patikimas. Metodo privalumai yra tie, kad tai yra faktais paremtas metodas, kuris gali būti naudojamas komandos, kuri nėra patyrusi toje srityje. Taip pat projekto kaina gali būti paremta prieš tai buvusių projektų kainomis.

Kiekvienas metodas yra naudojamas skirtingomis sistemos kūrimo stadijomis. Pavyzdžiui, funkcinių taškų metodas yra naudojamas kuriant sistemos dizainą, kodo eilučių skaičius kuriant sistemą, o panaudojimo atvejų taškų metodas – ankstyvosiose projekto stadijose – iš panaudojimo atvejų diagramų.

Panaudojimo atvejų taškų metodas – tai programinės įrangos vertinimo metodas, skirtas nustatyti sistemos ar programinės įrangos apimtį projekto metu. Metodas yra naudojamas pradinėse programinės įrangos kūrimo fazėse. Šis metodas yra matuojamas skaičiuojant aktorių ir transakcijų skaičių, įskaitant panaudojimo taškų modelius.

Panaudojimo atvejų modelis apibrėžia sistemos veiklos sritį. Panaudojimo atvejų taškų metodas yra skirtas išmatuoti programinės įrangos projekto sudėtingumą ir funkcionalumą.

UCP gali būti klasifikuojamas trimis grupėmis[2]:

- neklasifikuoti UCP;
- techninis sudėtingumo faktorius TCF;
- aplinkos sudėtingumo faktorius ECF;

Panaudojimo atvejų transakcijos padeda susidoroti su panaudojimo atvejų aprašymo ilgio variacijomis. Panaudojimo atvejų specifikacijos gali būti parašytos glaustai arba detalai. Jei dvi panaudojimo atvejų specifikacijos turi vienodą unikalių transakcijų skaičių, jos turi vienodą svorį.

Panaudojimo atvejų kūrėjas Ivaras Jacobsonas panaudojimo atvejų transakciją apibūdino kaip „kelionę į abi puses“, iš naudotojo į sistemą ir atgal į naudotoją. Transakcija yra užbaigiama, kai sistema sulaukia naujo įvesties stimulo. [6] Kitais žodžiais, vienoje transakcijoje aktorius atlieka tam tikrą veiksmą, kuris yra įvesti duomenis į sistemą, tuomet sistema sureaguoja, apdoroja įvesties

duomenis ir gražina rezultatą aktoriui. Nauja transakcija prasideda tada, kai aktorius sureaguoja į rezultatą, pateikia įvesties duomenis.

Pvz.:

Transakcija nr.1:

- naudotojas pasirenka X;
- sistema parodo, kad Y yra sujunta su X;

Transakcija nr.2:

- vartotojas pasirenka dar vieną Y ir patvirtina;
- sistema ieško rezultato ir jį pateikia;

1.4. Projekto apimties vertinimo, paremto UCP, žingsniai

1.4.1. Aktorių svorio skaičiavimas

Panaudojimo atvejų modelyje aktoriai yra skirstomi į tris kategorijas (1.1 lentelė):

- paprastas;
- vidutinis;
- sudėtingas.

1.1 lentelė UCP aktorių svoriai

Aktoriaus tipas	Aprašas	Faktorius
Paprastas	Programos vartotojo sąsaja	1
Vidutinis	Interaktyvi vartotojo sąsaja	2
Sudėtingas	Grafinė vartotojo sąsaja	3

Paprasti aktoriai tai kitos sistemos su apibrėžtomis API. Vidutis aktorius – tai taip pat kita sistema, kuri yra susieta su protokolu, pavyzdžiui TCP/IP, taip pat tai gali būti žmogus, sąveikaujantis su tekstine sąsaja, pavyzdžiui ASCII terminologija. Sudėtingas aktorius yra žmogus sąveikaujantis su vartotojo sąsaja.

Kiekvienas aktoriaus tipas yra suskaičiuojamas ir kiekvienas skaičius yra padauginamas iš svorio faktoriaus. Tada aktoriaus svorius yra suskaičiuojamas sudedant šias reikšmes.

1.4.2. Panaudojimo atvejų svorio skaičiavimas

Vertinant projekto apimtį reikia identifikuoti ir klasifikuoti visus panaudojimo atvejus, kurie priklauso programinei įrangai.

Panaudojimo atvejai, kaip ir aktoriai yra skirstomi į paprastus, vidutinius ir sudėtingus panaudojimo atvejus (1.2 lentelė). Šio etapo pagrindas yra panaudojimo atvejų transakcijų skaičius, įskaitant alternatyvųjį kelią. Paprastas panaudojimo atvejis turi 3 arba mažiau transakcijų, vidutinis nuo 4 iki 7 transakcijų, o sudėtingas daugiau nei 7 transakcijas. Kiekvienas panaudojimo atvejis yra suskaičiuojamas ir tada kiekvienas skaičius yra dauginamas iš svorio faktoriaus. Panaudojimo atvejų svoris yra apskaičiuojamas sudedant abi reikšmes.

1.2 lentelė Panaudojimo atvejų svorio faktoriai

Aktoriaus tipas	Aprašas	Faktorius
Paprastas	3 ar mažiau transakcijų	5
Vidutinis	4 – 7 transakcijos	10
Sudėtingas	Daugiau nei 7 transakcijos	15

1.4.3. UUCP (nekoreguotų panaudojimo atvejų taškų) skaičiavimas.

Tai yra apskaičiuojama sudedant galutinį aktorių ir panaudojimo atvejų svorį.

1.4.4. Techninių ir aplinkos faktorių skaičiavimas

Techniniai faktoriai nusako vartotojų lūkesčius programinei įrangai. Kitaip tariant, tai yra panašu į nefunkcinių reikalavimų vertinimą. Kiekvienas programinės įrangos projektas yra skirtingas, jis gali būti panašus, tačiau analogų pasitaiko labai retai, todėl visada reikia įvertinti skirtumus. Todėl šiame metode yra 13 techninių faktorių, kuriuos reikia apsvarstyti (1.3. lentelė).

1.3 lentelė Techniniai faktoriai

Faktorius	Pavadinimas	Aprašymas	Svoris
T1	Sistemos išskaidymas	Sprendimo architektūra turėtų būti centralizuota, priklausanti vienam klientui arba padalinta. Didesni skaičiai reprezentuoja labiau kompleksinę architektūrą.	2
T2	Sistemos našumas	Vartotojams labai svarbu yra sistemos atsakymo greitis. Pavyzdžiui, jei numatoma, kad serverio darbas bus labai lėtas, tai galima laikyti trivialiu faktoriumi. Didesni skaičiai reprezentuoja augančią atsakymo laiko svarbą.	1
T3	Galutinio vartotojo efektyvumas	Reikia įvertinti ar programinė įranga buvo sukurta vartotojo efektyvumui padidinti ar tik gebėjimams? Didesni skaičiai reprezentuoja, kurie projektai rimčiau žiūri į vartotojo efektyvumo didinimą.	1
T4	Sudėtinis vidinis algoritmų apdorojimas	Reikia įvertinti sudėtingų algoritmų ir testavimo kieki. Didesni skaičiai yra kompleksiški algoritmai, mažesni – paprastos duomenų bazės.	1
T5	Kodo universalumas	Ar bus pakartotinai panaudojamas programinis kodas? Kodo pakartotinis naudojimas sumažina apimtį, skirtą dislokuoti projektui. Tai sumažina laiką, skirtą klaidoms šalinti iš projekto. Pakartotinai gali būti panaudotos viešos bibliotekų funkcijos. Kuo didesnis pakartotinio panaudojimo lygis, tuo mažesnis skaičius.	1
T6	Paprastas diegimas	Reikia įvertinti, ar vartotojui yra svarbus parasti diegimo faktorius. Kuo didesnė vartotojo kompetencija, tuo mažesnis skaičius.	0.5
T7	Paprastas naudojimas (angl. <i>Usability</i>)	Reikia įvertinti, ar paprastas naudojimas yra pirminis faktorius vartotojui. Kuo didesnė svarba paprastam naudojimui, tuo aukštesnis skaičius.	0.5
T8	Perkeliamumas	Reikia įvertinti ar reikalingas sistemos palaikymas skirtingose platformose. Kuo daugiau platformų sistema turi palaikyti, tuo aukštesnis yra skaičius.	1
T9	Paprastas keitimas	Reikia įvertinti ar užsakovas svarsto galimybę keisti ar tobulinti sistemą ateityje. Kuo daugiau pakeitimų reikalaujama ateityje, tuo aukštesnis skaičius.	1
T10	Lygiagretumas	Kuo daugiau laiko skiriama duomenų ar programos problemoms spręsti, tuo aukštesnis skaičius.	1
T11	Specialių saugumo savybių įtraukimas	Reikia įvertinti ar galima panaudoti egzistuojančius saugumo sprendimus, ar jie bus kuriami pačių. Kuo daugiau saugumo sprendimų bus kuriama, tuo aukštesnis skaičius.	1
T12	Trečiųjų šalių tiesioginis priėjimas	Reikia įvertinti ar sistema reikalauja trečiųjų šalių kontrolės ar bibliotekų. Kuo daugiau trečiųjų šalių kodo, tuo mažesnis skaičius.	1
T13	Specialių vartotojo apmokymų priemonių reikalavimas.	Reikia įvertinti kiek reikės mokymų vartotojams. Kuo daugiau mokymų, tuo didesnis skaičius.	1

Kiekvieną iš techninių faktorių reikia įvertinti reikšme nuo 0 iki 5. Tuomet šios reikšmės yra sudauginamos iš kiekvieno aktoiaus svorio.

Techninis faktorius (TCF) yra apskaičiuojamas sudauginant kiekvieną faktorių pagal svorį ir juos susumuojant: $TFactor.TCF = 0.6 + (0.01 * TFactor)$.

Aplinkos faktoriai analizuoja komandos galimybes. Kai kurie analitikai aplinkos faktorius vadina patirties faktoriais. Vienas iš svarbiausių faktorių vykdant projektą ir yra komandos galimybės. Todėl yra svarbu įvertinti aštuonis aplinkos faktorius (1.4 lentelė).

1.4 lentelė Aplinkos faktoriai

Faktori-rius	Pavadinimas	Aprašymas	Svoris
F1	Komandos familiarumas su UML	Reikia įvertinti, kiek patirties turi komanda šioje srityje. Kuo didesnis patirties lygis, tuo aukštesnis skaičius.	1.5
F2	Komandos patirtis su aplikacija	Reikia įvertinti, kiek patirties turi komanda su aplikacija. Tai yra svarbu tik tada, kai yra daromi pokyčiai jau egzistuojančiai aplikacijai. Kuo didesnė patirtis, tuo didesnis skaičius. Naujai aplikacijai patirties lygis yra vertinamas 0.	0.5
F3	Objektinio programavimo patirtis	Reikia įvertinti programuotojų patirtį. Kuo daugiau patirties, tuo aukštesnis skaičius.	1
F4	Vadovaujančio analitiko gebėjimai	Reikia įvertinti analitiko patirtį, ypač reikalavimų specifikacijos srityje. Kuo didesnė patirtis, tuo aukštesnė reikšmė.	0.5
F5	Motyvacija	Reikia įvertinti, kokia yra komandos motyvacija. Kuo aukštesnė motyvacija, tuo aukštesnė reikšmė.	1
F6	Stabilūs reikalavimai	Reikia įvertinti reikalavimų pokyčių riziką ir kaip galima jų išvengti. Kuo daugiau pokyčių, tuo aukštesnė reikšmė.	2
F7	Nepilnos darbo dienos darbuotojai	Reikia įvertinti kiek yra darbuotojų, dirbančių nepilną darbo dieną, ir jų poveikį komandai. Tai yra neigiamą poveikį darantis faktorius.	-1
F8	Sudėtinga programavimo kalba	Kuo sunkesnė programavimo kalba, tuo aukštesnė reikšmė. Reikia vertinti ne programavimo kalbos sudėtingumą, bet turimos komandos gebėjimus, pavyzdžiui, PHP programuotojui Java turėtų būti sudėtinga programavimo kalba.	-1

Aplinkos faktoriai yra vertinami reikšmėmis nuo 0 iki 5.

Aplinkos faktorius (EF) yra apskaičiuojamas taip pat, kaip techninis. Jo formulė yra $EF = 1.4 + (-0.03 * EFactor)$.

1.4.5. Panaudojimo atvejų taškų skaičiavimas

Panaudojimo atvejų taškų skaičiavimo formulė yra $UCP = UUCP * TCF * EF$.

1.4.6. Apimties nustatymas

Tai yra paskutinis metodo skaičiavimo žingsnis, parodantis galutinį projekto apimties rezultatą. Šis žingsnis apskaičiuojamas sudauginant specifines UCP reikšmes (darbo valandas).

Mokslininkų siūlomos naudoti reikšmės:

- metodo išradėjas Karneris siūlo naudoti 20 valandų vertinimą vienam panaudojimo atvejo taškui;
- panaudojimo atvejų mokslininkų komanda, kuriai kuruoja Edas Karolas siūlo naudoti 28 valandas vienam panaudojimo atvejo taškui;

- mokslininkas Rojus Klemas, siūlo naudoti 15–30 valandų panaudojimo atvejo taškui.

Taip pat galima įvertinti projekto kompleksiskumą ir pagal tai nustatyti valandų skaičių. Kompleksiškas projektas – 28 valandos, paprastas – 20 valandų.

Norint nustatyti projekto kompleksiskumą reikia F1–F6 aplinkos faktoriams ir T7–T8 techniniams faktoriams, turintiems reikšmę didesnę negu 3, pridėti po vieną tašką. Tuomet reikia susumuoti taškus, jei gaunama mažiau negu 3 taškai, reikia taikyti 20 valandų per vieną panaudojimo atvejo tašką, jei 3–4 taikyti 28 valandas, o jei gaunama daugiau nei 5, restruktūrizuoti projektą, nes jis yra per daug kompleksiškas.

Gavus užsakymą iš užsakovo reikėtų nustatyti pirminius programinės įrangos vartotojus. Tuomet reikėtų apibrėžti vartotojų tikslus programinės įrangos naudojimui. Trečias žingsnis būtų panaudojimo atvejų apibrėžimas, tam galima naudoti tiek panaudojimo atvejų scenarijus, formalias panaudojimo atvejų dokumentacijas, neformalias panaudojimo atvejų dokumentacijas, ar UML diagramas.

Atlikus visus šiuos žingsnius, galima pradėti programinės įrangos vertinimą panaudojimo atvejų taškų metodu.

Panaudojimo atvejų taškų metodo plusai [8]:

1. panaudojimo atvejų taškų metodas gali būti automatizuotas. Kai kurie įrankiai, kurie dirba su panaudojimo atvejų diagramomis gali automatiškai suskaičiuoti panaudojimo atvejų taškus. Tai projekto komandai gali sutaupyti laiko projekto vertinimo etape;
2. naudojant panaudojimo atvejų taškų metodą organizacijoje yra įmanoma nustatyti vidutinį projekto atlikimo laiką. Tai gali būti labai naudinga planuojant būsimus projektų tvarkaraščius. Tačiau yra prielaida, kad ne visos panaudojimo atvejų diagramos yra parašytos laikantis to paties detalumo lygio;
3. panaudojimo atvejų taškai yra aiškus apimties matavimo vienetas. Geras vertinimas leidžia mums atskirti apimties vertinimą nuo gautos trukmės. Aplikacijos apimtis gali būti nepriklausoma nuo, įgūdžių, komandos patirties, kuri tai paveikia.

Panaudojimo atvejų taškų metodo minusai [8]:

1. vienas iš pagrindinių metodo minusų yra tas, kad skaičiavimai negali būti atlikti tol, kol nėra parašyti visi panaudojimo atvejai. Panaudojimo atvejų diagramų aprašymas dažniausiai užima 10–20% projekto laikotarpio;
2. taisyklės, kurios apibrėžia kas gali būti transakcija yra netikslios. Vartotojo žingsnių skaičiavimas scenarijuje yra tik apytikslis;
3. kai kurie techniniai faktoriai gali neturėti realaus poveikio projektui. Pavyzdžiui, techninis faktorius, kuris aprašo, kad sistema turi būti lengvai instaliuojama.

1.5. Panaudojimo atvejų taškų metodo vartotojai

Įmonėje panaudojimo atvejų diagramas dažniausiai naudoja projektų vadovai bei analitikai. Projekto apimties vertinimas jau dešimtmečius sukelia problemų tiek analitikams, tiek projektų vadovams, tiek programinės įrangos kūrėjams. Todėl panaudojimo atvejų taškų skaičiavimo vartotojai jie ir bus.

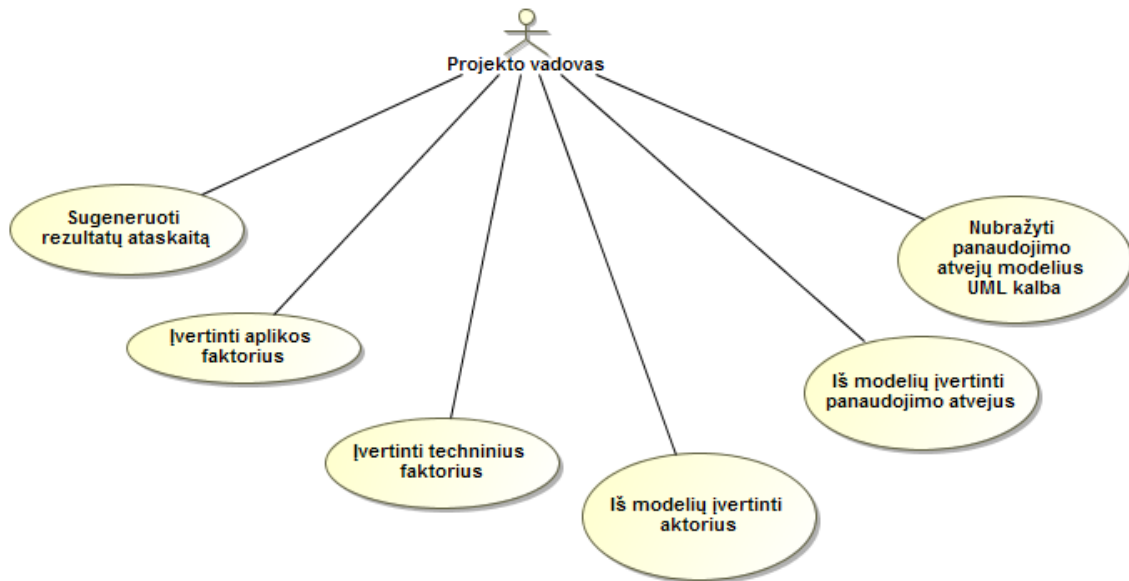
Projektų vadovo atsakomybė yra valdyti projektą, t.y. užtikrinti, kad viskas būtų atliekama laiku, pagal planą, kad būtų neviršijamas nustatytas biudžetas. Todėl svarbu iš anksto nustatyti projekto apimtį ir projekto kainą. Projekto apimtį nustatinėja ne tik projektų vadovas, bet ir analitikas.

Ankstyvose projekto kūrimo stadijose, projektų vadovai neretai susiduria su problema, kad jie negali tiksliai numatyti koks bus projektas, ypač jei tai yra nauja sritis. Užsakovai dažnai informaciją pateikia neišsamiai, todėl vien pateiktos informacijos iš užsakovo neužtenka.

Panaudojimo atvejų taškų metodo vartotojai yra kvalifikuoti srities specialistai, t.y. projektų vadovai ar analitikai. Prie projekto dažniausiai dirba vienas projekto vadovas, bet analitikų gali būti

ir daugiau. Kiekviename IT projekte yra rekomenduojama turėti ir projekto vadovą ir analitiką, todėl šiomis dienomis tai yra paklausios profesijos.

UCP metodo vartotojų panaudojimo atvejų diagrama susideda iš 6 panaudojimo atvejų (1.1 pav.).



1.1 pav. Panaudojimo atvejų diagrama, iliustruojanti projekto vadovą, vertinantį projekto apimtį

1.5 lentelė Vartotojų problemos vertinant projekto apimtį, naudojant UCP metodą.

Problema	Kaip vyksta
Projektų apimties skaičiavimai dažnai būna netikslūs.	Projektų vadovai naudoja įvairius skaičiavimo metodus, nėra vieningos nuomonės ir rekomendacijų ką reiktų naudoti.
Projekto apimties vertinimas, naudojant panaudojimo atvejų taškų metodą, atima daug laiko.	Daug informacijos reikia suvesti rankomis. Taip pat panaudojimo atvejų taškų metodas gali būti atliekamas tik turint visas panaudojimo atvejų diagramas.
Ankstyvuose projekto stadijose yra sunku apskaičiuoti projekto apimtį.	Daug metodų, sunku pasirinkti kokį naudoti.

1.6. Esamų panaudojimo atvejų taškų metodikų analizė

1.6.1. Mokslininkų sukurtas U–EST įrankis

Mokslininkai iš Osakos universiteto ir „Hitachi Systems & Services“ inžinieriai sukūrė panaudojimo atvejų matavimo įrankį [1]. Sprendimas yra vienas iš svarbiausių veiksmų skaičiuojant panaudojimo atvejų taškus, todėl mokslininkai sukūrė automatinį panaudojimo atvejų taškų matavimo įrankį. Taip pat, jie pasiūlė keletą taisyklių kaip klasifikuoti aktorių ir panaudojimo atvejų taškų svorius. Buvo pasirinkta naudoti modelius aprašytus XML kalba, nes dauguma UML diagramų gali būti eksportuojamos kaip XML.

Kiekvieno aktoriaus sudėtingumas aprašomas kaip vartotojo sąsaja tarp aktoriaus ir sistemos. Tačiau, aktoriaus apibrėžimas, panaudojimo atvejų modelyje, neduoda informacijos apie vartotojo sąsają. Būtent, dėl to, informacija gaunama tik apie aktoriaus vardą. Todėl šie mokslininkai pasiūlė tris žingsnius, kaip klasifikuoti aktorių sudėtingumą.

Aktorių sudėtingumo nustatymo taisyklės [1]:

1. aktoriaus vardu paremta klasifikacija. Paprastai aktorius yra žmogus arba išorinė sistema. Jei aktorius yra asmuo, tai sudėtingumas gali būti vidutinis arba sudėtingas, jei sistema – paprastas

arba vidutinis. Pirmiausia reikia išsiaiškinti ar aktorius yra žmogus ar sistema. Todėl, galima pasiruošti raktinių žodžių sąrašą. Pavyzdžiui sistemos aktoriumi gali būti naudojami raktažodžiai: „sistema“, „serveris“;

2. raktažodžiais paremta klasifikacija. Čia dėmesys yra skiriamas įvykių srautui, įskaitant panaudojimo atvejus, kur aktorius yra svarbus. Pirmiausia yra paruošiami trijų tipų raktažodžių sąrašai kiekvienam aktorių sudėtingumui. Pavyzdžiui, sudėtingo aktoriaus sąrašė galėtų būti tokie žodžiai kaip „mygtukas“, „vartotojo sąsaja“. Tada UML panaudojimo atvejų diagramose, nevertinant aktorių sudėtingumo, yra užrašomi aktoriui būdingi raktažodžiai. Galiausiai aktorių raktažodžiai ir aktorių svorio raktažodžiai yra sulyginami ir taip nustatomas sudėtingumas.

3. patirtimi paremta klasifikacija. Patirtis apima informaciją apie panaudojimo atvejų diagramą ir apie panaudojimo atvejį prieš tai buvusiuose projektuose. Jei egzistuoja keletas vardų, kurie yra tokie patys kaip pasirinkto aktoriaus, tada yra nusprendžiamas sudėtingumas, kuris buvo dažniausiai pasikartojantis.

Taip pat mokslininkai parašė ir panaudojimo atvejų sudėtingumo klasifikavimo taisykles. Panaudojimo atvejų sudėtingumas priklauso nuo transakcijų skaičiaus. Todėl reikia orientuotis į įvykių srautą panaudojimo atvejų diagramoje. Paprasčiausias būdas apskaičiuoti transakcijas yra suskaičiuoti kiekvieną įvykį. Nėra jokio standarto kaip aprašyti įvykių srautą, galima rašyti aprašymą naudojant natūralią kalbą.

Yra dešimt gairių kaip parašyti sėkmingą įvykių srauto scenarijų. Tačiau mokslininkai rašydami šiais taisyklėmis didžiausią dėmesį skyrė šioms dviem gairėm:

1. naudoti paprastą gramatiką. Sakinių struktūra turėtų būti visiškai paprasta. Kad būtų galima lengvai suprasti kas yra daiktavardis, veiksmažodis, papildinys ir prielinksnis;
2. įtraukti priimtinių veiksmų rinkinį. Pirminis aktorius siunčia duomenis sistemai, sistema patvirtina užklausą ir duomenis, sistema įspėja vidinę būseną ir atsako aktoriui su rezultatu.

Pagal šias gaires mokslininkai siūlo analizuoti įvykius morfologiškai ir sintaksiškai. Analizuojant tokiu būdu galima gauti informacija apie morfologijos vienetą ir priklausomybę tarp žodžių sakinyje.

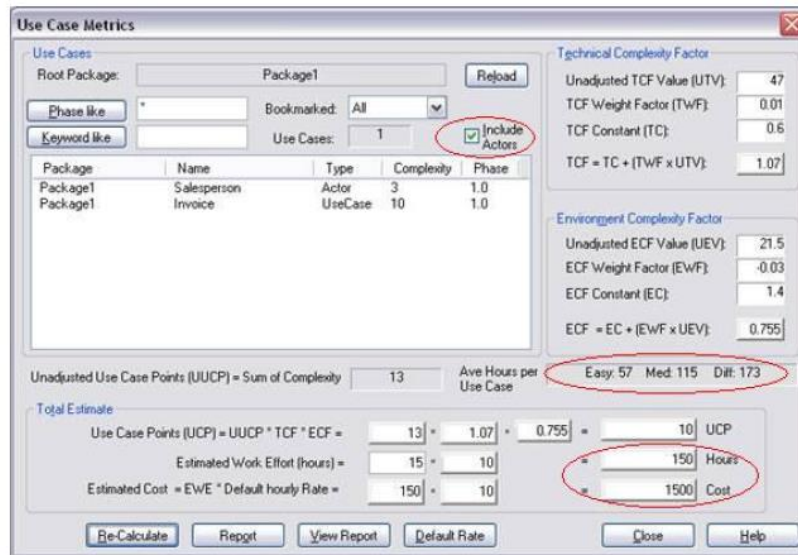
Mokslininkai panaudojo morfologinę analizę visiems įvykiams (sakiniam) ir gavo informaciją apie veiksnį ir tarinį. Todėl iš to parašė panaudojimo atvejų svorio gavimo taisykles:

1. atsižvelgti į kiekvieną tarinį ir veiksnį kaip transakcijos kandidatą;
2. tarp kandidatų identifikuoti vieną, kuris yra susietas su aktoriaus operacijai ir sistemos atsaku kaip aktorius.

Remdamiesi savo rekomendacijomis Osakos universiteto mokslininkai sukūrė prototipą, kurį pavadino U-EST (angl. *Use case based Estimation Supporting Tool*). Įrankis realizuotas *Java* bei *Xerces2 Java Parser* technologijomis. Jis yra skirtas analizuoti modelio failą. Įvestis – XML failas. Pirmiausia vartotojas sukuria panaudojimo atvejų modelius ir eksportuoja kaip XML failus. Tada XML automatinės analizės įrankis identifikuoja aktorius ir panaudojimo atvejus iš įvesties failo. Sudėtingumo analizės įrankis įvertina sudėtingumą ir apskaičiuoja UUCP. U-EST atvaizduoja aktorių ir panaudojimo atvejų sąrašą su jų sudėtingumu. Taip pat U-EST atvaizduoja įvykių sąrašą, veiksmų ir tarinių rinkinius panaudojimo atvejuose, kurie yra identifikuojami kaip transakcijos. Tada, nustatant techninius ir aplinkos faktorius, UCP skaičiuotuvai išveda rezultatą, kuris yra patalpinamas duomenų bazėje. Apimtis yra apskaičiuojama sudauginant specifines reikšmes iš UCP. Šis įrankis buvo panaudotas penkiems programinės įrangos projektams.

1.6.2. Sparx Enterprise Architect (EA) projekto apimties vertinimas

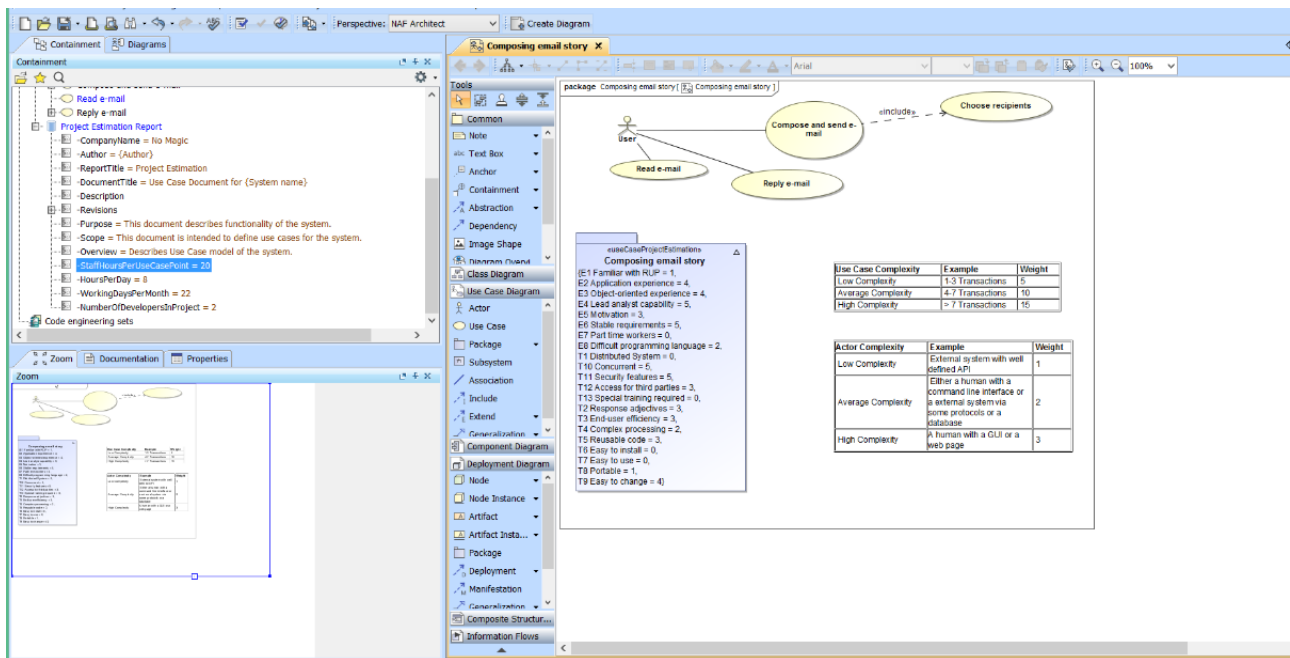
Pirmiausia reikia paruošti panaudojimo atvejų diagramas. Tuomet yra atliekami standartiniai metodo žingsniai: nustatomas aktorių ir panaudojimo atvejų sudėtingumas, įvertinami techniniai ir aplinkos faktoriai, produktyvumo faktorius. Galiausiai yra apskaičiuojamas UCP [12]. *Sparx Enterprise Architect* programos sąsaja pavaizduota paveikslėlyje 1.2.



1.2 pav. Sparx Enterprise Architect programos sąsaja

1.6.3. MagicDraw – projekto apimties vertinimo ataskaita

Dar vienas iš įrankių, tai analitikams ir projektų vadovams puikiai žinoma programinė įranga *MagicDraw* (1.3 pav.) [10].



1.3 pav. MagicDraw metodo taikymas

MagicDraw programinėje įrangoje yra galimybė vertinti projekto apimtį, naudojant tradicinį panaudojimo atvejų taškų metodą. Nubraižius visas panaudojimo atvejų diagramas atliekami tokie veiksmai (tai yra tradiciniai panaudojimo atvejų taškų skaičiavimo žingsniai):

1. aktorių klasifikacija;
2. panaudojimo atvejų klasifikacija;
3. projekto charakteristikų vertinimas: techniniai ir aplinkos faktoriai;
4. panaudojimo atvejų taškų skaičiavimas.

Įvertinus visus reikalingus kriterijus, galima sugeneruoti projekto vertinimo ataskaitą, kurioje yra atvaizduoti iš panaudojimo atvejų diagramų nustatyti panaudojimo atvejai (1.8 – 1.9 pav.) ir aktoriai (1.4 pav., 1.5 pav.).

Aktoriai gali būti paprasti, vidutiniai ir sudėtingi. Ataskaitoje yra išvedamas projekto pavadinimas, aktorius pavadinimas ir sudėtingumas (1.6 pav.).

Package	Name	Complexity
Composing email story	User	High Complexity

1.4 pav. Aktorius *MagicDraw* sugeneruotoje ataskaitoje

Ataskaitoje yra atvaizduojama kiek ir kokio sudėtingumo aktorių buvo projekte, taip pat išvedamas skaičiavimų rezultatas (1.7 pav.).

Actor Complexity	Description	Weight	Number of elements	Result
Low Complexity	External system with well - defined API	1	0	0
Average Complexity	Either a human with a command line interface or a external system via some protocols or a database	2	0	0
High Complexity	A human with a GUI or a web page	3	1	3
Unadjusted Actor Weights(UAW)				3

1.5 pav. Aktorių svoriai sugeneruotoje *MagicDraw* ataskaitoje

Panaudojimo atvejai gali būti paprasti, vidutiniai ir sudėtingi. Ataskaitoje yra išvedama kokie panaudojimo atvejai buvo projekte, panaudojimo atvejų pavadinimas ir projekto pavadinimas (1.6 pav.).

Package	Name	Complexity
Composing email story		Average Complexity
Composing email story	Choose recipients	Low Complexity
Composing email story	Compose and send e-mail	Average Complexity
Composing email story	Reply e-mail	Average Complexity

1.6 pav. Panaudojimo atvejai sugeneruotoje *MagicDraw* ataskaitoje

Ataskaitoje yra atvaizduojami panaudojimo atvejų svorio rezultatai (1.7 pav.).

Use Case Category	Description	Weight	Number of elements	Result
Low Complexity	1-3 transactions	5	1	5
Average Complexity	4-7 transactions	10	3	30
High Complexity	> 7 transactions	15	0	0
Unadjusted Use Case Weights (UUCW)				35

1.7 pav. Panaudojimo atvejų klasifikavimas sugeneruotoje *MagicDraw* ataskaitoje

Šie kriterijai yra automatiškai nuskaitomi iš UML panaudojimo atvejų modelių, prie aktorių ir panaudojimo galima nustatyti kokio jie bus sudėtingumo. Tuomet ataskaitoje yra pateikiami nurodytų techninių ir aplinkos faktorių apskaičiuoti svoriai (1.8 pav., 1.9 pav.).

Ataskaitoje yra išvedamas techninių ir aplinkos faktorių sąrašas, su jom priskirtom reikšmėm ir gautu galutiniu rezultatu (1.8 pav., 1.9 pav.).

Factor	Description	Weight	Assigned Value (0-5)	Weighted Value
T1	Distributed System	2.0	0.0	0.0
T2	Response adjectives	1.0	3.0	3.0
T3	End-user efficiency	1.0	3.0	3.0
T4	Complex processing	1.0	2.0	2.0
T5	Reusable code	1.0	3.0	3.0
T6	Easy to install	0.5	0.0	0.0
T7	Easy to use	0.5	0.0	0.0
T8	Portable	2.0	1.0	2.0
T9	Easy to change	1.0	4.0	4.0
T10	Concurrent	1.0	5.0	5.0
T11	Security features	1.0	5.0	5.0
T12	Access for third parties	1.0	3.0	3.0
T13	Special training required	1.0	0.0	0.0
Technical Factor Value (TFactor)				30.0

1.8 pav. Techninių faktorių svoriai

Factor	Description	Weight	Assigned Value (0-5)	Weighted Value
E1	Familiar with RUP	1.5	1.0	1.5
E2	Application experience	0.5	4.0	2.0
E3	Object-oriented experience	1.0	4.0	4.0
E4	Lead analyst capability	0.5	5.0	2.5
E5	Motivation	1.0	3.0	3.0
E6	Stable requirements	2.0	5.0	10.0
E7	Part time workers	-1.0	0.0	-0.0
E8	Difficult programming language	-1.0	2.0	-2.0
Environmental Factor Value (EFactor)				21.0

1.9 pav. Aplinkos faktorių svoriai

Projekto apimties vertinimo rezultatai yra išvedami išsamiai, atvaizduojant skaičiavimams reikalingas formules. Galutiniai rezultatai yra pateikiami valandomis, mėnesiais ir dienomis (1.12 pav.).

1. **(EF)** = $1.4 + (-0.03 * \text{EFactor}) = 0.7699999999999999$
2. **(UCP)** = $\text{UUCP} * \text{TCF} * \text{EF} = 38 * 0.8999999999999999 * 0.7699999999999999 = 26.333999999999993$
3. **Estimated Effort in Person Hours** = $\text{UCP} * \text{PHM} = 26.333999999999993 * 20.0 = 526.6799999999998$ hours
4. **Estimated Effort in schedule time** = $526.6799999999998 / 2.0 / 8.0 = 32.91749999999999$ days
5. **Estimated Effort in working days** = $32.91749999999999 / 22.0 = 1.4962499999999996$ months

1.10 pav. Projekto apimties vertinimas

Taigi, šis pavyzdinis projektas reikalauja 1.5 mėnesio, dirbant dviems programuotojams.

1.6.4. Panaudojimo atvejų taškų skaičiuotuvai

Skaičiuotuvai yra nemokami, pasiekiami internetinėje naršyklėje, nereikia pirkti licencijų (1.11 pav.) [9].

Šis skaičiuotuvai automatinis yra tik skaičiavimo procesas, visus kitus aspektus, kaip aktorius, panaudojimo atvejus, techninius ir aplinkos faktorius reikia susivesti rankomis, žiūrint į savo panaudojimo atvejų diagramą. Skaičiuotuve galima nustatyti minimalias ir maksimalias valandas, vienam panaudojimo atvejo taškui. Skaičiuotuvai veikia pagal pagrindinį panaudojimo atvejų taškų skaičiavimo metodą. Skaičiuotuvai išveda maksimaliais ir minimaliais numatomas projekto valandas.

Use Cases
 Simple(less than 4 transactions) 2 Average(4 to 7 transactions) 2 Complex(more than 7 transactions) 2

Actors
 Simple(System) 3 Average(System/Human) 3 Complex(Human) 3

Technical factors
 Distributed system: 0, Performance objectives: 0, End User efficiency: 0, Complex Processing: 0
 Reusable Code: 0, Ease of Installation: 0, Ease of Use: 0, Portable: 0
 Ease of Change: 0, Concurrent Use: 0, Special Security: 0, Access for Third Parties: 0
 Training Needs: 0

Environmental factors
 Familiar with Development Process: 0, Application Experience: 0, Object-Oriented Experience: 0
 Lead Analyst Capability: 0, Motivation: 0, Stable Requirements: 0
 Part-time Staff: 0, Difficult Programming Language: 0

Points
 UUCW - Unadjusted Use Case Weight: 60
 UAW - Unadjusted Actor Weight: 18
 UUCP - Unadjusted Use Case Points: 78
 TCF - Technical Complexity Factor: 0.6
 EF - Environmental Factor: 1.4
 UCP - Use Case Points: 66.52

Hours
 Minimum Hours per Use Case Point: 20
 Maximum Hours per Use Case Point: 25
 Average Hours per Use Case Point: 20
 Minimum hours: 1310.3999999999999
 Maximum hours: 1638

1.11 pav. Panaudojimo atvejų taškų skaičiuotuvus interneto naršyklėje

Interneto naršyklėje esanti skaičiuotuvą yra patogu vartoti, nes galima prisijungti iš bet kurio kompiuterio, turint tik interneto prieigą.

1.6.5. Excel skaičiuotuvus (verslo analitiko – Jono Hanseno)

Excel skaičiuotuvus iš esmės veikia taip pat, kaip skaičiuotuvus internetinėje naršyklėje. Informaciją reikia suvesti rankomis [11].

Pirmiausia suvedame panaudojimo atvejus ir aktorius, skaičiuotuve yra stulpeliai: kiekis, svoris ir rezultatas (1.12 pav.).

1	Use Case Category	Description	Count	Weight	Score
2	Simple	(1-3 transactions)	6	5	30
3	Average	(4-7 transactions)	2	10	20
4	Complex	(8+ transactions)	5	15	75
5	Sum (UUCW)		13		125
6					
7	Actor Category	Description	Count	Weight	Score
8	Simple	System through API	0	1	0
9	Average	System through communication protocol	0	2	0
10	Complex	Human actor through GUI	2	3	6
11	Sum (UAW)		2		6

1.12 pav. Aktorių ir panaudojimo atvejų vertinimas

Tuomet reikia suvesti techninius ir aplinkos faktorius (1.13 pav.).

Technical Factor	Weight	Relevance 0-5	Score
T1 Distributed System	2	1	2
T2 Performance	1	1	1
T3 End User Efficiency	1	1	1
T4 Complex Internal Processing	1	1	1
T5 Reusability	1	2	2
T6 Easy to Install	0.5	1	0.5
T7 Easy to Use	0.5	1	0.5
T8 Portability	2	3	6
T9 Easy to Change	1	2	2
T10 Concurrency	1	1	1
T11 Special Security Features	1	1	1
T12 Provides Direct Access for Third Parties	1	1	1
T13 Special User Training Facilities Are Required	1	1	1
Sum			20
Complexity Factor			0.6
C2			0.01
TCF			0.8
Environmental Factor	Weight	Impacte 0-5	Score
E1 Familiarity With UML	1.5	5	7.5
E2 Part-Time Workers	-1	0	0
E3 Analyst Capability	0.5	1	0.5
E4 Application Experience	0.5	5	2.5
E5 Object-Oriented Experience	1	0	0
E6 Motivation	1	0	0
E7 Difficult Programming Language	-1	0	0
E8 Stable Requirements	2	0	0
Sum (ETF)			10.5
ECF = 1.4 + (-0.03 * Environmental Total Factor)			1.085
PF (Historical based average man hours per use case point)			12

1.13 pav. Techninių ir aplinkos faktorių vertinimas

Susivedus visas reikalingas reikšmes yra atliekami skaičiavimai ir išvedami galutiniai rezultatai (1.14 pav.).

UCP = UUCP * TCF * ECF * PF	1364,496	
UCP = UUCP * TCF * ECF	113,708	
<i>Rough (gu)estimate</i>	1364,496	

1.14 pav. Skaičiavimai ir rezultatai

Autorius teigia, kad į skaičiavimus yra įtraukti dizaino darbai, programavimas ir vienetų (angl. *unit*) testavimas, tačiau neįtrauktas rankinis, funkcinis testavimas ir projekto valdymas.

Excel skaičiuotuvą galima lengvai patobulinti, pritaikant bet kurį UCP metodą.

Atlikus esamų įrankių analizę, galima teigti, kad ne visi įrankiai informaciją gali identifikuoti iš UML panaudojimo atvejų diagramų. Išsamesnis metodų palyginimas 1.5 lentelėje.

1.5 lentelė Palyginimo lentelė (Įrankiai)

Įrankio/programinės įrangos pavadinimas	Dalis informacijos automatiškai imama iš UML diagramų	Nemokamas	Automatizuotas skaičiavimas	Papildomi parametrai
MagicDraw	Taip „(MagicDraw“ aplinkoje)	Ne	Taip	<ul style="list-style-type: none"> • Valandų skaičius per dieną. • Darbo dienos per mėnesį. • Programuotojų skaičius projekte.
Panaudojimo atvejų taškų skaičiuotuvai	Ne	Taip	Taip	<ul style="list-style-type: none"> • Maksimalus valandų skaičius panaudojimo atvejo taškui. • Minimalus valandų skaičius panaudojimo atvejų taškui. • Vidutinis valandų skaičius panaudojimo atvejų skaičiui.
Excel skaičiuotuvai (Jonas Hansenas)	Ne	Taip	Taip	PF – produktyvumo faktorius.
U-EST įrankis	Taip (iš XML)	Ne	Taip	Nėra
Sparx Enterprise Architect	Ne	Ne	Taip	PF – produktyvumo faktorius.

1.6.6. Kitų mokslininkų tyrimų apžvalga

Vieningų rekomendacijų, kaip naudoti panaudojimo atvejų taškų metodą nėra, tačiau dauguma mokslininkų tai įvardina kaip potencialų metodą programinės įrangos apimčiai nustatyti.

Mokslininkas Schneideris 2001 metais savo moksliniame darbe paminėjo, kad UCP vertinimo technika vyksta geriau, nei kitos egzistuojančios programinės įrangos apimties skaičiavimo technikos [5].

M. Damodaranas ir A.Washingtonas 2002 metais savo tyrime apžvelgė sėkmingą UCP metodo naudojimą, programinės įrangos kūrimo projektuose. Mokslininkai pabrėžė, kad nėra teisingo standarto, kaip reikėtų naudotis UCP metodu, transakcijų skaičiavimas vis dar nėra aiškus. Tačiau, mokslininkai nurodo, kad UCP metodas turi potencialą būti patikimas programinės įrangos apimties vertinimui, ištaisius tam tikrus neaiškumus.

R.K. Klemonsas 2006 metais atliko mokslinį tyrimą, kuriame tyrė produktyvumo faktorius (PF) internetiniams informacinių sistemų kūrimo projektams, naudojant panaudojimo atvejų taškų metodą. Produktyvumo faktorius buvo naudojamas nustatyti kiek darbo valandų reikia užbaigti projektui. Tyrimas nustatė, kad UCP metodu galima iš anksto apskaičiuoti 20% projekto apimties.

Norint gauti precizišką vertinimą reikia įvertinti techninius ir aplinkos faktorius. Taip pat, tyrimo autorius paskatino kitus tyrėjus naudoti UCP metodą.

R. Uzalas 2007 metais savo tiriamajame darbe padarė išvadas, kad UCP metodas yra turbūt geriausias egzistuojantis įrankis projekto apimčiai apskaičiuoti ir pabrėžė, kad generalizuotas UCP metodo standartizavimas bus labai svarbus pasiekimas programinės įrangos industrijoje.

K. Periyasamio ir A. Ghodo patobulintas UCP metodas – e-UCP :

Mokslininkai savo darbe daugiausiai dėmesio skyrė vidinėmis panaudojimo atvejų savybėmis. Tyrėjai pasiūlė modifikuota projekto apimties skaičiavimo techniką, kuri vadinasi išplėstinis panaudojimo atvejų taškų modelis (e-UCP). Modelyje yra šeši parametrai: įvesties parametras, išvesties parametras, prognozės parametras, sąlygos prognozavimas, sėkmingo scenarijaus veiksmas ir išimtis. Aktorių svorių klasifikacija buvo pakeista iš trijų UCP modelyje į septynias pasiūlytame modelyje.

Aktoriai klasifikuojami į:

- labai paprastus (0.5);
- paprastus (1);
- mažiau nei vidutinius (1.5);
- vidutinius (2);
- sudėtingus (2.5);
- labai sudėtingus (3);
- sudėtingiausius (3.5).

Taip pat siūlomas modelio panaudos atvejų svorių specifikacija yra ne 3, o 4 punktų:

- paprastas (0.5);
- vidutinis (1);
- sudėtingas (2);
- sudėtingiausias (3).

Siūlomas modelis naudoja 20 valandų produktyvumo faktorių, kaip ir standartinis.

Mokslininkų Mudasiro Manzooro Kirmani, Syedo Mohsino Saifo ir Abdulo Wahido Re-UCP metodas (patikslintas UCP metodas) [13].

Re-UCP tai yra išplėstas tradicinio UCP ir išplėstinio UCP metodas. Tai yra visapusiškiausias metodas, kuris turi prisitaikantį elgesį įvairaus lygio projektams. Metodo autoriai pasiūlė kitaip klasifikuoti aktorius, panaudojimo atvejus, techninius ir aplinkos faktorius.

Aktoriai klasifikuojami į:

- paprastus (1);
- vidutinius (2);
- sudėtingus (3);
- kritinius (4);

Kritinis aktorius apibrėžia sistemą, kuri sąveikauja su modeliais veikiančiais realiu laiku

Panaudojimo atvejai klasifikuojami į:

- paprastus (5);
- vidutinius (10);
- sudėtingus (15);
- kritinius (20);

Paprastas – iki 4 transakcijų, vidutinis – nuo 5 iki 8, sudėtingas – nuo 9 iki 15. Kritinis panaudojimo atvejų svoris gali būti tada, kai transakcijų kiekis didesnis už 15.

Taip pat metodo kūrėjai pridėjo dar vieną techninį faktorių – T14 – mastelis. Kuris gali būti apibrėžiamas kaip sistemos gebėjimas tvarkytis su padidėjusiais darbo krūviais, nepridedant papildomų išteklių. Tai yra efektyvi strategija plėsti sistemos pajėgumus.

Dar vienas patobulinimas – tai aplinkos faktorius E9 – projekto metodologija. Tai apibūdina programuotojų patirtį pasirinktai projekto metodologijai.

1.6 lentelė Mokslininkų tyrimo rezultatų lentelė

Mokslininkas (–ai)	Tyrimo rezultatai
Mokslininkai iš Osakos universiteto ir „Hitachi Systems & Services“ inžinieriai (U–EST įrankis)	<ol style="list-style-type: none"> 1. aktorių svorio skaičiavimo taisyklės; 2. panaudojimo atvejų svorio skaičiavimo taisyklės; 3. gairės įvykių srauto scenarijui.
R.K Clemmonsas	Tyrimas nustatė, kad UCP metodu galima iš anksto apskaičiuoti 20% projekto apimties. Norint gauti precizišką vertinimą reikia įvertinti techninius ir aplinkos faktorius.
R. Uzasas	Tyrimas nustatė, kad UCP metodas yra turbūt geriausias egzistuojantis įrankis projekto apimčiai apskaičiuoti ir pabrėžė, kad generalizuotas UCP metodo standartizavimas bus labai svarbus pasiekimas programinės įrangos industrijoje.
K. Periyasamio ir A. Godo patobulintas UCP metodas – e–UCP	<p>Mokslininkai rekomendavo aktorius ir panaudos atvejus klasifikuoti į daugiau klasifikacijų.</p> <p>Aktoriai:</p> <ul style="list-style-type: none"> • labai paprastus (0.5); • paprastus (1); • mažiau nei vidutinius (1.5); • vidutinius (2); • sudėtingus (2.5); • labai sudėtingus (3); • sudėtingiausias (3.5). <p>Panaudojimo atvejai:</p> <ul style="list-style-type: none"> • paprastas (0.5); • vidutinis (1); • sudėtingas (2); • sudėtingiausias (3).
Mudasiro Manzooro Kirmani, Syedo Mohsino Saifo ir Abdulo Wahido Re–UCP metodas	<p>Aktoriai klasifikuojami į:</p> <ul style="list-style-type: none"> • paprastus (1); • vidutinius (2); • sudėtingus (3); • kritinius (4); <p>Panaudojimo atvejai klasifikuojami į:</p> <ul style="list-style-type: none"> • paprastus (5); • vidutinius (10); • sudėtingus (15); • kritinius (20); <p>Pridėtas techninis faktorius T14 – mastelis ir aplinkos faktorius E9 – Projekto metodologija.</p>

1.7. Siekiamo sprendimo apibrėžimas

Šiuo moksliniu darbu siekiama surinkti rekomendacijas kaip reikėtų naudoti panaudojimo atvejų taškų metodikas, norint apskaičiuoti projekto apimtį bei pritaikyti *MagicDraw* programinę įrangą automatiniam skaičiavimui, remiantis UML kalba aprašytomis diagramomis.

Yra įvairiausių įrankių bei metodų, kuriais galima vertinti projekto apimtį, tačiau nėra vieningų rekomendacijų kokius metodus naudoti, kada juos naudoti, ir kaip juos naudoti. Įrankiuose, sukurtuose, pagal vieną iš esamų panaudojimo atvejų taškų metodo variacijų, dažnai trūksta automatizavimo, visą informaciją reikia susivesti rankomis, todėl šiame darbe bus pateiktos skirtingų metodikų naudojimo rekomendacijos, paprasto UCP metodo (UCP), išplėsto UCP metodo (e-UCP) ir patikslinto UCP metodo (Re-UCP) automatizavimas, rezultatų ataskaitų generavimas, *MagicDraw* programinėje įrangoje, kad būtų galima sutaupyti laiko projekto apimties vertinimo stadijoje.

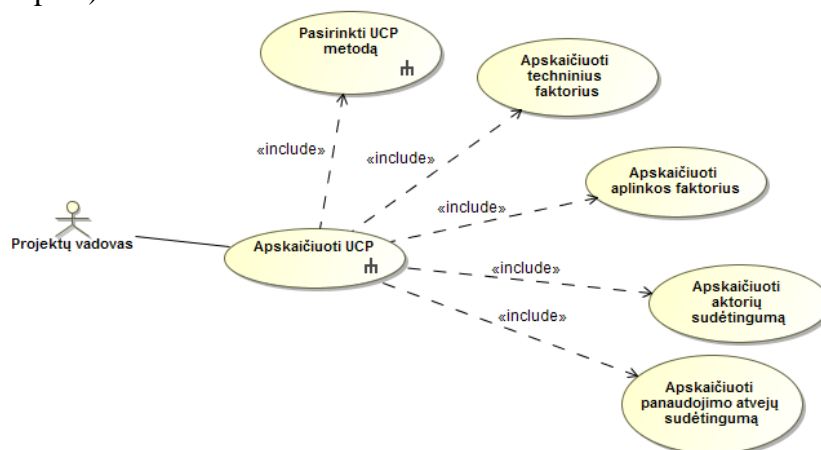
1.8. Analizės išvados

1. Nėra teisingo standarto kaip reikėtų naudoti panaudojimo atvejų taškų skaičiavimo metodą;
2. panaudojimo atvejų, aktorių, techninių ir aplinkos faktorių svorio vertinimas vis dar nėra aiškus;
3. UCP apimties vertinimo technika vyksta geriau nei kitos egzistuojančios programinės įrangos apimties skaičiavimo technikos (FP, LOC), vienas iš pagrindinių pranašumų yra tas, kad skaičiavimo metodas gali būti atliekamas pradinėje projekto kūrimo stadijoje;
4. išanalizavus mokslininkų tyrimu rezultatus, galima teigti, kad metodą reikia tobulinti sklandesniam naudojimui.

2. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO SPRENDIMO REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS

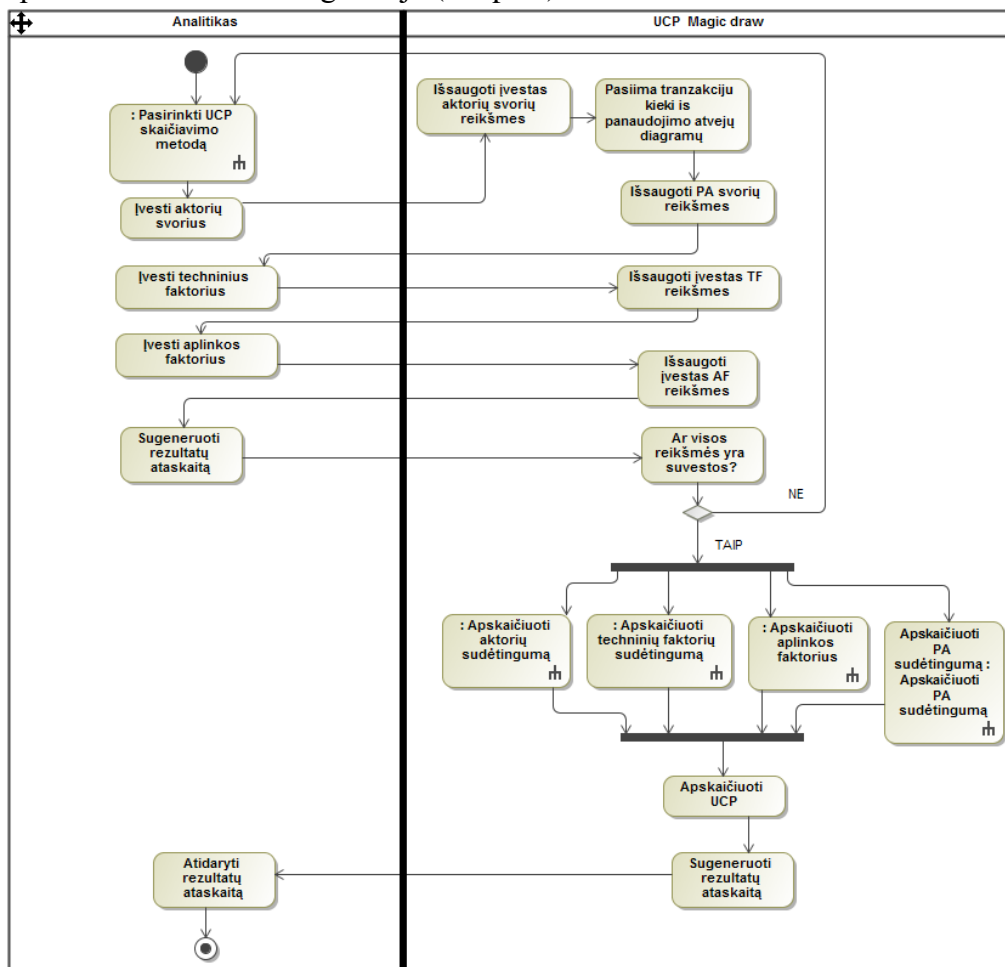
2.1. Reikalavimų specifikacija

Panaudojimo atvejų taškų metodas yra aktualus projektų vadovams bei analitikams. Panaudojimo atvejų taškų metodo įrankio pagrindinės funkcijos susideda iš šešių panaudojimo atvejų bei vieno aktoiaus (2.1 pav.).



2.1 pav. UCP skaičiavimo metodo atvejų diagrama

Projekto vadovo tikslas yra apskaičiuoti UCP. Detaliau panaudojimo atvejis „Apskaičiuoti UCP“ yra pateikiamas veiklos diagramoje (2.2 pav.).



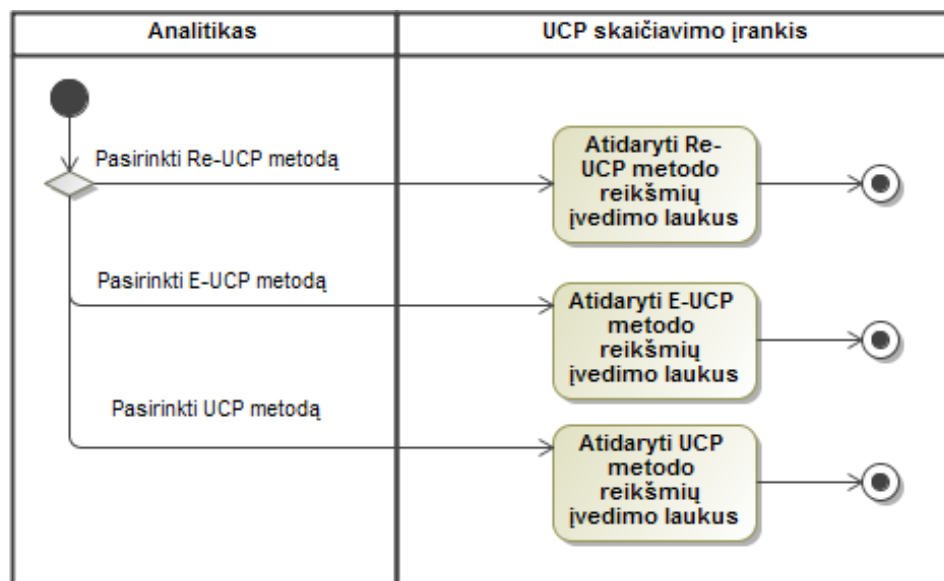
2.2 pav. „Apskaičiuoti UCP“ veiklos diagrama

2.1 lentelė. PA „Apskaičiuoti UCP“ formalus aprašymas

PA „Apskaičiuoti UCP“		
Prieš sąlyga		Vartotojas turi <i>MagicDraw</i> programinę įrangą. Vartotojas turi sistemos panaudojimo atvejų diagramą
Sužadinimo sąlyga		Vartotojas nori apskaičiuoti UCP.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Pasirinkti UCP metodą; Apskaičiuoti PA sudėtingumą; Apskaičiuoti aktorių sudėtingumą; Apskaičiuoti techninius faktorius; Apskaičiuoti aplinkos faktorius;
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka UCP skaičiavimo metodą.		Sistema pateikia metodo įvedimo laukus.
2. Vartotojas įvertina aktorių svorį.		Sistema išsaugo įvestus duomenis.
3. PA sudėtingumas nuskaitytas iš panaudojimo atvejų diagramos.		Sistema išsaugo įvestus duomenis.
4. Vartotojas įveda techninių faktorių reikšmes.		Sistema išsaugo įvestus duomenis.
5. Vartotojas įveda aplinkos faktorių reikšmes.		Sistema išsaugo įvestus duomenis.
6. Vartotojas generuoja rezultatų ataskaitą.		Sistema apskaičiuoja aktorių sudėtingumą, techninių faktorių sudėtingumą, aplinkos faktorius bei PA sudėtingumą; Sistema apskaičiuoja UCP; Sistema sugeneruoja rezultatų ataskaitą
7. Vartotojas peržiūri rezultatų ataskaitą.		Sistema sugeneruoja rezultatų ataskaitą
Po sąlyga:		Sugeneruota rezultatų ataskaita, apskaičiuotas UCP
Alternatyvūs scenarijai		

Panaudojimo atvejis „Apskaičiuoti UCP“ susideda iš penkių apimančių panaudojimo atvejų. Iš esmės skaičiavimai gali būti atliekami bet kokia tvarka, tačiau rekomenduojama atlikti tokiu eiliškumą kaip „Apskaičiuoti UCP“ veiklos diagramoje (pav. 2.2.). Toliau bus pateikiamos visų panaudojimo atvejų veiklos diagramos bei panaudojimo atvejų specifikacijos.

Pirmiausia UCP apskaičiavimas turėtų prasidėti nuo metodo pasirinkimo, pagal tai atitinkamai bus apskaičiuojamos visos reikšmės, būtinos apskaičiuoti UCP (pav. 2.3.).

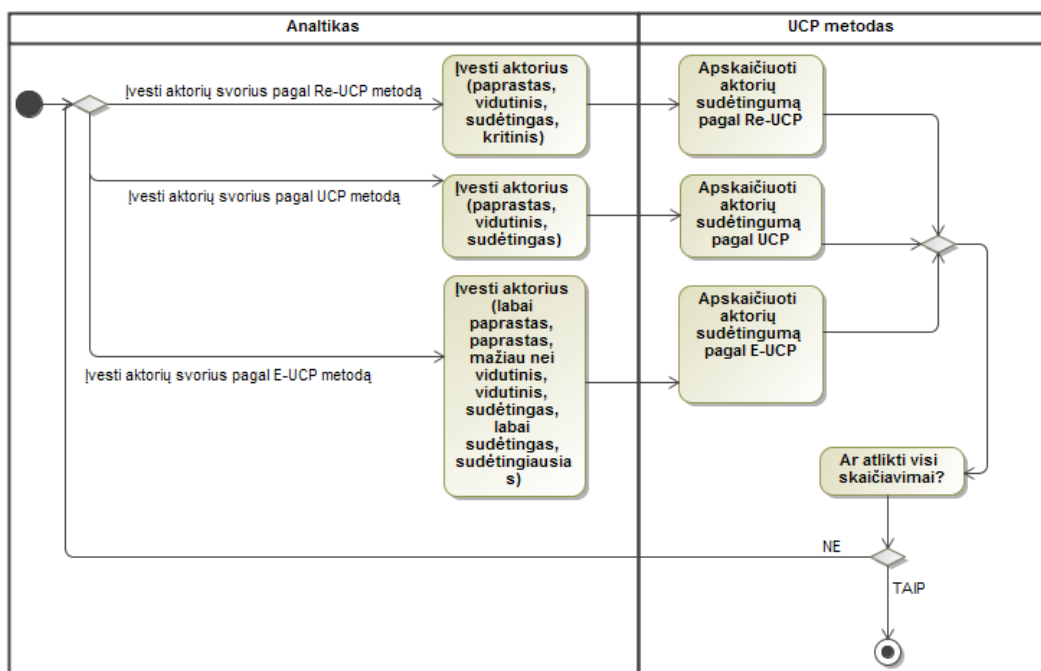


2.3 pav. „Pasirinkti UCP metodą“ veiklos diagrama

2.2 lentelė PA „Pasirinkti UCP metodą“ formalus aprašymas

PA „Pasirinkti UCP metodą“		
Prieš sąlyga		Vartotojas įsijungęs UCP skaičiavimo įrankį. Vartotojas turi sistemos panaudojimo atvejų diagramą.
Sužadinimo sąlyga		Vartotojas nori apskaičiuoti UCP.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	Apskaičiuoti UCP.
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka UCP skaičiavimo metodą.		Sistema pateikia metodo įvedimo laukus.
Po sąlyga:		Reikšmių įvedimo laukai, atitinkantys pasirinktą metodą.
Alternatyvūs scenarijai		

Pagal metodo variaciją gali skirtis aktorių svorių klasifikacija. Pavyzdžiui tradicinis UCP skaičiavimo metodas aktorius klasifikuoja į tris kategorijas, o išplėstas net į 7 kategorijas (pav.2.4.).

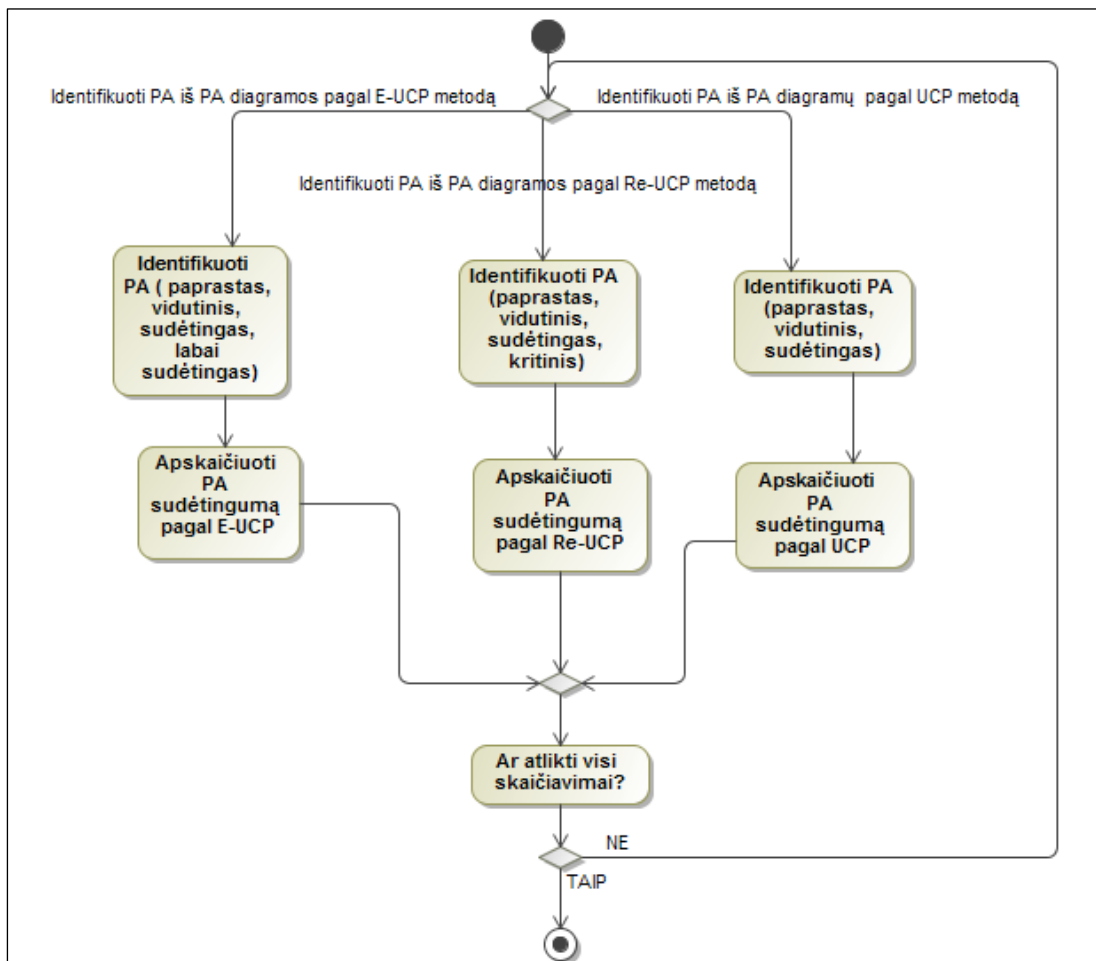


2.4 pav. Veiklos diagrama „Apskaičiuoti aktorių sudėtingumą“

2.3 lentelė PA „Pasirinkti UCP metodą“ formalus aprašymas

PA „Apskaičiuoti aktorių sudėtingumą“		
Prieš sąlyga		Vartotojas įsijungęs UCP skaičiavimo įrankį. Vartotojas turi sistemos panaudojimo atvejų diagramą.
Sužadinimo sąlyga		Vartotojas nori įvertinti aktorius.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	Apskaičiuoti UCP.
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda aktorių svorius pagal UCP metodą.		Sistema apskaičiuoja aktorių sudėtingumą pagal UCP metodą.
2. Vartotojas įveda aktorių svorius pagal E-UCP metodą.		Sistema apskaičiuoja aktorių sudėtingumą pagal E-UCP metodą.
3. Vartotojas įveda aktorių svorius pagal Re-UCP metodą.		Sistema apskaičiuoja aktorių sudėtingumą pagal Re-UCP metodą.
4.		Sistema patikrina ar visi skaičiavimai atlikti.
Po sąlyga:		Sistemoje išsaugoti skaičiavimų duomenys.
Alternatyvūs scenarijai		
A1. Jei įvestos ne visos reikšmės.		Sistema siūlo įvesti trūkstamas reikšmes.

Panaudojimo atvejų sudėtingumas yra įvertinamas iš panaudojimo atvejų diagramos, pagal transakcijų skaičių (2.5 pav.).

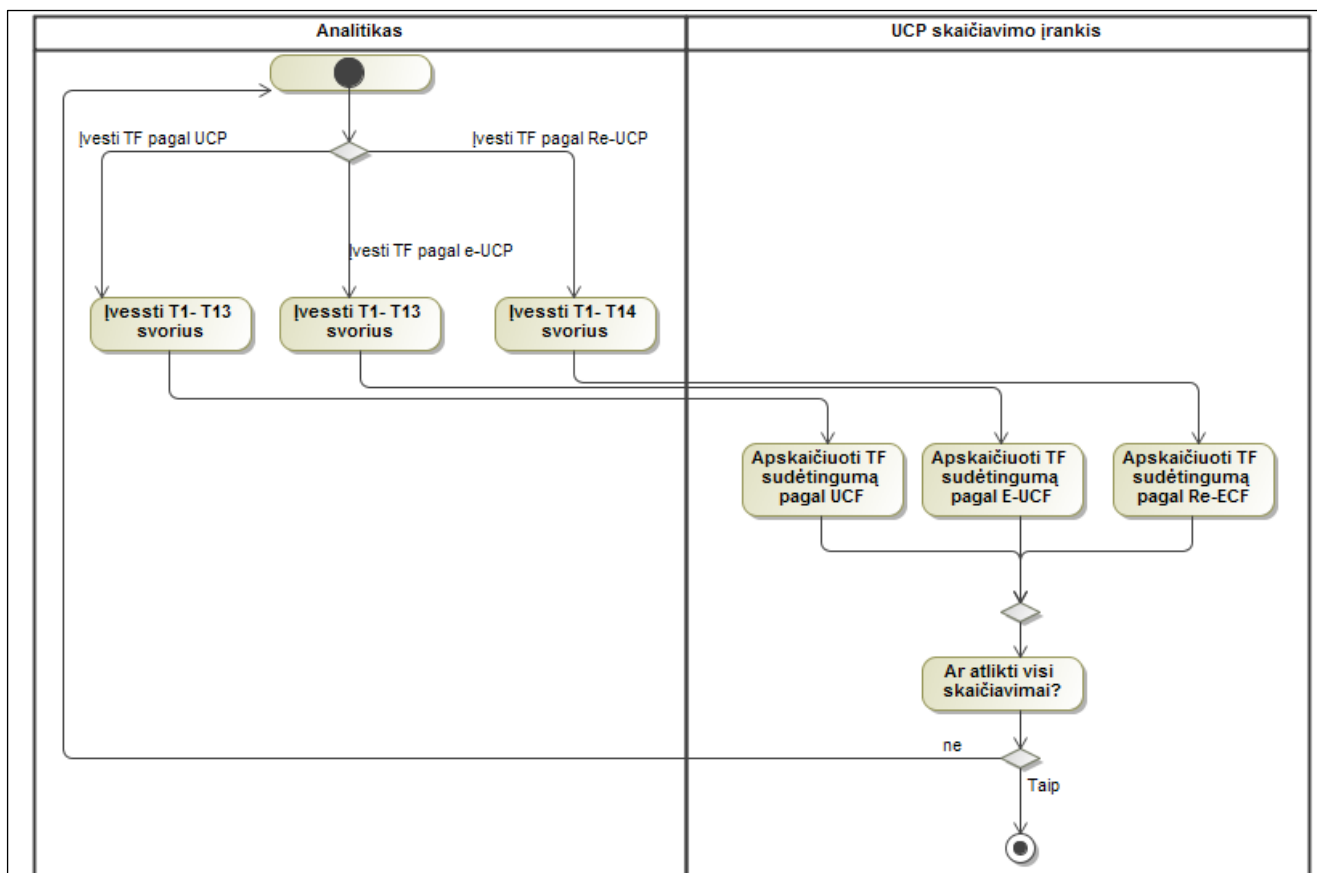


2.5 pav. „Apskaičiuoti panaudojimo atvejų sudėtingumą“ veiklos diagrama

2.4 lentelė PA „Apskaičiuoti panaudojimo atvejų sudėtingumą“ formalus aprašymas

PA „Apskaičiuoti panaudojimo atvejų sudėtingumą“							
Prieš sąlygą	Vartotojas įsijungęs UCP skaičiavimo įrankį. Vartotojas turi sistemos panaudojimo atvejų diagramą.						
Sužadinimo sąlyga	Sistema nori įvertinti panaudojimo atvejus.						
Susiję panaudojimo atvejai	<table border="1"> <tr> <td>Išplečia PA</td> <td></td> </tr> <tr> <td>Apima PA</td> <td></td> </tr> <tr> <td>Specializuoja PA</td> <td>Apskaičiuoti UCP.</td> </tr> </table>	Išplečia PA		Apima PA		Specializuoja PA	Apskaičiuoti UCP.
Išplečia PA							
Apima PA							
Specializuoja PA	Apskaičiuoti UCP.						
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai						
1. Iš panaudojimo atvejų diagramos identifikuojami PA svoriai pagal UCP metodą.	Sistema apskaičiuoja PA sudėtingumą pagal UCP metodą.						
2. Iš panaudojimo atvejų diagramos identifikuojami PA svoriai pagal E–UCP metodą.	Sistema apskaičiuoja PA sudėtingumą pagal E–UCP metodą.						
3. Iš panaudojimo atvejų diagramos identifikuojami PA svoriai pagal Re–UCP metodą.	Sistema apskaičiuoja PA sudėtingumą pagal Re–UCP metodą.						
Po sąlyga:	Sistemoje išsaugoti skaičiavimų duomenys.						
Alternatyvūs scenarijai							
A1. Jei įvestos ne visos reikšmės.	Sistema siūlo įvesti trūkstamas reikšmes.						

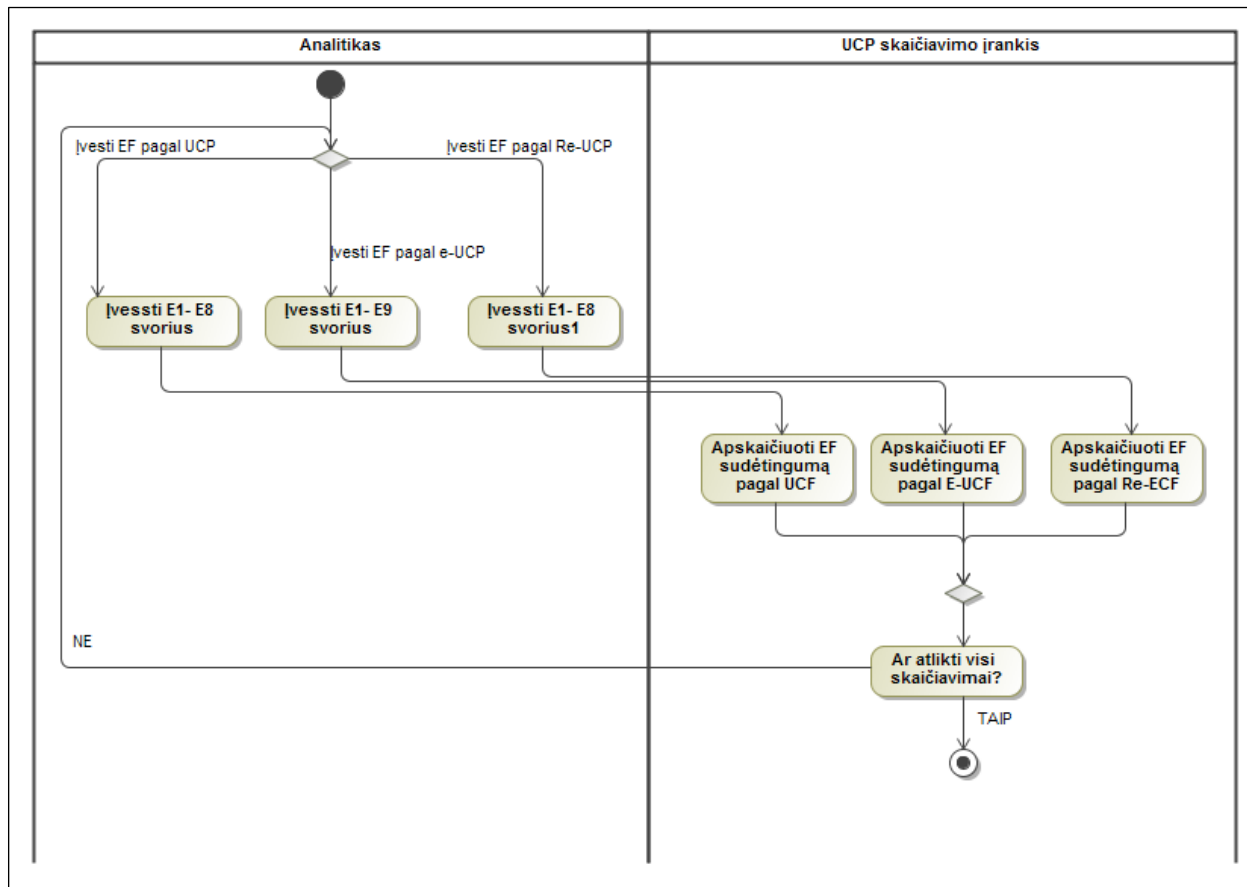
Dar vienas žingsnis skaičiuojant UCP yra techninių ir aplinkos faktorių įvertinimas (2.6 pav., 2.7 pav.). Tiek techniniai, tiek aplinkos faktoriai yra vertinami tokiu pačiu principu, įvedant faktorių svorius, pagal kiekvieną metodą, o tuomet sistema apskaičiuoja techninio ar aplinkos faktoriaus sudėtingumą.



2.6 pav. „Apskaičiuoti techninius faktorius“ veiklos diagrama

2.5 lentelė PA „Apskaičiuoti techninius faktorius“ formalus aprašymas

PA „Apskaičiuoti techninius faktorius“		
Prieš sąlygą		Vartotojas įsijungęs UCP skaičiavimo įrankį. Vartotojas turi sistemos panaudojimo atvejų diagramą.
Sužadinimo sąlyga		Vartotojas nori įvertinti panaudojimo atvejus.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	Apskaičiuoti UCP.
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1.	Įvertinami TF pagal UCP metodą.	Sistema apskaičiuoja TF pagal UCP metodą.
2.	Įvertinami TF pagal E–UCP metodą.	Sistema apskaičiuoja TF pagal E–UCP metodą.
3.	Įvertinami TF pagal Re–UCP metodą.	Sistema apskaičiuoja TF pagal Re–UCP metodą.
Po sąlyga:		Sistemoje išsaugoti skaičiavimų duomenys.
Alternatyvūs scenarijai		
A1.	Jei įvestos ne visos reikšmės.	Sistema siūlo įvesti trūkstamas reikšmes.



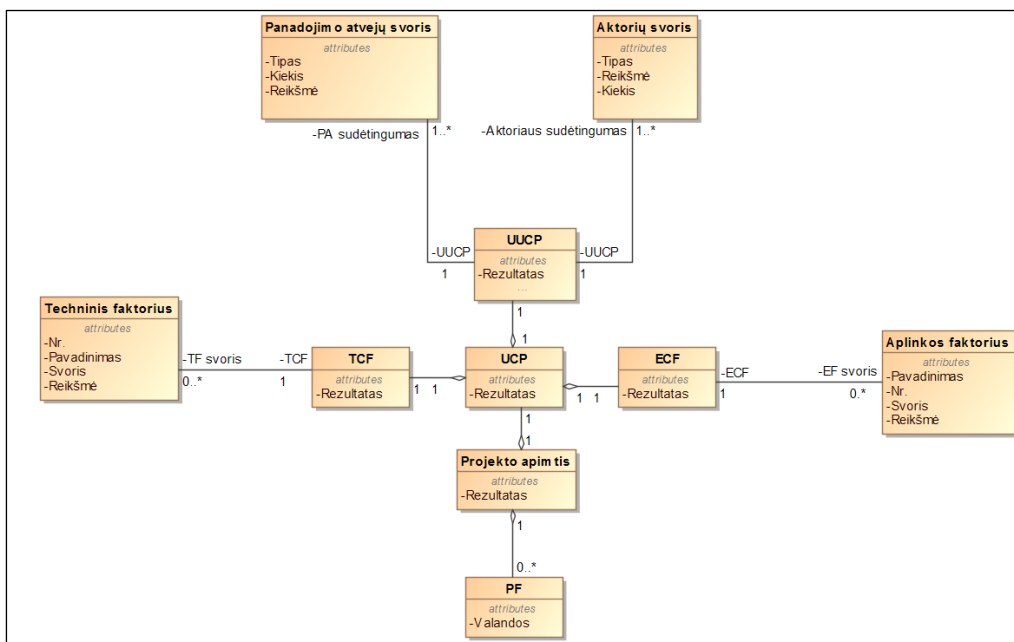
2.7 pav. „Aplinkos faktorių apskaičiavimas“

2.6 lentelė PA “ Apskaičiuoti aplinkos faktorius“ formalus aprašymas

PA „Apskaičiuoti aplinkos faktorius“		
Prieš sąlygą		Vartotojas įsijungęs UCP skaičiavimo įrankį. Vartotojas turi sistemos panaudojimo atvejų diagramą.
Sužadinimo sąlyga		Vartotojas nori įvertinti panaudojimo atvejus.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	Apskaičiuoti UCP.
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Įvertinami EF pagal UCP metodą.		Sistema apskaičiuoja EF pagal UCP metodą.
2. Įvertinami EF pagal E–UCP metodą.		Sistema apskaičiuoja EF pagal E–UCP metodą.
3. Įvertinami EF pagal Re–UCP metodą.		Sistema apskaičiuoja EF pagal Re–UCP metodą.
Po sąlyga:		Sistemoje išsaugoti skaičiavimų duomenys.
Alternatyvūs scenarijai		
A1. Jei įvestos ne visos reikšmės.		Sistema siūlo įvesti trūkstamas reikšmes.

2.2. Dalykinės srities modelis

Dalykinės srities modelis, atitinkantis UCP metodą, susideda iš 9 esybių, kurios yra susietos ryšiais, apibūdinančiais panaudojimo atvejų taškų metodo skaičiavimą. Šio modelio pagrindu bus kuriamas panaudojimo atvejų taškų metodo apskaičiavimo įrankis (2.8 pav.).



2.8 pav. UCP dalykinės srities modelis

2.3. Vartotojo sąsajos modelis

Vartotojo sąsaja *MagicDraw* aplinkoje susidės iš skirtingų profilių kiekvienam metodui, kurie skirti vertinti TF, EF, PA ir aktorius (2.9 pav.). Analitikas galės pasirinkti koku metodu nori atlikti skaičiavimus ir pagal tai įvesti atitinkamas reikšmes (2.10 pav.). Vartotojas suvedęs reikšmes, visiems metodams, norėdamas pamatyti rezultatus, turės sugeneruoti ataskaitą. Ataskaita bus pateikiama *.doc* formato faile, joje bus galima pamatyti rezultatus, apskaičiuotus skirtingais metodais, juos palyginti, pamatyti vidurkį. 2.10 pav. – pateiktas ataskaitos fragmentas (panaudojimo atvejų svorio vertinimas), visų sekančių metodo žingsnių skaičiavimas ataskaitoje bus vaizduojamas tokiu pačiu principu.

Use Case Complexity	Example	Weight
Low Complexity	1-3 Transactions	5
Average Complexity	4-7 Transactions	10
High Complexity	> 7 Transactions	15

Actor Complexity	Example	Weight
Low Complexity	External system with well defined API	1
Average Complexity	Either a human with a command line interface or a external system via some protocols or a database	2
High Complexity	A human with a GUI or a web page	3

«useCaseProjectEstimation»
UCP

(E1 Familiar with RUP = 3,
E2 Application experience = 3,
E3 Object-oriented experience = 4,
E4 Lead analyst capability = 3,
E5 Motivation = 1,
E6 Stable requirements = 2,
E7 Part time workers = 0,
E8 Difficult programming language = 2,
T1 Distributed System = 0,
T10 Concurrent = 5,
T11 Security features = 5,
T12 Access for third parties = 3,
T13 Special training required = 0,
T2 Response adjectives = 3,
T3 End-user efficiency = 3,
T4 Complex processing = 2,
T5 Reusable code = 0,
T6 Easy to install = 0,
T7 Easy to use = 0,
T8 Portable = 1,
T9 Easy to change = 4)

«useCaseProjectEstimation»
E-UCP

(E1 Familiar with RUP = 3,
E2 Application experience = 3,
E3 Object-oriented experience = 4,
E4 Lead analyst capability = 3,
E5 Motivation = 1,
E6 Stable requirements = 2,
E7 Part time workers = 0,
E8 Difficult programming language = 2,
T1 Distributed System = 0,
T10 Concurrent = 5,
T11 Security features = 5,
T12 Access for third parties = 3,
T13 Special training required = 0,
T2 Response adjectives = 3,
T3 End-user efficiency = 3,
T4 Complex processing = 2,
T5 Reusable code = 0,
T6 Easy to install = 0,
T7 Easy to use = 0,
T8 Portable = 1,
T9 Easy to change = 4)

«useCaseProjectEstimation»
Re-UJCP

(E1 Familiar with RUP = 3,
E2 Application experience = 3,
E3 Object-oriented experience = 4,
E4 Lead analyst capability = 3,
E5 Motivation = 1,
E6 Stable requirements = 2,
E7 Part time workers = 0,
E8 Difficult programming language = 2,
E9 Project methodology = 3,
T1 Distributed System = 0,
T10 Concurrent = 5,
T11 Security features = 5,
T12 Access for third parties = 3,
T13 Special training required = 0,
T2 Response adjectives = 3,
T3 End-user efficiency = 3,
T4 Complex processing = 2,
T5 Reusable code = 0,
T6 Easy to install = 0,
T7 Easy to use = 0,
T8 Portable = 1,
T9 Easy to change = 4)
T14 Scalability

2.9 pav. MagicDraw vartotojo sąsaja

2.1.1. Aktorių sudėtingumo vertinimas

UCP

Aktoriaus sudėtingumas	Aprašymas	Svoris	Elementu skaičius	Rezultatas
Paprastas	Programos vartotojo sąsaja	1	0	0
Vidutinis	Interaktyvi vartotojo sąsaja	2	1	2
Aukštas	Grafinė vartotojo sąsaja	3	1	3
Unadjusted Actor Svoriss(UAW)				6

Re-UCP

Aktoriaus sudėtingumas	Aprašymas	Svoris	Elementu skaičius	Rezultatas
Paprastas	Programos vartotojo sąsaja	1	0	0
Vidutinis	Interaktyvi vartotojo sąsaja	2	0	0
Aukštas	Grafinė vartotojo sąsaja	3	1	3
Kritinis	Aktoriaus veikimas su modeliais dirbančiais realiu laiku.	4	1	4
Unadjusted Actor Svoriss(UAW)				7

E-UCP

Aktoriaus sudėtingumas	Aprašymas	Svoris	Elementu skaičius	Rezultatas
Paprastas	Programos vartotojo sąsaja	1	0	0
Vidutinis	Interaktyvi vartotojo sąsaja	2	0	0
Aukštas	Grafinė vartotojo sąsaja	3	1	3
Labai sudėtingas	Aktoriaus veikimas su modeliais dirbančiais realiu laiku.	4	0	0
Unadjusted Actor Svoriss(UAW)				3

Palyginimas

UAW	E-UAW	Re-UAW	Vidurkis UAW
5	3	7	5

2.10 pav. Sugeneruotos rezultatų ataskaitos fragmentas

2.4. Nefunkciniai reikalavimai

2.7 lentelė Nefunkciniai reikalavimai

Reikalavimo tipas	Reikalavimo aprašymas
Panaudojamumas	Turi būti galimybė koreguoti duomenis
Panaudojamumas	Turi būti galimybė koreguoti diagramas
Patikimumas	Turi būti galimybė taisyti klaidas
Panaudojamumas	Projektai turi būti specifikuojami naudojant UML priemones.
Panaudojamumas	Informacija turi būti imama iš UML panaudojimo atvejų diagramų

2.5. UCP skaičiavimo įrankio projektas

2.5.1. Projekto tikslas

Projekto tikslas – suprojektuoti ir realizuoti tradicinio UCP metodo, išplėsto UCP metodo ir patikslinto UCP metodo skaičiavimo galimybes *MagicDraw* programos aplinkoje, pateikti rezultatus skirtingais metodais bei suteikti galimybę lyginti rezultatus.

2.5.2. Metodų taikymo etapai

Pirmas etapas, vertinant projekto apimtį, yra aktorių svorio nustatymas. UCP metodikose aktorių svoriai yra klasifikuojami skirtingai (2.8 lentelė).

2.8 lentelė Aktorių svorio koeficientai pagal skirtingus metodus

UCP metodas		E-UCP metodas		Re-UCP metodas	
Aktoriaus sudėtingumas	Svoris	Aktoriaus sudėtingumas	Svoris	Aktoriaus sudėtingumas	Svoris
		Labai paprastas	0.5		
Paprastas	1	Paprastas	1	Paprastas	1
		Mažiau nei vidutinis	1.5		
Vidutinis	2	Vidutinis	2	Vidutinis	2
Sudėtingas	3	Sudėtingas	2.5	Sudėtingas	3
		Labai sudėtingas	3		
		Kritinis	3.5	Kritinis	4

Pagal visus metodus aktorių sudėtingumas yra apskaičiuojamas taip pat – t.y. susumuojant aktorius ir padauginant iš aktorių svorio.

UCP metodo aktorių sudėtingumo vertinimas apskaičiuojamas taip:

$$UAW = \sum_{c \in C} aSvoris(c) \times aKiekis(c); \quad (1)$$

čia $C = \{\text{paprastas, vidutinis, sudėtingas}\}$;

$aSvoris(c): \{1, 2, 3\}$;

UAW – tradicinio UCP metodo nekoreguotinas aktoriaus svoris;

C – aktoriaus sudėtingumas;

$aSvoris$ – aktoriaus svoris;

$aKiekis$ – aktoriaus kiekis.

E–UCP metodo aktorių sudėtingumo vertinimas apskaičiuojamas taip:

$$EUAW = \sum_{c \in C} aSvoris(c) \times aKiekis(c); \quad (2)$$

čia $C = \{\text{labai paprastas, paprastas, mažiau nei vidutinis, vidutinis, sudėtingas, labai sudėtingas, sudėtingiausias}\}$;

$aSvoris(c): \{0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$;

$EUAW$ – išplėsto UCP metodo nekoreguotinas aktoriaus svoris;

Re–UCP metodo aktorių sudėtingumo vertinimas apskaičiuojamas taip:

$$ReUAW = \sum_{c \in C} aSvoris(c) \times aKiekis(c); \quad (3)$$

čia $C = \{\text{paprastas, vidutinis, sudėtingas, kritinis}\}$;

$aSvoris(c): \{1, 2, 3, 4\}$;

$EUAW$ – patikslinto UCP metodo nekoreguotinas aktoriaus svoris.

Sekantis projekto apimties vertinimo etapas yra panaudojimo atvejų svorio skaičiavimas (2.9 lentelė).

2.9 lentelė Panaudojimo atvejų svorio koeficientai pagal skirtingus metodus

UCP metodas		E–UCP metodas		Re–UCP metodas	
PA sudėtingumas	Svoris	PA sudėtingumas	Svoris	PA sudėtingumas	Svoris
Paprastas	5	Paprastas	5	Paprastas	5
Vidutinis	10	Vidutinis	10	Vidutinis	10
Sudėtingas	15	Sudėtingas	15	Sudėtingas	15
		Kritinis	20	Kritinis	20

UCP metodo PA sudėtingumo vertinimas apskaičiuojamas taip:

$$uSvoris(u) = \begin{cases} \text{paprastas if } trans(u) < 4 \\ \text{vidutinis if } 4 \ll trans(u) \ll 7; \\ \text{sudėtingas if } trans(u) > 7 \end{cases} \quad (4)$$

čia U – panaudojimo atvejis;

$trans(u)$ – transakcijų kiekis;

$$UUCW = \sum_{c \in C} uSvoris(c) \times uKiekis(c) \quad (5)$$

čia $C = \{\text{paprastas, vidutinis, sudėtingas}\}$;

$uSvoris(u): \{5, 10, 15\}$;

$UUCW$ – tradicinio UCP metodo nekoreguotinas panaudojimo atvejo svoris;

$uSvoris$ – panaudojimo atvejo svoris;

$uKiekis$ – panaudojimo atvejo kiekis.

E–UCP metodo PA sudėtingumo vertinimas apskaičiuojamas taip:

$$uSvoris(u) = \begin{cases} \text{paprastas if } trans(u) < 4 \\ \text{vidutinis if } 5 \ll trans(u) \ll 8 \\ \text{sudėtingas if } 9 \ll trans(u) \ll 15; \\ \text{labai sudėtingas if } trans(u) > 15 \end{cases} \quad (6)$$

čia U – panaudojimo atvejis;

$trans(u)$ – transakcijų kiekis;

$$EUUCW = \sum_{c \in C} uSvoris(c) \times uKiekis(c) \quad (7)$$

$C = \{\text{paprastas, vidutinis, sudėtingas, labai sudėtingas}\}$;

$uSvoris(u)$: {5, 10, 15, 20};

EUUCW – išplėsto UCP metodo nekoreguotinas panaudojimo atvejo svoris.

Re-UCP metodo PA sudėtingumo vertinimas apskaičiuojamas taip:

$$uSvoris(u) = \begin{cases} \text{paprastas if } trans(u) < 4 \\ \text{vidutinis if } 5 \ll trans(u) \ll 8 \\ \text{sudėtingas if } 9 \ll trans(u) \ll 15 \\ \text{kritinis if } trans(u) > 15 \end{cases}; (8)$$

čia U – panaudojimo atvejis;

$trans(u)$ – transakcijų kiekis.

$$ReUUCW = \sum_{c \in C} uSvoris(c) \times uKiekis(c); \quad (9)$$

$C = \{\text{paprastas, vidutinis, sudėtingas, kritinis}\}$;

$uSvoris(u)$: {5, 10, 15, 20};

REUUCW – patikslinto UCP metodo nekoreguotinas panaudojimo atvejo svoris.

Nekoreguotų panaudojimo atvejų taškų skaičiavimas pagal tradicinį UCP metodą atliekamas sudedant nekoreguotinus svorius ir panaudojimo atvejus $UUCP = UAW + UUCW$, analogiškai atliekama ir pagal kitus metodus $EUUCP = EUAW + EUUCW$, $REUUCP = REUAW + REUUCW$.

Techninių faktorių skaičiavimas atliekamas susumavus techninių faktorių svorių ir priskirtų reikšmių sandaugas.

2.10 lentelė Techninių faktorių svorio koeficientai pagal skirtingus metodus

UCP metodas			E-UCP metodas			Re-UCP metodas		
Faktorius	Pavadinimas	Svoris	Faktorius	Pavadinimas	Svoris	Faktorius	Pavadinimas	Svoris
T1	Sistemos išskaidymas	2	T1	Sistemos išskaidymas	2	T1	Sistemos išskaidymas	2
T2	Sistemos našumas	1	T2	Sistemos našumas	1	T2	Sistemos našumas	1
T3	Galutinio vartotojo efektyvumas	1	T3	Galutinio vartotojo efektyvumas	1	T3	Galutinio vartotojo efektyvumas	1
T4	Sudėtinis vidinis algoritmų apdorojimas	1	T4	Sudėtinis vidinis algoritmų apdorojimas	1	T4	Sudėtinis vidinis algoritmų apdorojimas	1
T5	Kodo universalumas	1	T5	Kodo universalumas	1	T5	Kodo universalumas	1
T6	Paprastas diegimas	0.5	T6	Paprastas diegimas	0.5	T6	Paprastas diegimas	0.5
T7	Paprastas naudojimas	0.5	T7	Paprastas naudojimas	0.5	T7	Paprastas naudojimas	0.5
T8	Perkeliamumas	1	T8	Perkeliamumas	1	T8	Perkeliamumas	1
T9	Paprastas keitimas	1	T9	Paprastas keitimas	1	T9	Paprastas keitimas	1
T10	Lygiagretumas	1	T10	Lygiagretumas	1	T10	Lygiagretumas	1
T11	Specialių saugumo savybių įtraukimas	1	T11	Specialių saugumo savybių įtraukimas	1	T11	Specialių saugumo savybių įtraukimas	1
T12	Trečiųjų šalių tiesioginis priėjimas	1	T12	Trečiųjų šalių tiesioginis priėjimas	1	T12	Trečiųjų šalių tiesioginis priėjimas	1
T13	Specialių vartotojo apmokymų priemonių reikalavimas.	1	T13	Specialių vartotojo apmokymų priemonių reikalavimas.	1	T13	Specialių vartotojo apmokymų priemonių reikalavimas.	1
						T14	Mastelis	2

UCP metodo TF sudėtingumo vertinimas apskaičiuojamas taip:

$$TCF = 0.6 + (0.01 \times \sum_{i=1}^{13} TFsvoris_i \times reikšmė_i); \quad (10)$$

čia $TFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0 – 5}.

E-UCP metodo TF sudėtingumo vertinimas apskaičiuojamas taip:

$$ETCF = 0.6 + (0.01 \times \sum_{i=1}^{13} TFsvoris_i \times reikšmė_i); \quad (11)$$

čia $TFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0–5}.

Re–UCP metodo TF sudėtingumo vertinimas apskaičiuojamas taip:

$$ReTCF = 0.6 + (0.01 \times \sum_{i=1}^{14} TFsvoris_i \times reikšmė_i); \quad (12)$$

čia $TFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0 – 5}.

Aplinkos faktoriai yra apskaičiuojami analogišku principu kaip ir techniniai faktoriai (13 – 15 formulės).

2.11 lentelė Aplinkos faktorių svorio koeficientai pagal skirtingus metodus

UCP metodas			E–UCP metodas			Re–UCP metodas		
Faktoriai	Pavadinimas	Svoris	Faktoriai	Pavadinimas	Svoris	Faktoriai	Pavadinimas	Svoris
E1	Komandos susipažinimas su projektu	1.5	E1	Komandos susipažinimas su projektu	1.5	E1	Komandos susipažinimas su projektu	1.5
E2	Komandos patirtis su aplikacija	0.5	E2	Komandos patirtis su aplikacija	0.5	E2	Komandos patirtis su aplikacija	0.5
E3	Objektinio programavimo patirtis	1	E3	Objektinio programavimo patirtis	1	E3	Objektinio programavimo patirtis	1
E4	Vadovaujančio analitiko gebėjimai	0.5	E4	Vadovaujančio analitiko gebėjimai	0.5	E4	Vadovaujančio analitiko gebėjimai	0.5
E5	Motyvacija	1	E5	Motyvacija	1	E5	Motyvacija	1
E6	Stabilūs reikalavimai	2	E6	Stabilūs reikalavimai	2	E6	Stabilūs reikalavimai	2
E7	Darbuotojai, dirbantys ne pilną darbo dieną	–1	E7	Darbuotojai, dirbantys ne pilną darbo dieną	–1	E7	Darbuotojai, dirbantys ne pilną darbo dieną	–1
E8	Sudėtinga programavimo kalba	–1	E8	Sudėtinga programavimo kalba	–1	E8	Sudėtinga programavimo kalba	–1
						E9	Projekto metodologija	1

UCP metodo EF sudėtingumo vertinimas apskaičiuojamas taip:

$$TCF = 1.4 + (-0.03 \times \sum_{i=1}^8 EFsvoris_i \times reikšmė_i); \quad (13)$$

čia $EFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0 – 5}.

E–UCP metodo EF sudėtingumo vertinimas apskaičiuojamas taip:

$$ETCF = 1.4 + (-0.03 \times \sum_{i=1}^8 EFsvoris_i \times reikšmė_i); \quad (14)$$

čia $EFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0 – 5}.

Re–UCP metodo EF sudėtingumo vertinimas apskaičiuojamas taip:

$$ReTCF = 1.4 + (-0.03 \times \sum_{i=1}^9 EFsvoris_i \times reikšmė_i); \quad (15)$$

čia $EFsvoris$ – techninio faktoriaus svoris;

Reikšmė: {0 – 5}.

UCP apskaičiavimas atliekamas sudauginus gautus panaudojimo atvejų taškus su techninių ir aplinkos faktorių sudėtingumo rezultatu (formulės (16)-(20)).

$$UCP = UUCP \times TCF \times ECF; \quad (16)$$

$$EUUCP = EUUCP \times ETCF \times EECF; \quad (17)$$

$$REUCP = REUCP \times RETCF \times REECF; \quad (18)$$

$$APIMTIS = UCP \times PF; \quad (19)$$

$$PF = 20val/UCP \text{ (Karnerio valandos)}. \quad (20)$$

MagicDraw programinėje įrangoje UCP skaičiavimo metodas realizuotas tik tradiciniu UCP skaičiavimo metodu. Šiame darbe realizuojami dar du metodai – išplėstasis ir patikslintas. Vartotojas nebus tik vieno metodo rėmuose, nes sistemos gali būti labai skirtingos – nuo paprastos iki labai

sudėtingos. Nuo to skiriasi panaudojimo atvejų, aktorių klasifikavimas, techniniai ir aplinkos faktoriai. Visais atvejais vartotojas turi turėti pilnai pabaigtą panaudojimo atvejų diagramą.

3. PANAUDOJIMO ATVEJŲ TAŠKŲ METODO REALIZACIJOS PROJEKTAS

Panaudojimo atvejų taškų metodo tyrimo projektas buvo įgyvendintas naudojant *MagicDraw* programinę įrangą.

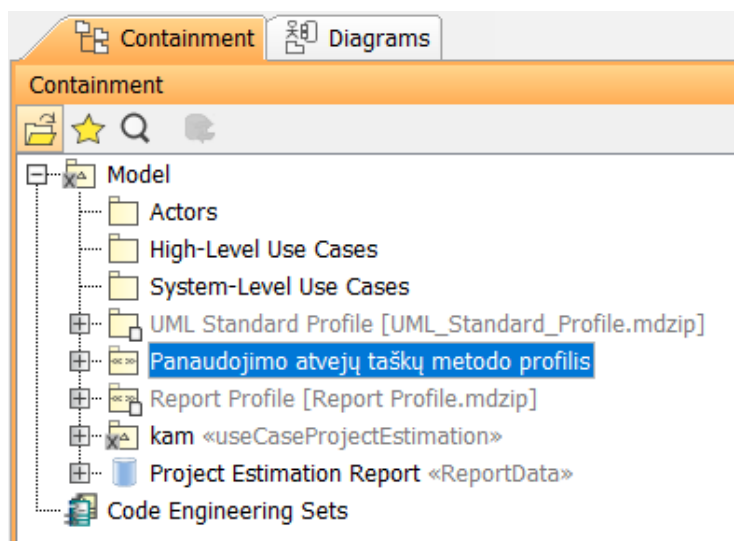
Vietoj standartinio skaičiavimo vienu panaudojimo atvejų metodu buvo pasirinkta skaičiavimus atlikti trimis metodų variacijomis, palyginti metodų rezultatus, išvesti vidurkius sugeneruojant keturias ataskaitas. Pagrindinė ataskaita yra bendra metodų skaičiavimo rezultatų ataskaita, kuri neišsamiai pateikia skaičiavimo tarpinius rezultatus bei galutinę projekto apimtį pagal kiekvieną metodą. Trys ataskaitos yra skirtos išsamiesiems vieno metodo rezultatams pateikti.

3.1. Projekto įgyvendinimas MagicDraw programinėje įrangoje

3.1.1. Panaudojimo atvejų taškų metodo profilis realizuotas MagicDraw programinėje įrangoje

Įgyvendinat metodų realizaciją *MagicDraw* programinėje įrangoje buvo sukurtas pavyzdinis projektas, kurio tikslas vartotojui suteikti informaciją kaip reikėtų naudoti *MagicDraw* programinę įrangą, norit sugeneruoti teisingą projekto vertinimo ataskaitą, paremtą UCP, e-UCP ir Re-UCP metodais.

Realizuojant projektą, svarbiausias objektas yra profilis (3.1 pav.). Profilis yra apibūdinamas kaip atskiras modelis, kuris turi būti įkeltas pradedant darbą su projektu. Projekte yra naudojami tik tie modeliai, kurie yra skirti būtent tam projektui. Profilis buvo aprašytas naudojant UML kalbą, jis suteikė galimybę praplėsti jau standartiškai esančią UML specifikaciją panaudojimo atvejų diagramoms, kurti bei generuoti ataskaitas, apskaičiuojančias projekto apimtį pagal skirtingus metodus.



3.1 pav. Panaudojimo atvejų taškų metodo profilis

Svarbiausi profilio elementai ataskaitos formavimui:

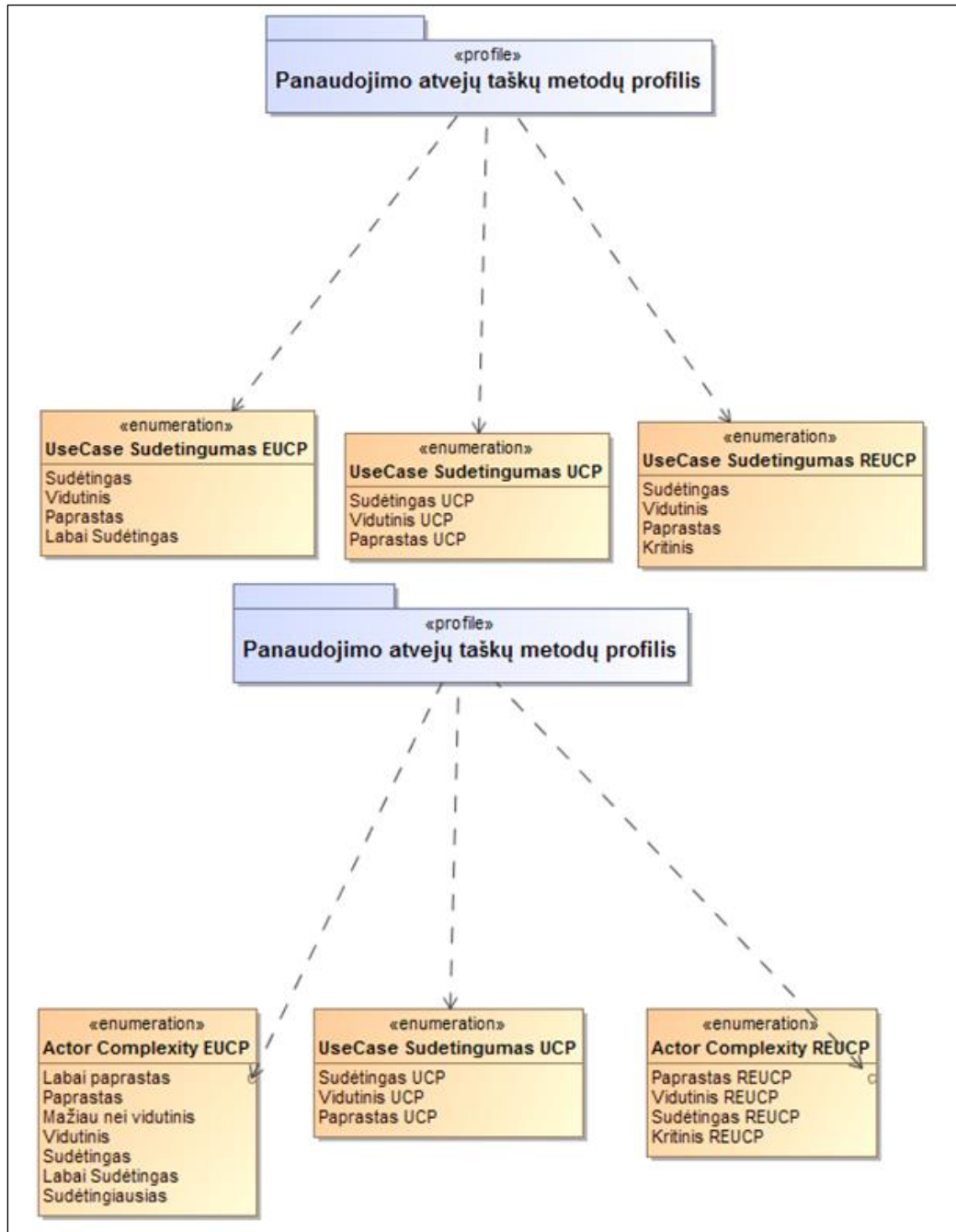
- sąrašai (angl. *enumerations*);
- pritaikymo elementų (angl. *customization*);
- stereotipai (angl. *stereotype*);
- savybės (angl. *property*).

3.1.2. Sąrašai (angl. *enumerations*)

Ataskaitos skaičiavimams yra reikalingos aktorių ir panaudojimo atvejų sudėtingumo reikšmės. Taip pat techninių ir aplinkos faktorių reikšmės.

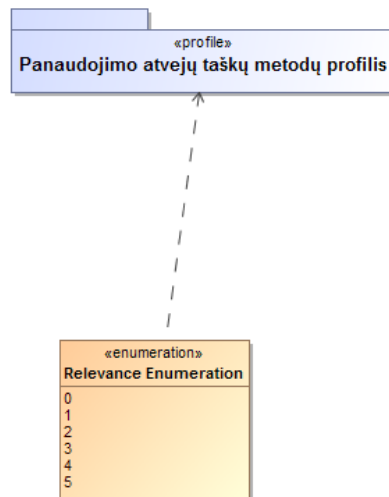
Todėl norint įtraukti į ataskaitą reikšmes buvo panaudotas sąrašo (angl. *enumeration*) elementas, kuris yra skirtas parinkti norimą reikšmę.

Panaudojus pasirinkties (angl. *enumeration*) elementą, *MagicDraw* vartotojo sąsaja papildė tam tikrais elementais. Šiuo atveju aktorių ar panaudojimo atvejų savybių langas buvo papildytas pasirinkties (angl. *dropdown*) elementais, leidžiančiais priskirti norimą reikšmę prie konkretaus pasirinkto aktoriaus ar panaudojimo atvejo (3.3 pav.).



3.2 pav. Aktorių ir panaudojimo atvejų sudėtingumo sąrašai

Norint pasirinkti techninį faktorių, diagramoje buvo sukurtas paketo (angl. *package*) elementas. Atidarius jo savybių langą galima prie techninio faktoriaus priskirti norimas reikšmes naudojant žymas (3.4 pav.).

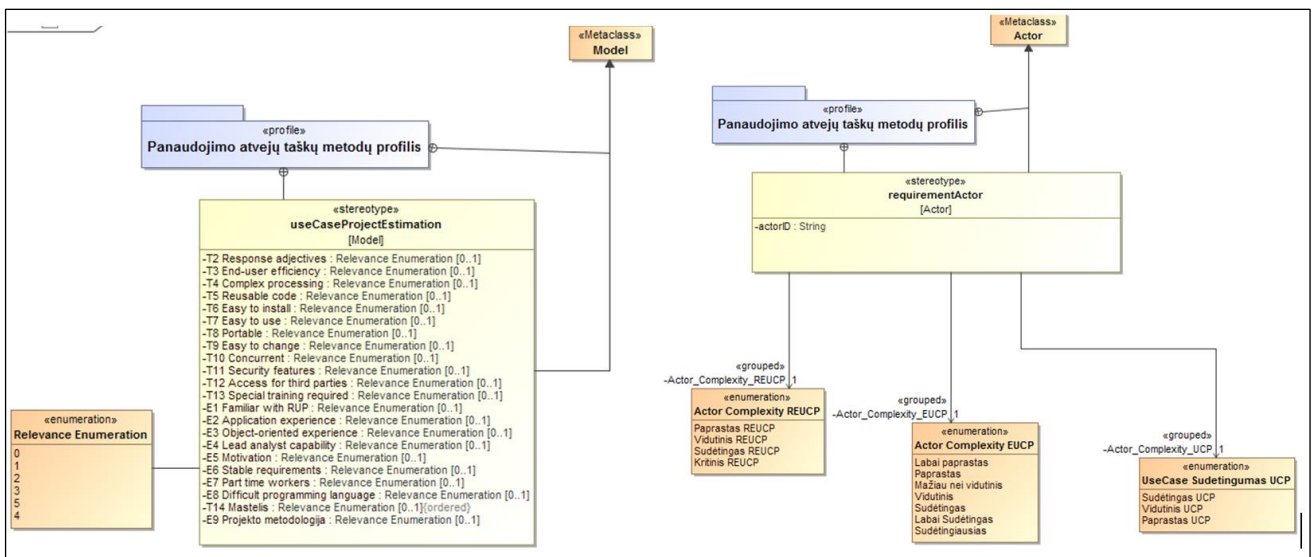


3.3 pav. Techninių ir aplinkos faktorių sąrašai

3.1.3. Stereotipai (angl. stereotype)

Stereotipai išplečia egzistuojančią metaklasę tam tikroje dalykinėje srityje. Stereotipai turi savybes (angl. *properties*), kurios gali būti naudojamos kaip žymos (angl. *tag definitions*) (tai buvo panaudota realizuojant techninių ir aplinkos faktorių reikšmių pasirinkimą). Kai stereotipas yra priskiriamas modelio elementui, savybių reikšmės yra paverčiamos žymų reikšmėmis.

Bet koks UML modelio elementas gali būti išplėstas naudojant stereotipus. Stereotipai saugo papildomą informaciją, kuri nėra standartiškai pateikiama UML kalboje. Šiuo atveju stereotipai saugo informaciją apie panaudojimo atvejų, aktorių sudėtingumus, techninių bei aplinkos faktorių skaitines reikšmes. Klasei suteikus stereotipą pačios klasės savybės lieka nepakitusios. Stereotipo priskyrimas klasei suteikia papildomus požymius.



3.4 pav. Stereotipai skirti darbui su aktoriais ir panaudojimo atvejais.

3.1.4. Rezultatų ataskaitos

Buvo sukurtos trys ataskaitos, išvedančios skaičiavimo rezultatus, pagal UCP, e-UCP ir Re-UCP metodus. Bei viena – bendra ataskaita, išvedanti bendrus tarpinius rezultatus, pagal visus metodus bei jų vidurkius.

1.3. Apimtis

UCP Apimtis (val.)	EUCP Apimtis (val.)	REUCP (val.)	Vidurkis
381.78	498.43499999999995	474.19199999999999 99	451.46899999999999 99

UCP Apimtis (mėn.)	EUCP Apimtis (mėn.)	REUCP Apimtis (mėn.)	Vidurkis
23.86125	31.1521874999999997	29.636999999999999 993	28.216812499999999 993

UCP Apimtis (d.)	EUCP Apimtis (d.)	REUCP Apimtis (d.)	Apimtis dienomis
1.084602272727 2726	1.4160085227272725	1.3471363636363 634	1.2825823863636 36

2. Išvados

Projektui, kuriame dirba 2.0 programuotojai pagal UCP metodą atlikti prireiks **1.0846022727272726** mėnesių, pagal E-UCP metodą: **1.4160085227272725** mėnesių bei pagal Re-UCP metodą **1.3471363636363634** mėnesių.

(mėnesis: 22.0 darbo dienos)

3.5 pav. Metodų ataskaitos fragmentas

Ataskaitos buvo sukurtos naudojant *MagicDraw* programinės įrangos galimybes, *JAVA* programavimo kalba paremta *Velocity Template* technologija.

4. PANAUDOJIMO ATVEJŲ TAŠKŲ METODIKŲ REALIZACIJA IR TESTAVIMAS

4.1. Sprendimo realizacijos ir veikimo aprašas

Sprendimas buvo įgyvendintas remiantis atlikta analize, kurioje buvo išnagrinėti šie metodai:

- UCP;
- Re-UCP ;
- E-UCP .

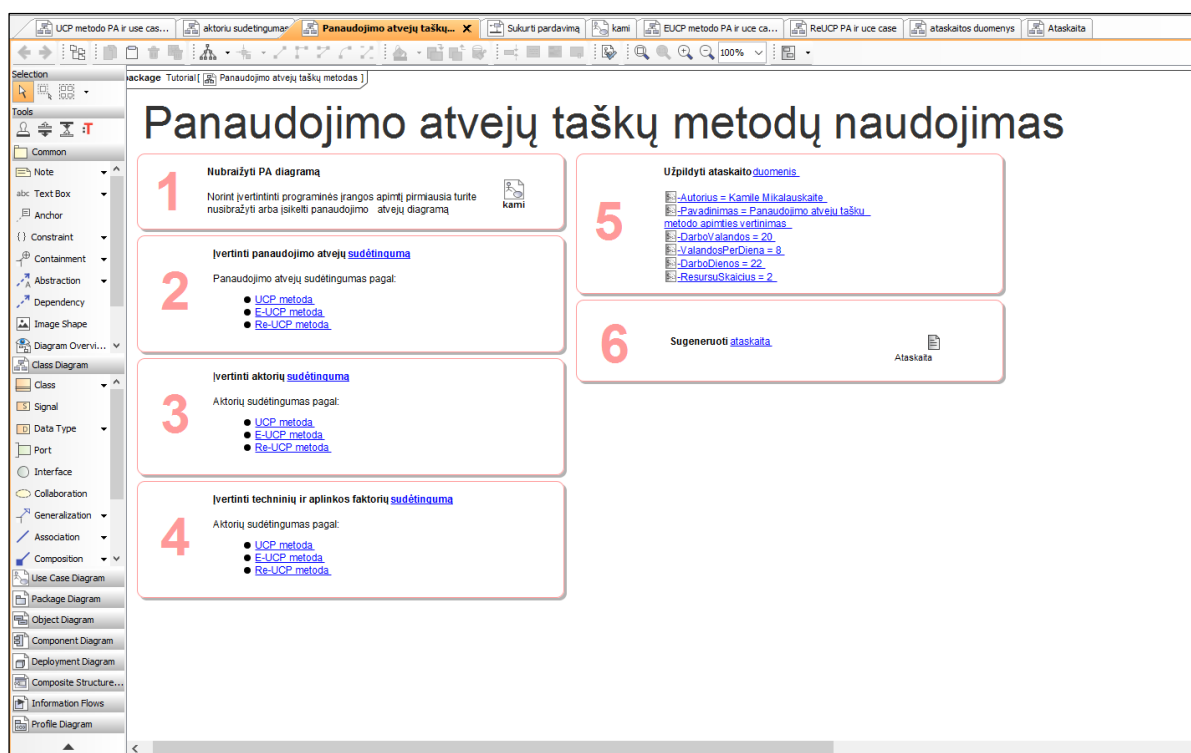
Sprendimui realizuoti pasirinkta *MagicDraw* programinė įranga, joje sukurtas pavyzdinis projektas, profilis, išplečiantis darbą su panaudojimo atvejų diagramomis ir metodo naudojimosi vedlys. Ataskaitos buvo sukurtos, išnaudojant integruotas *MagicDraw* galimybes, ataskaitą aprašant *Velocity Template* technologija.

Pirmiausia, atsidarius *MagicDraw* programinę įrangą, reikia įsikelti norimą profilį. (*Modules* → *Open Module as Project.*). Norint, kad ataskaita teisingai generuotų duomenis, reikia į projektą įsidėti tokius profilius:

- panaudojimo atvejų taškų metodo profilį;
- *UML Standart profile* – standartinis, *MagicDraw* pateikiamas UML profilis, skirtas dirbti su UML kalbą naudojančiomis diagramomis;
- *Report Data profile* – standartinis, *MagicDraw* pateikiamas profilis, skirtas darbui su ataskaitos duomenimis.

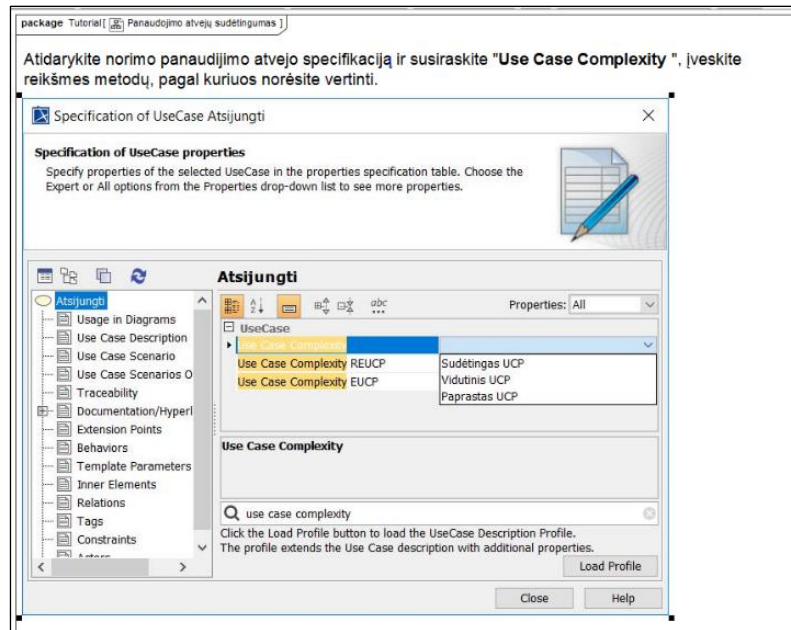
Norint gauti tikslius projekto vertinimus turi būti nubraižytos visos panaudojimo atvejų diagramos, taip pat prie jų panaudojimo atvejų ir aktorių turi būti nurodyti sudėtingumai, ir visam projektui turi būti nurodyti aplinkos ir techninių faktorių reikšmių sudėtingumai.

Naudojantis metodu pirmą kartą gali padėti sukurtas vedlys. Jame yra nurodyti kokie žingsniai ir kaip turėtų būti atliekami (4.1pav. – 4.3pav).



4.1 pav. Metodo naudojimo MagicDraw aplinkoje vedlys

Kiekviename metodo vedlio žingsnyje yra nuorodos, kurias paspaudus galima matyti išsamesnius paaiškinimus ir aprašymus (4.2 pav., 4.3 pav.).

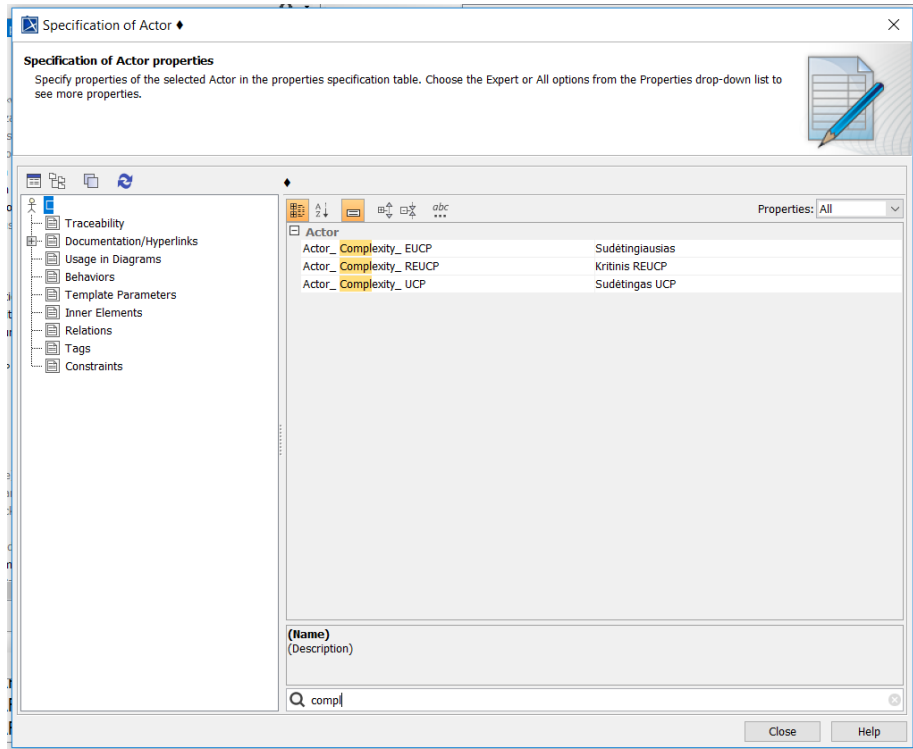


4.2 pav. Metodo naudojimo MagicDraw aplinkoje vedlys Nr. 2

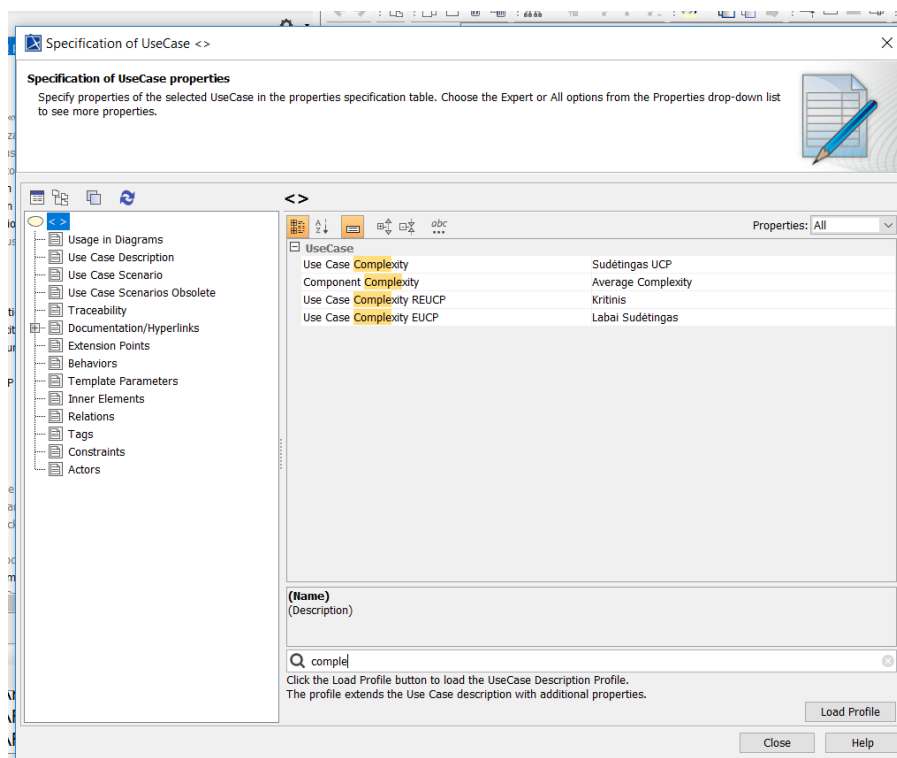
UCP METODAS		
PANAUDOJIMO ATVEJŲ SUDĖTINGUMAS		
UCP metodas - Panaudojimo atvejų sudėtingumas		
Sudėtingumas	Aprašymas	Svoris
Paprastas	1-3 Tranzakcijos	5
Vidutinis	4-7 tranzakcijos	10
Sudėtingas	> 7 Tranzakcijos	15
AKTORIŲ SUDĖTINGUMAS		
UCP metodo aktorių sudėtingumas		
Sudėtingumas	Aprašymas	Svoris
Paprastas	Programos vartotojo sąsaja	1
Vidutinis	Interaktyvi vartotojo sąsaja	2
Sudėtingas	Grafinė vartotojo sąsaja	3

4.3 pav. Metodo naudojimo MagicDraw aplinkoje vedlys Nr. 3

Norint nurodyti panaudojimo atvejo arba aktoriaus sudėtingumą reikia atidaryti aktoriaus ar panaudojimo atvejo specifikacijos langą, ten reikia susirasti aktoriaus ir panaudojimo atvejo pasirinkimo laukus. Pasirinkimo laukai yra sukurti kiekvienam metodui, su jam priklausančiom reikšmėm (4.4 – 4.5 pav.).



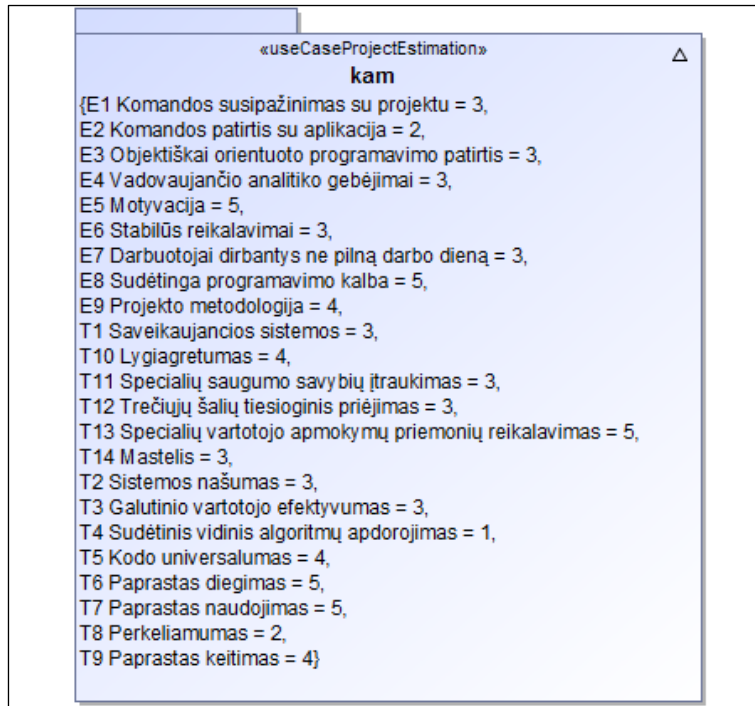
4.4 pav. Aktorių sudėtingumo pasirinkimas



4.5 pav. Panaudojimo atvejų sudėtingumo pasirinkimas

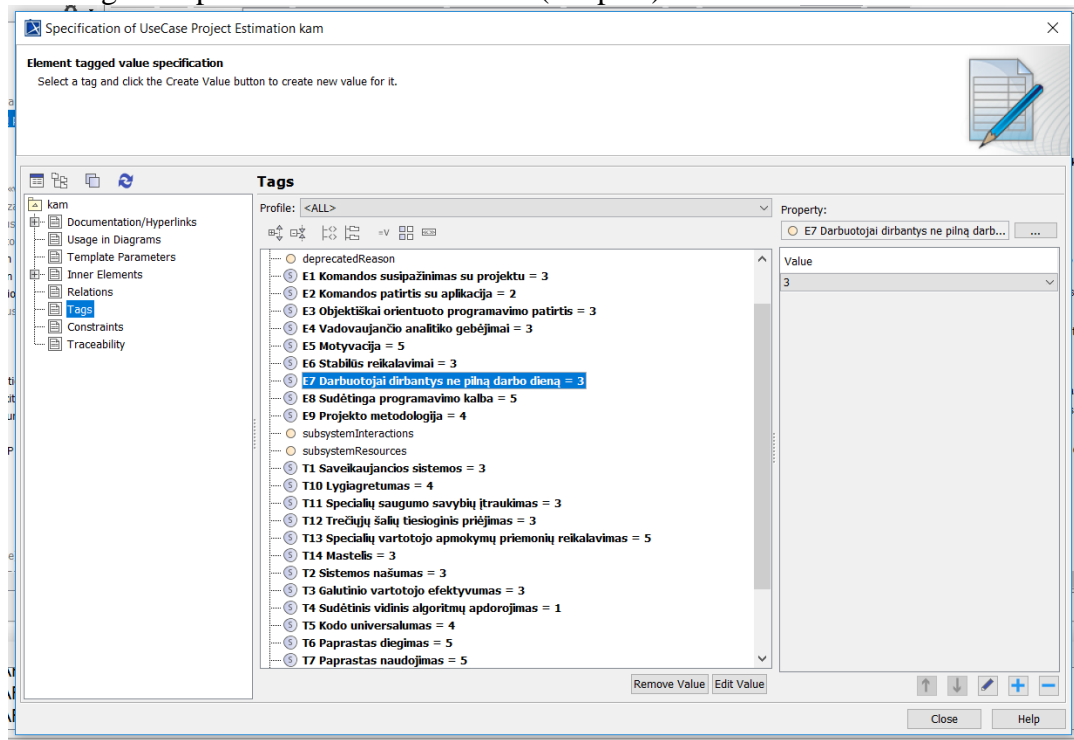
Pavyzdiniame projekte yra pateikti aprašymai apie kiekvieno iš aktorių ir panaudojimo atvejų sudėtingumą, jei reikia, galima atsižvelgti ir į tai.

Priskyrus prie kiekvieno aktoriaus ir panaudojimo atvejo sudėtingumo faktorius reikia priskirti reikšmes prie techninių ir aplinkos faktorių. Aplinkos ir techniniai faktoriai yra realizuoti panaudojant modelio tipo stereotipą, todėl, kad jis atsirastų diagramoje pirmiausia iš projekto meniu juostos reikia nutempti projekto modelio (angl. *model*) elementą į diagramos langą – diagramos lange tai bus atvaizduota kaip paketas (angl. *package*) (4.6 pav.).



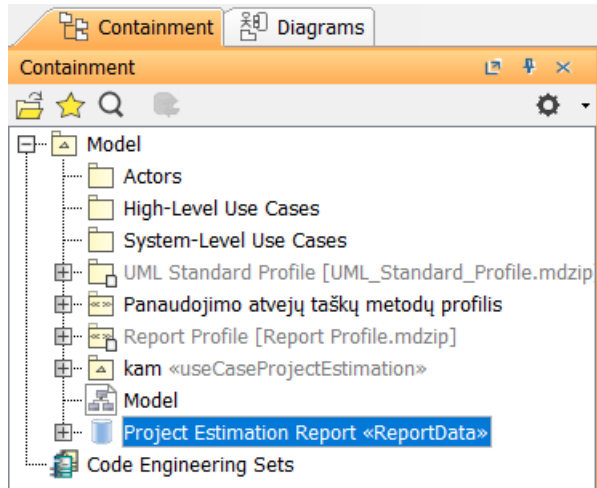
4.6 pav. Aplinkos ir techniniai faktoriai

Norint priskirti reikšmes reikia atidaryti šio elemento specifikacijų langą ir pasirinkti meniu skiltį TAGS, ten bus galima priskirti norimas reikšmes (4.7 pav.).



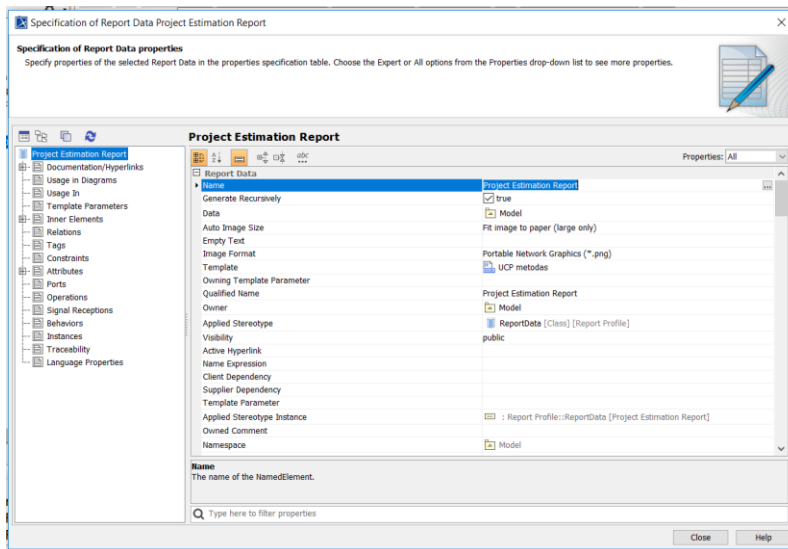
4.7 pav. Techninių ir aplinkos faktorių reikšmių priskyrimas

Priskyrus visas reikšmes prie panaudojimo atvejų, aktorių, techninių ir aplinkos faktorių reikia nustatyti tam tikrus duomenis, kurie bus paimami į ataskaitą. Tam reikia susikurti *Report Data* tipo elementą, kuriame bus nurodomos reikšmės (4.8 pav.).



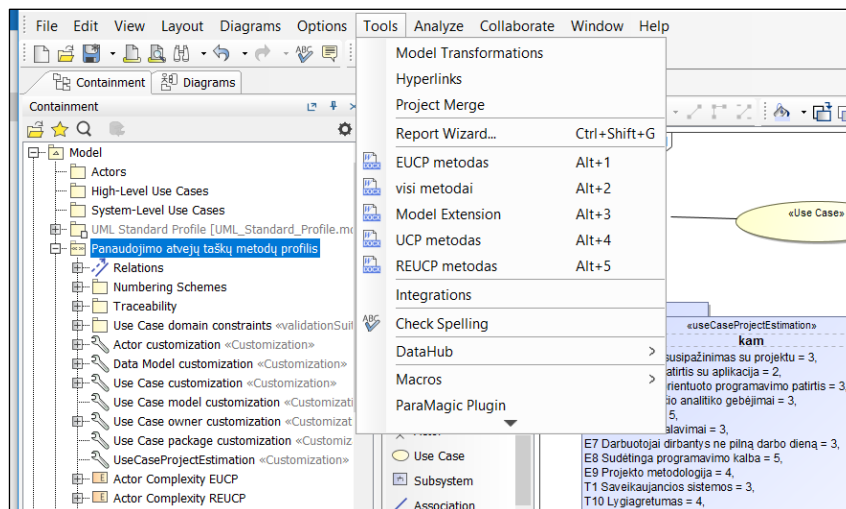
4.8 pav. Projekto ataskaitos duomenų saugojimas

Šiame žingsnyje, prieš formuojant ataskaitą reikia nurodyti kuris ataskaitos šablonas naudos šiuos duomenis (4.9 pav.).



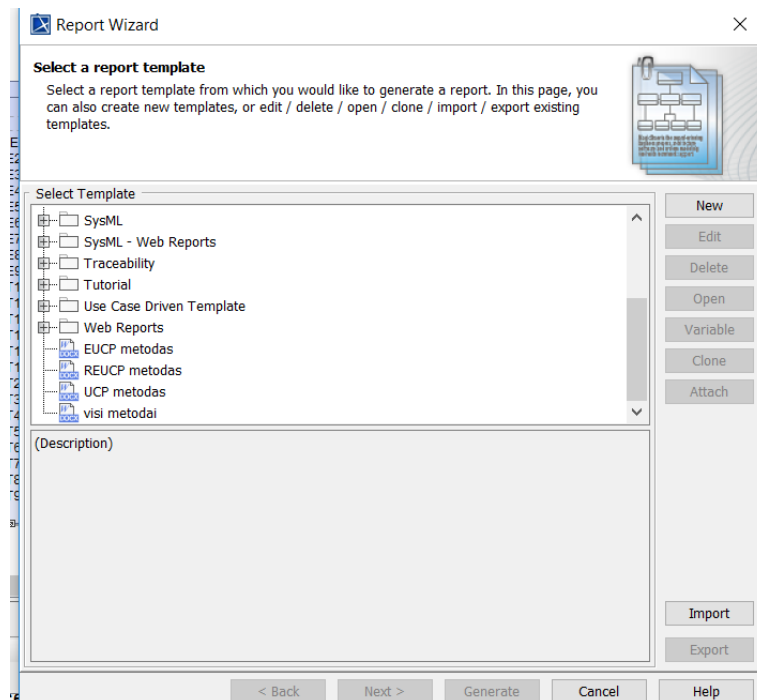
4.9 pav. Ataskaitos duomenų priskyrimas prie šablono

Suvedus visas reikiamas reikšmes, programuotojų skaičių, darbo dienų skaičių ir panašiai, galima generuoti ataskaitą. Meniu juostoje reikia pasirinkti *Tools* → *Report Wizard* (4.10 pav.).



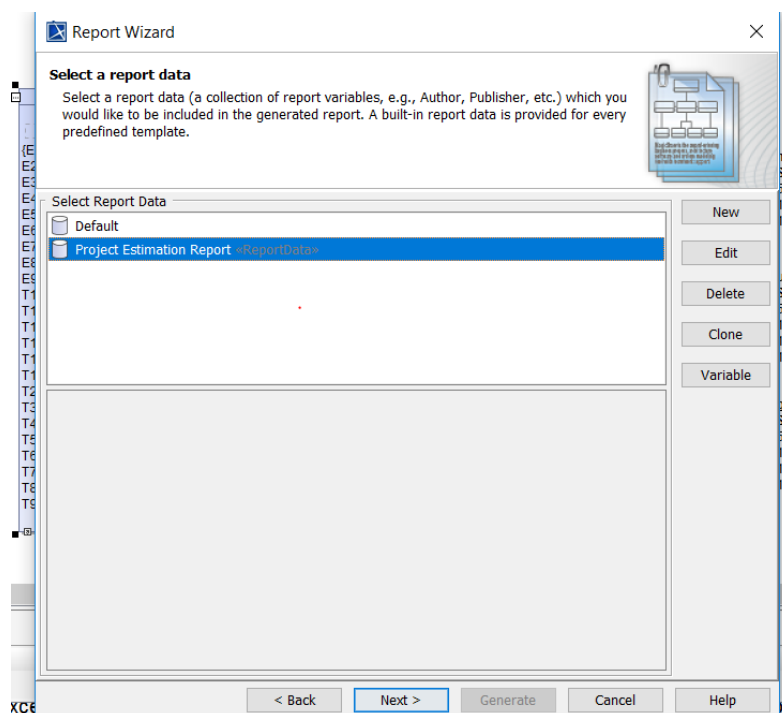
4.10 pav. Ataskaitos generavimo žingsnis nr. 1

Atsidariusiame lange reikia pasirinkti ataskaitą, kurią norite suformuoti ir spausti mygtuką *Next* (4.11 pav.).



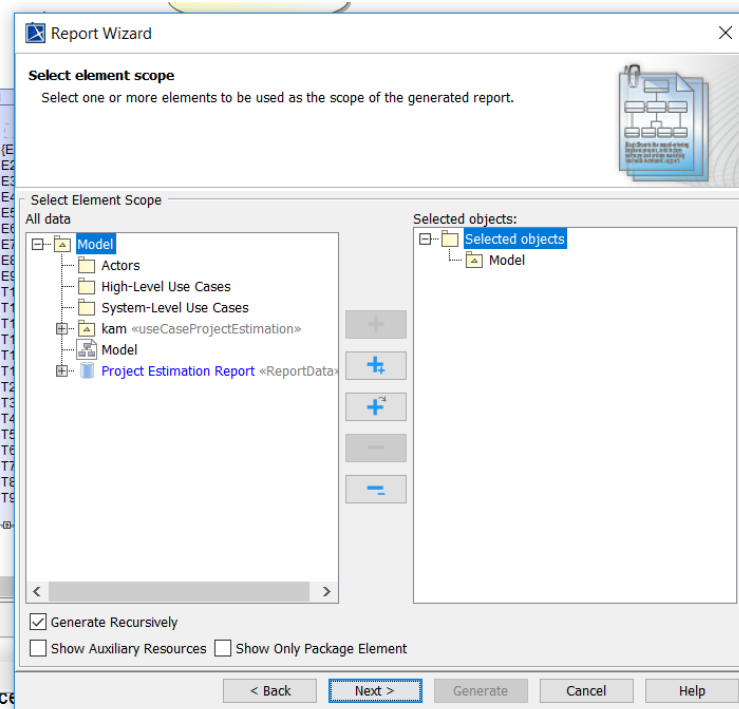
4.11 pav. Ataskaitos generavimo žingsnis nr. 2

Toliau reikia pasirinkti iš kur bus įtraukiami duomenys ataskaitai – reikėtų pasirinkti *Project Estimation Report* (4.12 pav.).



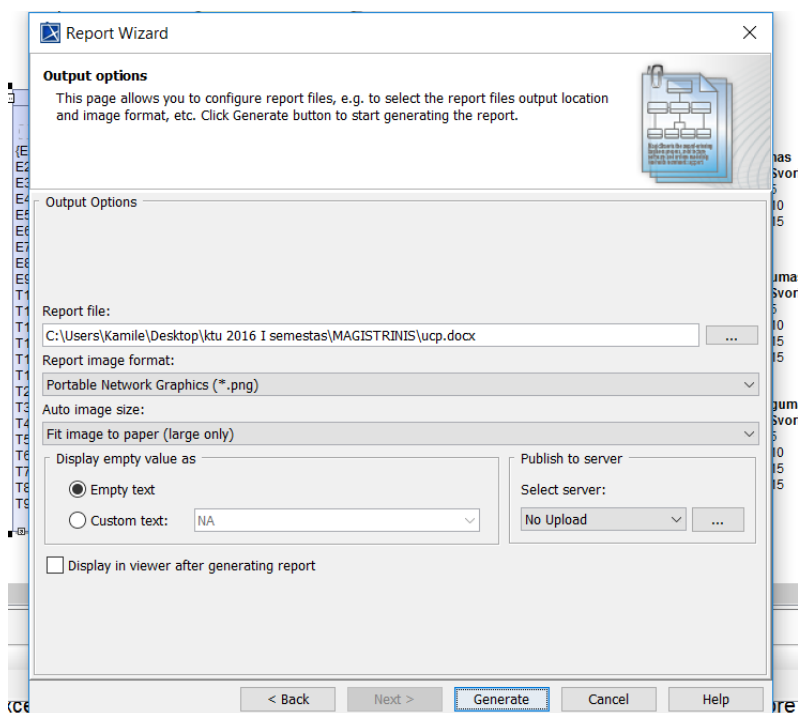
4.12 pav. Ataskaitos generavimo žingsnis nr. 3

Paskutiniame žingsnyje reikia pasirinkti elementus iš kurių bus formuojama ataskaita (4.13 pav.).



4.13 pav. Ataskaitos generavimo žingsnis nr. 4

Bei nurodyti vietą ir failo formatą, kokių bus išsaugota ataskaita (4.14 pav.).



4.14 pav. Ataskaitos generavimo žingsnis nr. 5

Galima generuoti keturias skirtingas ataskaitas. Pagrindinė ataskaita yra bendra metodų ataskaita, išvedanti visų trijų metodų rezultatus, joje yra atvaizduoti tik svarbiausi tarpiniai rezultatai, jų vidurkiai bei galutiniai apimties skaičiavimo rezultatai (4.15 pav.).

Norint pamatyti išsamiau, galima generuoti atskiras ataskaitas, pagal kiekvieną metodą. Jose yra pateikiami visi skaičiavimui naudoti duomenys ir visi tarpiniai rezultatai, būtent pagal tam tikrą metodą (4.16 pav.).

1.1. Panaudojimo atvejų taškai			
1.1.1. Aktorių svoris			
UCP Aktorių svoris	EUCP Aktorių svoris	REUCP Aktorių svoris	Vidurkis
3	3.5	4	3.5

1.1.2. Panaudojimo atvejų svoris			
UCP Panaudojimo atvejų svoris	EUCP Panaudojimo atvejų Aktorių svoris	REUCP Panaudojimo atvejų Aktorių svoris	Vidurkis
15	20	20	18.0

UCP Panaudojimo atvejų taškai	EUCP Panaudojimo atvejų taškai	REUCP Panaudojimo atvejų taškai	Vidurkis
18	23.5	24	21.833333 3333333 2

4.15 pav. Panaudojimo atvejų taškų metodo bendros ataskaitos fragmentas, rodantis aktorių ir panaudojimo atvejų tarpinius rezultatus

1.3. Apimtis			
UCP Apimtis (val.)	EUCP Apimtis (val.)	REUCP (val.)	Vidurkis
381.78	498.43499999999995	474.19199999999999	451.46899999999999

UCP Apimtis (mėn.)	EUCP Apimtis (mėn.)	REUCP Apimtis (mėn.)	Vidurkis
23.86125	31.1521874999999997	29.6369999999999999	28.2168124999999999

UCP Apimtis (d.)	EUCP Apimtis (d.)	REUCP Apimtis (d.)	Apimtis dienomis
1.0846022727272727	1.4160085227272725	1.347136363636363	1.282582386363636

2. Išvados			
Projektui, kuriame dirba 2.0 programuotojai pagal UCP metodą atlikti prireiks 1.0846022727272726 mėnesių, pagal E-UCP metodą: 1.4160085227272725 mėnesių bei pagal Re-UCP metodą 1.3471363636363634 mėnesių.			
(mėnesis: 22.0 darbo dienos)			

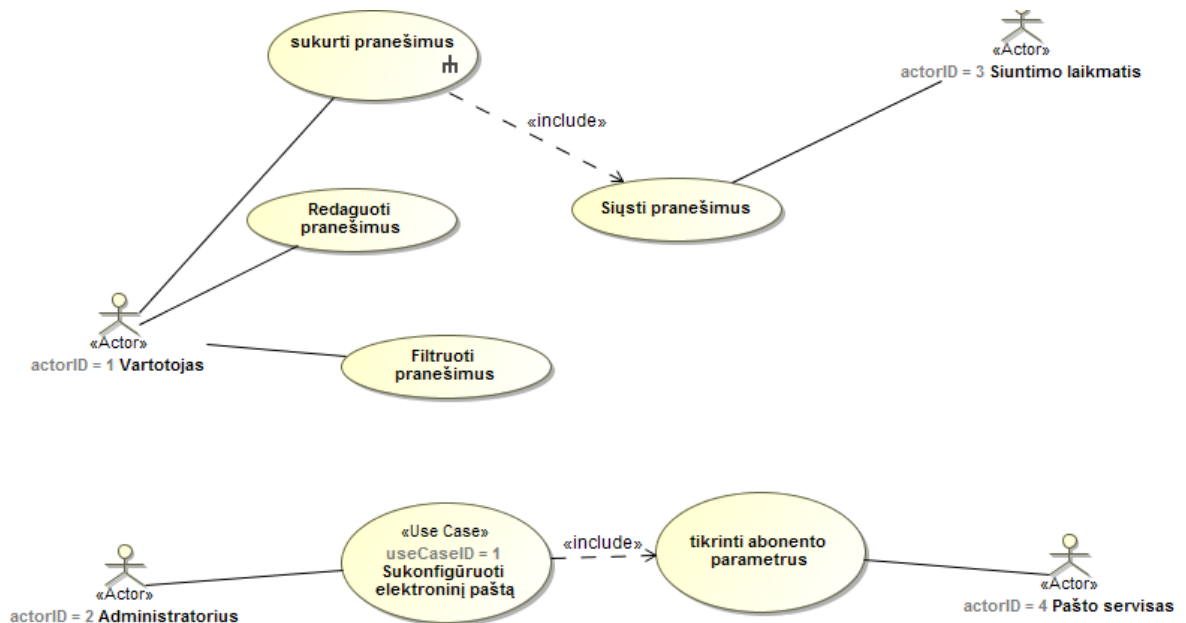
4.16 pav. Panaudojimo atvejų taškų metodo bendros ataskaitos fragmentas, rodantis galutinius rezultatus

5.1.2. Projekto „Pranešimai“ eksperimento planas

Projektas „Pranešimai“ yra skirtas siųsti laiškus – pranešimus iš sistemos į kliento nurodytą elektroninį paštą. Sukurtame sistemos modulyje galima nurodyti vieną ar kelis gavėjus, įrašyti temą, turinį, nustatyti siuntimo laiką ir prioritetą, taip pat pridėti prisegtuką. Projektas apima du modulius: pranešimų siuntimo ir pranešimų konfigūravimo.

Projekto komanda buvo sudaryta iš vieno projektų vadovo ir vieno programuotojo.

Projektas sudarytas iš šešių panaudojimo atvejų ir keturių aktorių, iš kurių du aktoriai yra žmonės, o kiti du sistemos (5.2 pav.). Daugiau informacijos, reikalingos vertinti projektą, naudojant UCP metodus yra pateikta priede (8.6 pav. 8.8 pav. ir 8.7 lentelė – 8.9 lentelė).



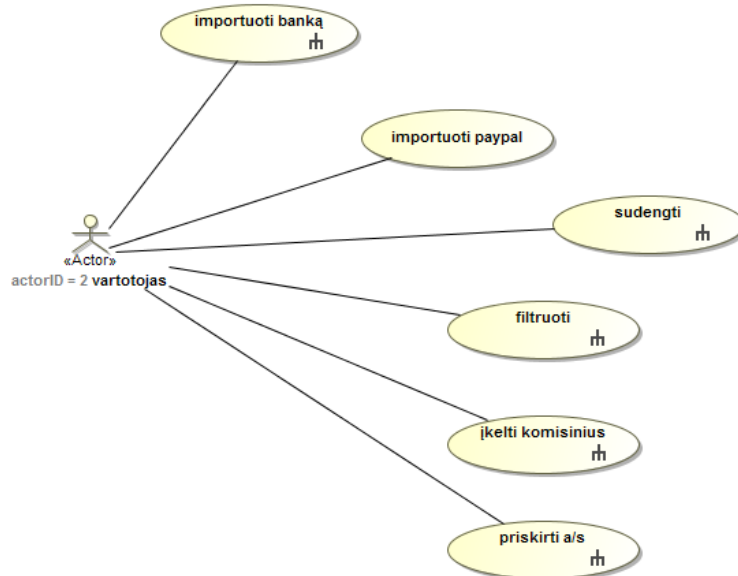
5.2 pav. Projekto „Pranešimai“ panaudojimo atvejų diagrama

5.1.3. Projekto „Banko importas“ eksperimento planas

Projektas „Banko importas“ yra skirtas suimportuoti iš banko į sistemą bankines operacijas (gautinas bei mokėtinas sumas), prie operacijos priskirti pirkėją, prie pirkėjo priskirti atsiskaitomąją sąskaitą, įkelti į sistemą komisinius ir sudengti bankinį įrašą su sistemos modulyje „Atsiskaitymai“ esančia atsiskaitymų operacija.

Projekto komanda buvo sudaryta iš vieno projektų vadovo ir vieno programuotojo.

Projektas yra sudarytas iš vieno aktoriaus ir septynių panaudojimo atvejų (5.3 pav.). Daugiau informacijos, reikalingos vertinti projektą, naudojant UCP metodus yra pateikta priede (8.9 pav. 8.14 pav. ir 8.10 lentelė – 8.15 lentelė).



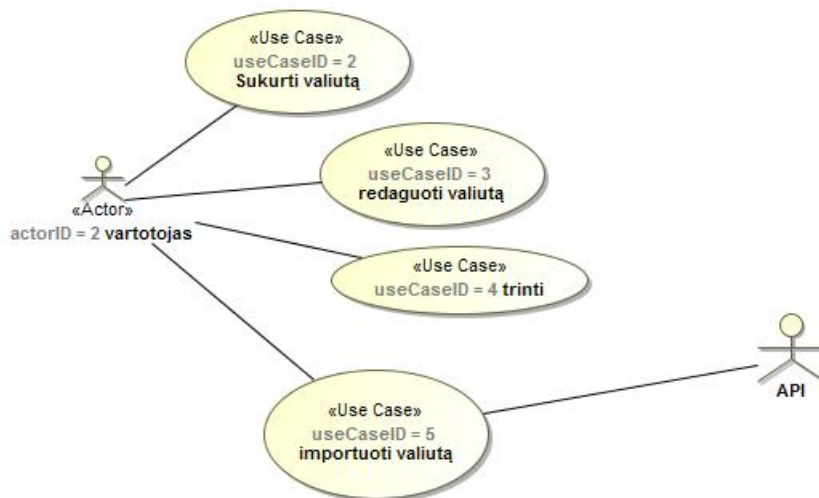
5.3 pav. Projekto „Banko importas“ panaudojimo atvejų diagrama

5.1.4. Projekto „Valiutos“ eksperimento planas

Projektas „Valiutos“ yra skirtas į sistemą susivesti arba iš Lietuvos banko suimportuoti valiutų kursus.

Projekto komanda buvo sudaryta iš vieno programuotojo ir vieno projektų vadovo.

Projekte yra keturi panaudojimo atvejai ir du aktoriai, iš kurių vienas yra žmogus, kitas sistema (5.4 pav.). Daugiau informacijos, reikalingos vertinti projektą, naudojant UCP metodus yra pateikta priede (8.15 pav. 8.17 pav. ir 8.16 lentelė – 8.18 lentelė).



5.4 pav. Projekto „Valiutos“ panaudojimo atvejų diagrama

5.2. Eksperimento rezultatai

Eksperimento rezultatai buvo palyginti su realiais programavimo rezultatais. Įmonė, kurios projektai pasirinkti analizuoti, netaiko jokios moksliskai pagrįstos programos apimties skaičiavimo strategijos.

Pirmiausia pardavimų vadovas parduoda, pasiūlo klientui pirkti projektą, tuomet projektų vadovas išsiaiškina poreikius – poreikiai nėra specifiskai dokumentuojami – svarbios detalės kažkur užrašomos. Tada pardavimų vadybininkas pateikia preliminarį kainą. Tuomet projektų vadovas siunčia užduoties aprašymą užsakovui – užduotis yra aprašoma keletu sakinių, funkcionalumas nėra detalizuojamas. Kaina yra vertinama pagal prieš tai buvusias programavimo užduotis.

Problema su kuria įmonė dažniausiai susiduria yra netiksliai numatytas programavimo darbų laikas, dėl to dažnai už mažesnę kainą tenka dirbti ilgiau. Žinoma, kartais pasitaiko, kad įvertinta kaina yra per didelė, palyginti su sugaištu laiku.

5.2.1. Projekto „Pardavimai“ eksperimento rezultatai

Projekte „Pardavimai“ aktoriai buvo skirtingai klasifikuoti pagal kiekvieną metodą (5.1 lentelė).

5.1 lentelė Projekto „Pardavimai“ aktorių klasifikavimas

Aktorius	UCP metodas	e-UCP metodas	Re-UCP metodas
iVaz API	Paprastas	Labai paprastas	Paprastas
Vartotojas	Sudėtingas	Kritinis	Kritinis

5.2 lentelė Projekto „Pardavimai“ aktorių svoriai

UCP Aktorių svoris	e-UCP Aktorių svoris	Re-UCP Aktorių svoris	Vidurkis
4	4.0	5	4.33

Projekto „Pardavimai“ UCP ir e-UCP metodų svoriai yra vienodi, Re-UCP svoris skiriasi (5.2 lentelė), tai yra dėl skirtingų aktorių svorių reikšmių klasifikavimo.

Projektas susideda iš dvylikos skirtingo sudėtingumo panaudojimo atvejų (5.3 lentelė). Tradicinio UCP metodo čia jau buvo per mažai, todėl, nes vienas iš panaudojimo atvejų turėjo daugiau nei 14 transakcijų. E-UCP ir Re-UCP metodai turi papildomus klasifikavimo kriterijus vertinti labai sudėtingiems panaudojimo atvejams, kas turi įtakos galutiniams skaičiavimams (5.4–5.11 lentelės).

5.3 lentelė Projekto „Pardavimai“ panaudojimo atvejų klasifikavimas

Panaudojimo atvejis	UCP metodas	e-UCP metodas	Re-UCP metodas
Generuoti SF	Paprastas	Paprastas	Paprastas
Importuoti pardavimą	Paprastas	Paprastas	Paprastas
Panaikinti iVaz registraciją	Paprastas	Paprastas	Paprastas
Redaguoti pardavimą	Sudėtingas	Labai sudėtingas	Kritinis
Registruoti iVaz	Paprastas	Paprastas	Paprastas
Sukurti dok. nr.	Paprastas	Paprastas	Paprastas
Sukurti pardavimą	Sudėtingas	Labai sudėtingas	Kritinis
Trinti pardavimą	Paprastas	Paprastas	Paprastas
Įvesti apmokėjimą	Vidutinis	Vidutinis	Vidutinis
Įvesti daugiau nei vieną prekę	Sudėtingas	Sudėtingas	Sudėtingas
Įvesti papildomą informaciją	Paprastas	Paprastas	Paprastas
Įvesti prekę	Sudėtingas	Sudėtingas	Sudėtingas

5.4 lentelė Projekto „Pardavimai“ panaudojimo atvejų svoriai

UCP Panaudojimo atvejų svoris	e-UCP Panaudojimo atvejų svoris	Re-UCP Panaudojimo atvejų svoris	Vidurkis
105	115	115	111.67

5.5 lentelė Projekto „Pardavimai“ panaudojimo atvejų taškai

UCP Panaudojimo atvejų taškai	e-UCP Panaudojimo atvejų taškai	Re-UCP Panaudojimo atvejų taškai	Vidurkis
109	119	120	116

5.6 lentelė Projekto „Pardavimai“ techninių faktorių klasifikavimas

TF kodas / Priskirta reikšmė	UCP metodas	e-UCP metodas	Re-UCP metodas
T1	1	1	1
T2	3	3	3
T3	3	3	3
T4	4	4	4
T5	2	2	2
T6	1	1	1
T7	3	3	3
T8	2	2	2
T9	4	4	4
T10	1	1	1
T11	0	0	0
T12	0	0	0
T13	3	3	3
T14			4

5.7 lentelė Projekto „Pardavimai“ techninių faktorių svoriai

UCP Techninių faktorių svoris	e-UCP Techninių faktorių svoris	Re-UCP Techninių faktorių svoris	Vidurkis
28	28	36	30.67

5.8 lentelė Projekto „Pardavimai“ techninių faktorių sudėtingumas

UCP Techninių faktorių sudėtingumas	e-UCP Techninių faktorių sudėtingumas	Re-UCP Techninių faktorių sudėtingumas	Vidurkis
0.88	0.88	0.96	0.91

5.9 lentelė Projekto „Pardavimai“ aplinkos faktorių klasifikavimas

EF kodas / Priskirta reikšmė	UCP metodas	e-UCP metodas	Re-UCP metodas
E1	4	4	4
E2	5	5	5
E3	4	4	4
E4	3	3	3
E5	4	4	4
E6	3	3	3
E7	0	0	0
E8	0	0	0
E9		1	3

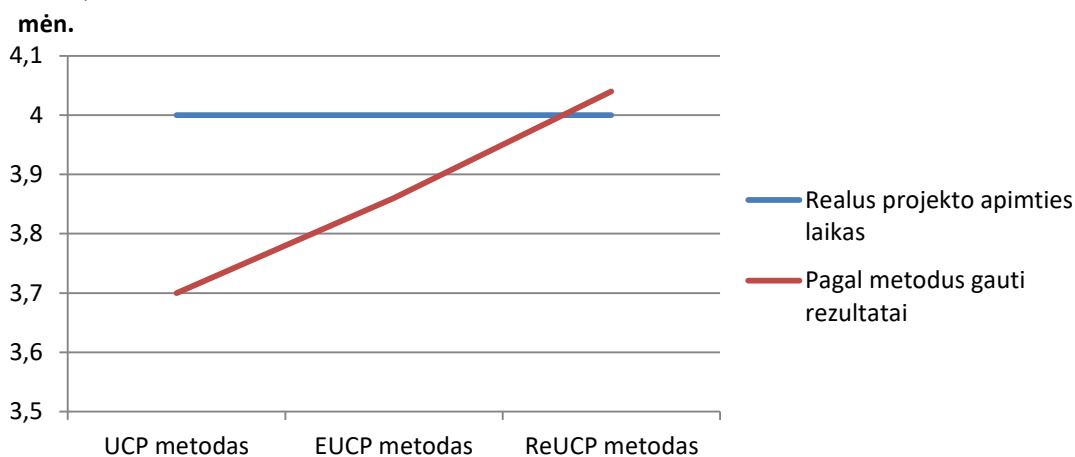
5.10 lentelė Projekto „Pardavimai“ aplinkos faktorių svoriai

UCP aplinkos faktorių svoris	e-UCP aplinkos faktorių svoris	Re-UCP aplinkos faktorių svoris	Vidurkis
24	24	37	25

5.11 lentelė Projekto „Pardavimai“ aplinkos faktorių sudėtingumas

UCP aplinkos faktorių sudėtingumas	e-UCP aplinkos faktorių sudėtingumas	Re-UCP aplinkos faktorių sudėtingumas	Vidurkis
0.68	0.68	0.59	0.65

Projektui, kuriame dirba du programuotojai, pagal UCP metodą atlikti prireiks 3.70 mėnesių, pagal e-UCP metodą: 4.04 mėnesių bei pagal Re-UCP metodą 3.86 mėnesių. Realybėje šis projektas buvo įgyvendintas per 4 mėnesius, todėl arčiausias rezultatas yra e-UCP metodo (5.5 pav., 5.12 lentelė).



5.5 pav. Projekto „Pardavimai“ eksperimento rezultatų grafikas

5.12 lentelė Projekto „Pardavimai“ galutiniai skaičiavimai

Metodas	Apskaičiuota apimtis (mėn.)	Reali apimtis (mėn.)	Procentinis skirtumas
UCP	3.70	4	7.7%
E-UCP	4.04	4	0.99%
Re-UCP	3.86	4	3.56%
vidurkis	3.87	4	4.08%

Nuokrypis tarp mažiausios metodo apimtį 3.70 mėn. ir didžiausios metodo apimtį 4.04 mėn. yra 9%.

5.2.2. Projekto „Pranešimai“ eksperimento rezultatai

Šis projektas susideda iš paprasto, mažiau nei vidutinio ir sudėtingo panaudojimo atvejų (5.13 lentelė).

5.13 lentelė Projekto „Pranešimai“ aktorių klasifikavimas

Aktorius	UCP metodas	e-UCP metodas	Re-UCP metodas
Administratorius	Sudėtingas	Sudėtingas	Sudėtingas
Pašto servisas	Paprastas	Mažiau nei vidutinis	Paprastas
Siuntimo laikmatis	Paprastas	Paprastas	Paprastas

Vartotojas	Sudėtingas	Sudėtingas	Sudėtingas
------------	------------	------------	------------

5.14 lentelė Projekto „Pranešimai“ aktorių svoriai

UCP Aktorių svoris	e–UCP Aktorių svoris	Re–UCP Aktorių svoris	Vidurkis
8	8.5	8	8.17

UCP ir REUCP metodų aktorių svoriai yra vienodi, nes buvo pasirinktas toks pats aktorių vertinimas, metodo EUCP aktorių svoris skiriasi dėl to, kad vienas aktorius buvo įvertintas ne „Paprastas“, o „Mažiau nei vidutinis“ (5.14 lentelė).

Pagal panaudojimo atvejų transakcijų skaičių kiekvienas metodo panaudojimo atvejis yra įvertintas kaip paprastas arba sudėtingas (5.15 lentelė).

5.15 lentelė Projekto „Pranešimai“ panaudojimo atvejų sudėtingumo klasifikacija

Panaudojimo atvejis	UCP metodas	e–UCP metodas	Re–UCP metodas
Filtruoti pranešimus	Vidutinis	Vidutinis	Vidutinis
Redaguoti pranešimus	Sudėtingas	Sudėtingas	Sudėtingas
Siųsti pranešimus	Paprastas	Paprastas	Paprastas
Sukonfigūruoti elektroninį paštą	Vidutinis	Vidutinis	Vidutinis
Registruoti iVaz	Paprastas	Paprastas	Paprastas
Sukurti pranešimus	Sudėtingas	Sudėtingas	Sudėtingas
Tikrinti abonento parametrus	Paprastas	Paprastas	Paprastas

Visų panaudojimo atvejų svoris yra vienodas. Naudojant metodą e–UCP panaudojimo atvejų taškai skiriasi, dėl EUCP metodo aktorių svorio skaičiavimo (5.16–5.17 lentelės).

5.16 lentelė Projekto „Pranešimai“ panaudojimo atvejų svoris

UCP Panaudojimo atvejų svoris	e–UCP Panaudojimo atvejų svoris	Re–UCP Panaudojimo atvejų svoris	Vidurkis
55	55	55	55.0

5.17 lentelė Projekto „Pranešimai“ panaudojimo atvejų sudėtingumas

UCP Panaudojimo atvejų taškai	e–UCP Panaudojimo atvejų taškai	Re–UCP Panaudojimo atvejų taškai	Vidurkis
63	63.5	63	63.16

Techninių faktorių rezultatas UCP ir e–UCP metodo yra vienodas, skaičiavimai yra identiški, Re–UCP metodo rezultatas skiriasi dėl prisidėjusio techninio faktoriaus: „T14 Mastelis“ (5.18 – 5.20 lentelės).

5.18 lentelė Projekto „Pranešimai“ techninių faktorių klasifikacija

TF kodas / Priskirta reikšmė	UCP metodas	e–UCP metodas	Re–UCP metodas
T1	2	2	2
T2	3	3	3
T3	3	3	3
T4	1	1	1
T5	3	3	3

T6	3	3	3
T7	5	5	5
T8	2	2	2
T9	2	2	2
T10	3	3	3
T11	0	0	0
T12	1	1	1
T13	1	1	1
T14			2

5.19 lentelė Projekto „Pranešimai“ techninių faktorių svoris

UCP Techninių faktorių svoris	e–UCP Techninių faktorių svoris	Re–UCP Techninių faktorių svoris	Vidurkis
29	29	31	29.66

5.20 lentelė Projekto „Pranešimai“ techninių faktorių sudėtingumas

UCP Techninių faktorių sudėtingumas	e–UCP Techninių faktorių sudėtingumas	Re–UCP Techninių faktorių sudėtingumas	Vidurkis
0.89	0.89	0.91	0.89

Aplinkos faktorių rezultatas UCP ir e–UCP metodo yra vienodas, skaičiavimai yra identiški, Re–UCP metodo rezultatas skiriasi dėl prisidėjusio aplinkos faktoriaus: „E9 Projekto metodologija“ (5.21 – 5.23 lentelės).

5.21 lentelė Projekto „Pranešimai“ aplinkos faktorių klasifikavimas

EF kodas / Priskirta reikšmė	UCP metodas	e–UCP metodas	Re–UCP metodas
E1	4	4	4
E2	5	5	5
E3	4	4	4
E4	3	3	3
E5	4	4	4
E6	4	4	4
E7	0	0	0
E8	0	0	0
E9			2

5.22 lentelė Projekto „Pranešimai“ aplinkos faktorių svoris

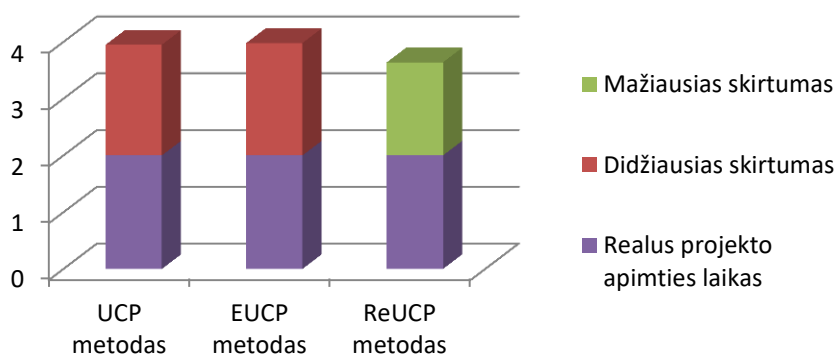
UCP Aplinkos faktorių svoris	e–UCP Aplinkos faktorių svoris	Re–UCP Aplinkos faktorių svoris	Vidurkis
26	26	28	26

5.23 lentelė Projekto „Pranešimai“ aplinkos faktorių sudėtingumas

UCP Aplinkos faktorių sudėtingumas	e–UCP Aplinkos faktorių sudėtingumas	Re–UCP Aplinkos faktorių sudėtingumas	Vidurkis
0.62	0.62	0.56	0.6

Projektui, kuriame dirba vienas programuotojas, pagal UCP metodą atlikti prireiks 3,95 mėnesių, pagal e–UCP metodą: 3.98 mėnesių bei pagal Re–UCP metodą 3.64 mėnesių. Realybėje

šis projektas buvo įgyvendintas per 2 mėnesius, todėl arčiausias rezultatas yra Re-UCP metodo (5.6 pav., 5.24 lentelė).



5.6 pav. Projekto „Pranešimai“ eksperimento rezultatai

5.24 lentelė Galutiniai projekto „Pranešimai“ rezultatai

Metodas	Apskaičiuota apimtis (mėn.)	Reali apimtis (mėn.)	Procentinis skirtumas
UCP	3,95	2	65.55%
E-UCP	3,98	2	66.22%
Re-UCP	3,64	2	58.16%
vidurkis	3,86	2	63,64

Nuokrypis tarp mažiausios metodo apimties 58.16 ir didžiausios metodo apimties 66.22 yra 13%.

5.2.3. Projekto „Banko importas“ eksperimento rezultatai

Šiame projekte yra tik vienas aktorius, kuris pagal visus metodus klasifikuojamas kaip sudėtingas (5.25 lentelė).

5.25 lentelė Projekto „Banko importas“ aktorių klasifikacija

Aktorius	UCP metodas	e-UCP metodas	Re-UCP metodas
Vartotojas	Sudėtingas	Sudėtingas	Sudėtingas

UCP ir REUCP aktorių svoriai yra vienodi, tačiau metodas EUCP turi daugiau aktorių klasifikacijų, šio metodo sudėtingo aktoriaus svoris yra 2.5 – tai gali būti vienas iš kriterijų galutiniam rezultatui (5.26 lentelė).

5.26 lentelė Projekto „Banko importas“ aktorių svoriai

UCP Aktorių svoris	e-UCP Aktorių svoris	Re-UCP Aktorių svoris	Vidurkis
3	2.5	3	2.83

Pagal panaudojimo atvejų transakcijų skaičių kiekvienas metodo panaudojimo atvejis yra paprastas arba sudėtingas (5.27 lentelė).

5.27 lentelė Projekto „Banko importas“ panaudojimo atvejų klasifikacija

Panaudojimo atvejis	UCP metodas	e-UCP metodas	Re-UCP metodas
Filtruoti	Paprastas	Paprastas	Paprastas
Importuoti banką	Paprastas	Paprastas	Paprastas
Importuoti PayPal	Paprastas	Paprastas	Paprastas
Priskirti a/s	Vidutinis	Vidutinis	Vidutinis
Sudengti	Vidutinis	Vidutinis	Vidutinis

Įkelti komisinius	Paprastas	Paprastas	Paprastas
-------------------	-----------	-----------	-----------

Kadangi projekto metu pagal visus metodo atvejus panaudojimo atvejai yra įvertinti vienodai: parasti arba vidutiniai, tai ir aktorių svoris visų trijų metodų yra vienodas (5.28 lentelė).

5.28 lentelė Projekto „Banko importas“ panaudojimo atvejų svoriai

UCP Panaudojimo atvejų svoris	e–UCP Panaudojimo atvejų Aktorių svoris	Re–UCP Panaudojimo atvejų Aktorių svoris	Vidurkis
40	40	40	40

Tačiau panaudojimo atvejų rezultatas skiriasi dėl e–UCP metodo aktorių svorio skaičiavimų skirtumo (5.29 lentelė).

5.29 lentelė Projekto „Banko importas“ panaudojimo atvejų sudėtingumas

UCP Panaudojimo atvejų taškai	e–UCP Panaudojimo atvejų taškai	Re–UCP Panaudojimo atvejų taškai	Vidurkis
43	42.5	43	42.83

Techninių faktorių rezultatas UCP ir e–UCP metodo yra vienodas, skaičiavimai yra identiški, Re–UCP metodo rezultatas skiriasi dėl prisidėjusio techninio faktoriaus: „T14 Mastelis“ (5.30 – 5.32 lentelė).

5.30 lentelė Projekto „Banko importas“ techninių faktorių klasifikavimas

TF kodas / Priskirta reikšmė	UCP metodas	e–UCP metodas	Re–UCP metodas
T1	0	0	0
T2	3	3	3
T3	3	3	3
T4	1	1	1
T5	3	3	3
T6	3	3	3
T7	4	4	4
T8	3	3	3
T9	3	3	3
T10	0	0	0
T11	0	0	0
T12	0	0	0
T13	0	0	0
T14			2

5.31 lentelė Projekto „Banko importas“ techninių faktorių svoriai

UCP Techninių faktorių svoris	e–UCP Techninių faktorių svoris	Re–UCP Techninių faktorių svoris	Vidurkis
22.5	22.5	24.5	23.17

5.32 lentelė Projekto „Banko importas“ techninių faktorių sudėtingumas

UCP Techninių faktorių sudėtingumas	e–UCP Techninių faktorių sudėtingumas	Re–UCP Techninių faktorių sudėtingumas	Vidurkis
0.825	0.825	0.845	0.83

Aplinkos faktorių rezultatas UCP ir e–UCP metodo yra vienodas, skaičiavimai yra identiški, Re–UCP metodo rezultatas skiriasi dėl prisidėjusio aplinkos faktoriaus: „E9 Projekto metodologija“ (5.33 – 5.35 lentelė).

5.33 lentelė Projekto „Banko importas“ aplinkos faktorių klasifikavimas

EF kodas / Priskirta reikšmė	UCP metodas	e–UCP metodas	Re–UCP metodas
E1	5	5	5
E2	4	4	4
E3	4	4	4
E4	3	3	3
E5	3	3	3
E6	3	3	3
E7	0	0	0
E8	0	0	0
E9			2

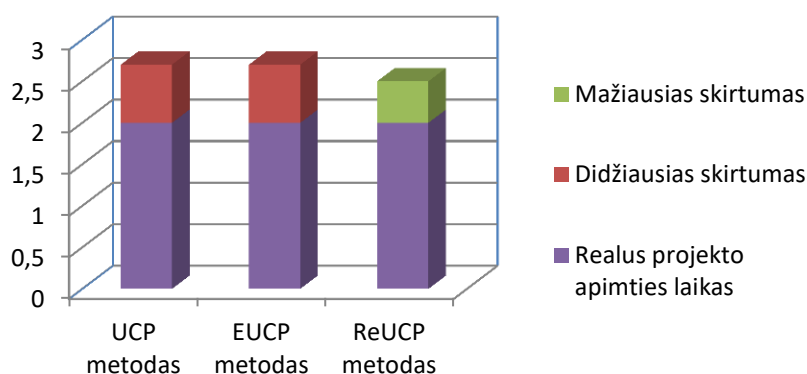
5.34 lentelė Projekto „Banko importas“ techninių faktorių svoriai

UCP Aplinkos faktorių svoris	e–UCP Aplinkos faktorių svoris	Re–UCP Aplinkos faktorių svoris	Vidurkis
24.0	24.0	26.0	24.66

5.35 lentelė Projekto „Banko importas“ techninių faktorių sudėtingumas

UCP Aplinkos faktorių sudėtingumas	e–UCP Aplinkos faktorių sudėtingumas	Re–UCP Aplinkos faktorių sudėtingumas	Vidurkis
0.68	0.68	0.62	0.66

Projektui, kuriame dirba 1 programuotojas, pagal UCP metodą atlikti prireiks 2.7 mėnesių, pagal e–UCP metodą: 2.7 mėnesių bei pagal Re–UCP metodą 2.5 mėnesių. Realybėje šis projektas buvo įgyvendintas per 2 mėnesius, todėl arčiausias rezultatas yra Re–UCP metodo (5.7 pav., 5.36 lentelė.).



5.7 pav. Projekto „Banko importas“ eksperimento rezultatai

5.36 lentelė Galutiniai projekto „Banko importas“ rezultatai

Metodas	Apskaičiuota apimtis (mėn.)	Realus apimtis (mėn.)	Procentinis skirtumas
UCP	2,7	2	29.79%
E–UCP	2,7	2	29.79%

Re-UCP	2,5	2	22.22%
	2.63	2	27.27

Nuokrypis tarp mažiausios metodo apimties 22.22 ir didžiausios metodo apimties 29.97 yra 29%.

5.2.4. Projekto „Valiutos“ eksperimento rezultatai

Šiame projekte, pagal visus tris metodus, aktoriai buvo klasifikuojami į paprastą ir sudėtingą (5.37 lentelė).

5.37 lentelė Projekto „Valiutos“ aktorių klasifikavimas

Aktorius	UCP metodas	e-UCP metodas	Re-UCP metodas
API	Paprastas	Paprastas	Paprastas
Vartotojas	Sudėtingas	Sudėtingas	Sudėtingas

UCP ir Re-UCP aktorių svoriai yra vienodi, tačiau kadangi metodas e-UCP turi daugiau aktorių klasifikacijų, šio metodo svoris yra 3.5 – tai gali būti vienas iš kriterijų galutiniam rezultatui (5.38 lentelė).

5.38 lentelė Projekto „Valiutos“ aktorių svoriai

UCP Aktorių svoris	e-UCP Aktorių svoris	Re-UCP Aktorių svoris	Vidurkis
4	3.5	4	3.83

Pagal panaudojimo atvejų transakcijų skaičių kiekvieno metodo panaudojimo atvejų sudėtingumas yra klasifikuojamas vienodai (5.39 lentelė).

5.39 lentelė Projekto „Valiutos“ panaudojimo atvejų sudėtingumo klasifikavimas

Panaudojimo atvejis	UCP metodas	e-UCP metodas	Re-UCP metodas
Importuoti valiutą	Paprastas	Paprastas	Paprastas
Redaguoti valiutą	Paprastas	Paprastas	Paprastas
Sukurti valiutą	Paprastas	Paprastas	Paprastas
Trinti	Paprastas	Paprastas	Paprastas

Panaudojimo atvejų svoris pagal visus metodus yra vienodas. Skiriasi tik panaudojimo atvejų taškai, dėl kitokio EUCP metodo aktorių skaičiavimo (5.40 – 5.41 lentelės).

5.40 lentelė Projekto „Valiutos“ panaudojimo atvejų svoris

UCP Panaudojimo atvejų svoris	e-UCP Panaudojimo atvejų Aktorių svoris	Re-UCP Panaudojimo atvejų Aktorių svoris	Vidurkis
20	20	20	20

5.41 lentelė Projekto „Valiutos“ panaudojimo atvejų taškai

UCP Panaudojimo atvejų taškai	e-UCP Panaudojimo atvejų taškai	Re-UCP Panaudojimo atvejų taškai	Vidurkis
24	23.5	24	23.83

Techninių faktorių rezultatas UCP ir EUCP metodo yra vienodas, skaičiavimai yra identiški, REUCP metodo rezultatas skiriasi dėl prisidėjusio techninio faktoriaus: „T14 Mastelis“ (5.42 – 5.44 lentelės).

5.42 lentelė Projekto „Valiutos“ techninių faktorių reikšmės

TF kodas / Priskirta reikšmė	UCP metodas	e-UCP metodas	Re-UCP metodas
T1	0	0	0
T2	3	3	3
T3	3	3	3
T4	1	1	1
T5	2	2	2
T6	0	0	0
T7	4	4	4
T8	1	1	1
T9	2	2	2
T10	0	0	0
T11	0	0	0
T12	0	0	0
T13	0	0	0
T14		3	

5.43 lentelė Projekto „Valiutos“ techninių faktorių svoris

UCP Techninių faktorių svoris	e-UCP Techninių faktorių svoris	Re-UCP Techninių faktorių svoris	Vidurkis
15.0	15.0	21.0	17.0

5.44 lentelė Projekto „Valiutos“ techninių faktorių sudėtingumas

UCP Techninių faktorių sudėtingumas	e-UCP Techninių faktorių sudėtingumas	Re-UCP Techninių faktorių sudėtingumas	Vidurkis
0.75	0.75	0.81	0.77

Aplinkos faktorių rezultatas UCP ir e-UCP metodo yra vienodas, skaičiavimai yra identiški, Re-UCP metodo rezultatas skiriasi dėl prisidėjusio aplinkos faktoriaus: „E9 Projekto metodologija“ (5.45 – 5.47 lentelės).

5.45 lentelė Projekto „Valiutos“ aplinkos faktorių reikšmės

EF kodas / Priskirta reikšmė	UCP metodas	e-UCP metodas	Re-UCP metodas
E1	4	4	4
E2	5	5	5
E3	5	5	5
E4	4	4	4
E5	3	3	3
E6	4	4	4
E7	0	0	0
E8	0	0	0
E9		1	

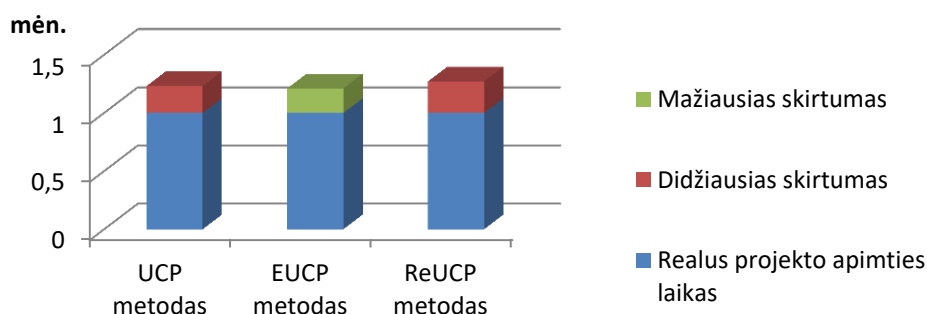
5.46 lentelė Projekto „Valiutos“ aplinkos faktorių svoriai

UCP Aplinkos faktorių svoris	e-UCP Aplinkos faktorių svoris	Re-UCP Aplinkos faktorių svoris	Vidurkis
26.5	26.5	27.5	26.83

5.47 lentelė Projekto „Valiutos“ aplinkos faktorių sudėtingumas

UCP Aplinkos faktorių sudėtingumas	e-UCP Aplinkos faktorių sudėtingumas	Re-UCP Aplinkos faktorių sudėtingumas	Vidurkis
0.605	0.605	0.575	0.595

Projektui, kuriame dirba 1 programuotojas, pagal UCP metodą atlikti prireiks 1.23 mėnesių, pagal e-UCP metodą: 1.21 mėnesių bei pagal Re-UCP metodą 1.27 mėnesių. Realybėje šis projektas buvo įgyvendintas per 1 mėnesį, todėl arčiausias rezultatas yra e-UCP metodo (5.8 pav.).



5.8 pav. Projekto „Valiutos“ eksperimento rezultatų grafikas

5.48 lentelė Projekto „Valiutos“ galutiniai skaičiavimai

Metodas	Apskaičiuota apimtis (mėn.)	Realus apimtis (mėn.)	Procentinis skirtumas
UCP	1.23	1	20%
E-UCP	1.21	1	19%
Re-UCP	1.27	1	24%
vidurkis	1.24	1	22%

Nuokrypis tarp mažiausios metodo apimties 1.23 ir didžiausios metodo apimties 1.27 yra 2%.

5.3. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas

Eksperimentas buvo atliktas naudojant keturis skirtingos apimties projektus su skirtingomis panaudojimo atvejų diagramomis. Dviejų projektų reali atlikimo trukmė buvo 2 mėnesiai, kiti – 8 mėn. (dirbant dviems programuotojams 4 mėn.) ir 1 mėn.

5.49 lentelė Rezultatų palyginimas pagal metodus

Projektas	Realus projekto apimtis (mėn.)	Programuotojų kiekis	Apskaičiuotos projekto apimtis (mėn.)			Procentinio skirtumo vidurkis
			UCP	EUCP	REUCP	
Pardavimai	4	2	3.70	4.04	3.86	4%
Pranešimai	2	1	3.95	3.98	3.64	63%
Valiutos	1	1	1.23	1.24	1.27	22%
Banko importas	2	1	2.7	2.7	2.5	27%

Atlikus eksperimentą matome, kad procentinis skirtumas yra nuo 4% iki 63%. Mažiausias procentinis skirtumas buvo ilgiausios trukmės projekto, šio projekto rezultatai yra artimiausi realiai įgyvendinto projekto trukmei, todėl galima teigti, kad metodai labiausiai tinkami naudoti stambesniems projektams (5.49 lentelė).

Didžiausiais procentinis skirtumas buvo 63 proc. projekte „Pranešimai“, tačiau vienareikšmiškai vertinti, kad metodas suveikė blogai negalima, nes galutinius rezultatus lemia daug veiksnių. Galėjo būti blogai įvertinti aktorių ar panaudojimo atvejų sudėtingumai, dėl analitiko patirties trūkumo ar tam tikrų technologinių sprendimų nežinojimo, pvz.: vietoj 3 transakcijų buvo suskaičiuotos 4, dėl to aktorius buvo įvertintas kaip vidutinis, o turėjo būti paprastas. Taip pat didelę įtaką skaičiavimams daro techniniai ir aplinkos faktoriai, jų vertinimui nėra aiškių rekomendacijų, tai labiau paremta asmenine patirtimi, pvz., kartais tam tikrą faktorių įvertinti 3 ar 4 gali būti ganėtinai sudėtinga, tačiau tas reikšmės pokytis prie galutinių rezultatų prisideda.

Kitų dviejų projektų rezultatas buvo tarp 22% ir 27% , tai yra neblogas rezultatas, jei visi aktoriai ir panaudojimo atvejai įvertinti gerai, procentinį skirtumą gali nulemti techniniai ir aplinkos faktoriai, todėl pakeitus jų variacijas galima gauti tikslesnius rezultatus.

Taip pat galutiniai rezultatai gali skirtis ir dėl netinkamai nubraižytos panaudojimo atvejų diagramos.

5.50 lentelė Rezultatų palyginimas, pagal metodus

Projektas	Reali projekto apimtis (mėn.)	Program uotojų kiekis	Apskaičiuotos projekto apimtis (mėn.)			Procentinis skirtumas tarp metodo žemiausio ir aukščiausio rezultato
			UCP	EUCP	REUC P	
Pardavimai	4	2	3.70	4.04	3.86	9%
Pranešimai	2	1	3.95	3.98	3.64	13%
Valiutos	1	1	1.23	1.24	1.27	2%
Banko importas	2	1	2.7	2.7	2.5	29%

Remiantis gautais rezultatais matome (5.50 lentelė), kad procentinis skirtumas tarp metodų nėra labai didelis, iš kitos pusės, skirtumas turėtų būti, nebent metodai būtų naudojami vertinat faktorius identiškai. Metodas e-UCP turi plačiausią aktorių ir panaudojimo atvejų klasifikaciją, Re-UCP metodo klasifikavimas taip pat yra praplėstas, palyginus su standartiniu UCP metodu, tai ir įtakoja galutinius rezultatus. Tiksliausias metodo naudojimo pasirinkimas turėtų būti nustatomas remiantis analitiko patirtimi. Metodų rezultatai gali priklausyti tiek nuo projekto apimties, tiek nuo panaudojimo atvejų, aktorių ir faktorių įvertinimo. Vienareikšmiškai teigti, kuris metodas yra teisingas, negalima.

5.4. Sprendimo taikymo rekomendacijos

1. Panaudojimo atvejų taškų metodas turėtų būti naudojamas projekto komandoje, kurioje dalyvauja vienas analitikas, jei komanda yra sudaryta iš daugiau nei vieno analitiko, jie turėtų dirbti panašiu principu, kad panaudojimo atvejų diagramos būtų kuo panašesnės, tikslesnės ir dėl to nedarytų įtakos skaičiavimams.
2. Analitikai ar projektų vadovai, ruošiantys PA diagramas, turėtų būti pažengę UML kalbos vartotojai, mokantys teisingai ruošti panaudojimo atvejų diagramas.
3. Nebūtina išsirinkti vieno metodo, kuris bus naudojamas. Atsižvelgiant į situaciją, vieną kartą tikslesnis gali būti vienas metodas, kitą kartą kitas. Tiksliausias metodo pasirinkimas, atsižvelgiant į projekto apimtį ir tam tikrus veiksnius, turėtų būti paremtas patirtimi, remiantis ankstesnių projektų vertinimais ir realia trukme. Ateityje vertinant metodus būtų galima tikėtis gauti kuo tikslesnį rezultatą.
4. Techninių ir aplinkos faktorių vertinimas nėra aiškiai aprašytas, todėl reikėtų stengtis kuo daugiau įsigilinti į faktoriaus poveikį kuriamai sistemai. Faktorių vertinimo įgūdžiai turėtų stiprėti su metodo naudojimo patirtimi. Todėl, kuo daugiau projektų analitikas vertins, tuo geresni bus ateities rezultatai. Kartais yra sunku atskirti ribą tarp reikšmės 3 ir 4, tačiau galutiniams skaičiavimų rezultatams tai turi įtakos.

5. Jei yra galimybė, prieš pradėdant naudoti metodus, būtų naudinga metodus ištestuoti, susikuriant panaudojimo atvejų diagramas, jau atiliktiems 3 – 5 skirtingos apimties ar sudėtingumo projektams. Dėl to gali būti lengviau suprasti kokius faktorius naudoti ar kokį metodą parinkti.
6. Išplėstasis e–UCP metodas galėtų būti naudojamas tik didesniems projektams, su daug skirtingų aktorių ir panaudojimo atvejų, nes palyginus gautus rezultatus naudojant e–UCP metodą mažos apimties projektui, rezultatai nedaug skyrėsi. Vertinant mažos apimties projektą nėra tikslo naudoti tiek aktorių klasifikavimo.
7. Žmogus ne visada yra linkęs vadovautis kompiuterio pateiktais skaičiavimais, šioje vietoje suveikia nepasitikėjimo faktorius. Vertinant projektą trimis metodais, skaičiavimų rezultatų pasitikėjimas turėtų būti didesnis, pvz., vartotojas gavo du labai panašius rezultatus, ir vieną kitokį, jam bus lengviau apsispręsti, kurį rezultatą naudoti galutiniam projekto vertinimui.
8. Kadangi kiekvienas metodas yra kažkuo papildytas, pvz., labiau klasifikuoti aktoriai ar panaudojimo atvejai, pridėti techniniai ir aplinkos faktoriai, vartotojas gali aiškiai matyti, kiek ir kokie veiksniai turėjo įtakos galutinei projekto apimčiai. Tai gali būti naudinga ne tik projekto apimčiai įvertinti, bet ir nustatyti kokius kriterijus kuriant programinę įrangą reikia keisti.

6. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

1. Atlikus UCP, e– UCP ir Re– UCP metodų analizę, nustatyta, kad nėra teisingo standarto kaip reikėtų naudoti panaudojimo atvejų taškų skaičiavimo metodus.
2. Išanalizavus panaudojimo atvejų taškų metodą, pastebėta, kad panaudojimo atvejų ir aktorių svorio vertinimas vis dar nėra aiškus, todėl vertinimas turėtų būti paremtas patirtimi.
3. Atlikus išsamią mokslininkų tyrimų analizę, buvo rastos kelios skirtingos projekto vertinimo variacijos, panaudojimo atvejų taškų metodo pagrindinis pranašumas, kad jis gali būti atliekamas pradinėje projekto stadijoje.
4. Atliktas eksperimentas su trimis metodų variacijomis ir penkiais skirtingais projektais, atskleidė, kad UCP metodo mažiausias procentinis skirtumas buvo 8 proc., didžiausias 66 proc., e– UCP metodo mažiausias procentinis skirtumas buvo 1 proc., didžiausias 66 proc., Re– UCP metodo mažiausias 4 proc. didžiausias 58 proc. Procentinis skirtumas gali skirtis dėl skirtingos PA, aktorių, techninių ir aplinkos faktorių klasifikacijos.
5. Remiantis atliktais eksperimentais, su pasirinktais projektais, galima teigti, kad mažesnės apimties projektams reikėtų vengti e– UCP metodo, nes jo vertinimui reikia daugiau pastangų, o galutinis rezultatas nedaug skiriasi nuo kitų metodų, o didesnės apimties projektams reiktų jį naudoti, nes rezultatas naudojant šį metodą buvo gautas tiksliausias.
6. Eksperimento metu pastebėta, kad naudojant tris PA taškų skaičiavimo variacijas kartu yra lengviau suprasti kaip ir koks veiksnys veikia projekto apimtį.
7. Atliktų eksperimentų patirtis leidžia tikėtis, kad PA taškų skaičiavimo variacijas galima pritaikyti projekto planavimo srityse kuo tikslesniam projekto apimties nustatymui, taip pat tai gali prisidėti prie projekto planavimo – paruošus preliminarinius panaudojimo atvejų diagramas ir pritaikius metodus galima matyti kokį jie poveikį turės galutinei projekto apimčiai.

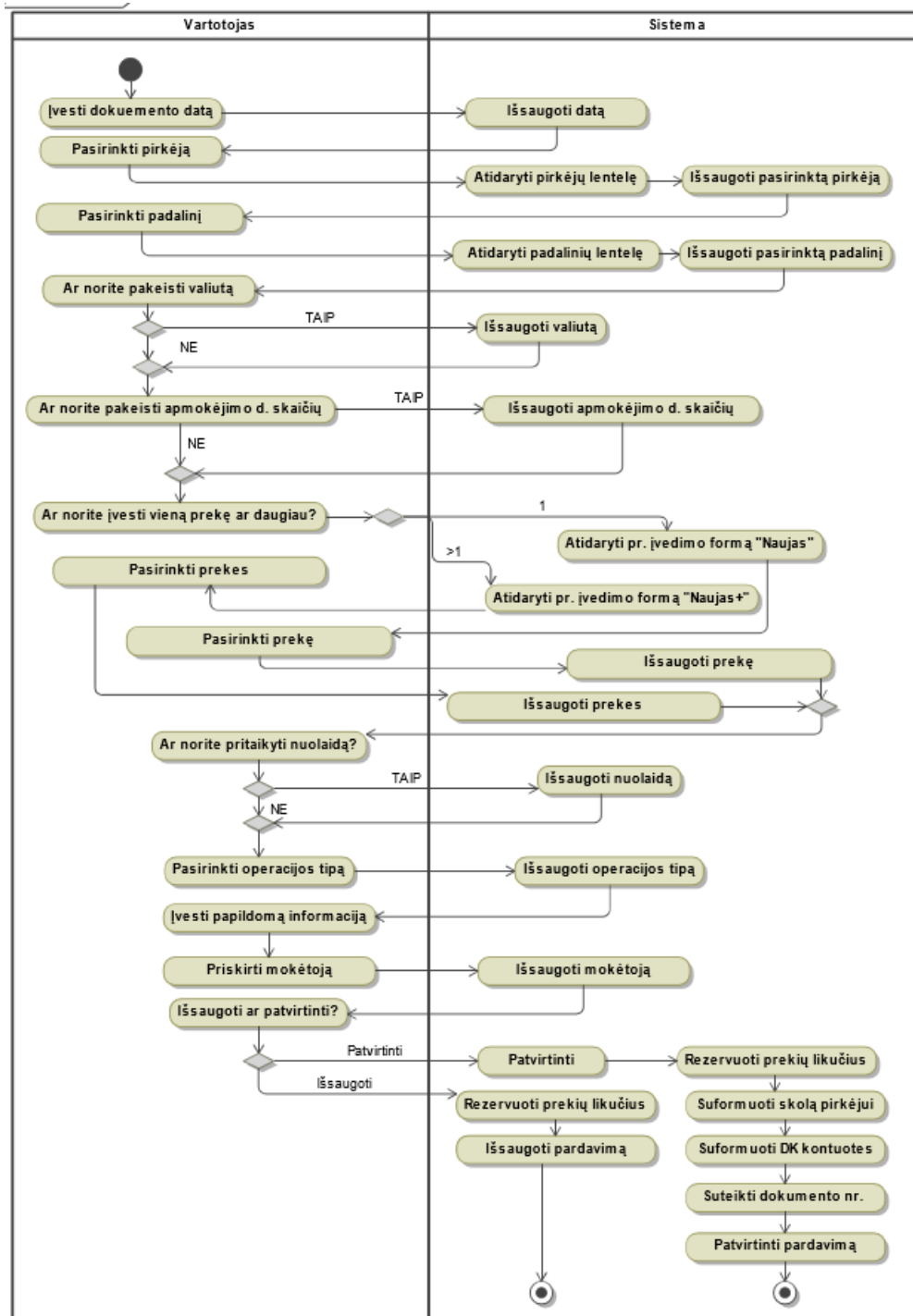
7. LITERATŪRA

- [1] Shinji Kusumoto, Fumikazu Matukawa, Katsuro Inoue „Estimating Effort by Use Case Points: Method, Tool and Case Study“, Graduate School of Information Science and Technology, Osaka University, 2004.
- [2] Farhad Soleimanian Gharehchopogh, Isa Maleki „Analysis of Use Case Points Models for Software Cost Estimation“, Hacettepe University, Islamic Azad University, 2014.
- [3] M. Ochodek ↑, J. Nawrocki, K. Kwarciak „Simplifying effort estimation based on Use Case Points“, Information and Software Technology nr. 53, pp. 200–213, 2011.
- [4] M. Ochodek ↑, B. Alchimowicz, J. Jurkiewicz, J. Nawrocki „Improving the reliability of transaction identification in use cases“, „, Information and Software Technology nr. 53, pp. 885 – 897, 2011.
- [5] Mudasir Manzoor Kirmani, Abdul Wahid „Use Case Point Method of Software Effort Estimation: A Review“, School of CS & IT, Manuu, Hyderabad, India., pp. 43–48, 2015.
- [6] Jacobson, Ivar et al., „Object–Oriented Software Engineering. A Use Case Driven Approach“, revised printing, Addison–Wesley 1993.
- [7] John Erik Hansen, „Use Case Point Estimation – Estimate Your Project By Looking At Your Use Cases“, 2012. Prieiga per internetą: <<http://www.allaboutrequirements.com/2012/12/use-case-point-estimation-estimate-your-project-by-looking-at-your-use-cases.html> > [kreiptasi 2015-12-15]
- [8] Mike Cohn „Estimating With Use Case Points“, 2012. Prieiga per internetą: <http://www.cs.cmu.edu/~jhm/Readings/Cohn%20-%20Estimating%20with%20Use%20Case%20Points_v2%202012-24-50-761.pdf> [kreiptasi 2015-10-12]
- [9] „MagicDraw Report Wizard User Guide“, 2015. Prieiga per internetą: <<http://www.nomagic.com/files/manuals/MagicDraw%20ReportWizard%20UserGuide.pdf>> [kreiptasi 2016-01-19]
- [10] „Free Use Case Point Estimation Template In Excel Format“, 2012. Prieiga per internetą: <<http://www.allaboutrequirements.com/2012/12/free-use-case-point-estimation-template-in-excel-format.html>> [kreiptasi 2016-03-01]
- [11] Abdollah Zawari, „Project estimation with Use Case Points using Enterprise Architect (EA)“
- [12] Mudasir Manzoor Kirmani , Abdul Wahid , „Revised Use Case Point (Re–UCP) Model for Software Effort Estimation“, (IJACSA) International Journal of Advanced Computer Science and Applications, Research Scholar, School of CS & IT, Dean and Head School of CS & IT, Maulana Azad National Urdu University Hyderabad, India, 2015. Prieiga per internetą: <http://thesai.org/Downloads/Volume6No3/Paper_10-Revised_Use_Case_Point_Re-UCP_Model_for_Software.pdf?> [kreiptasi 2015-11-02]

8. PRIEDAI

8.1. Priedas. Sprendimo eksperimento PA aprašymai ir veiklos diagramos

8.1.1. Projektas „Pardavimai“

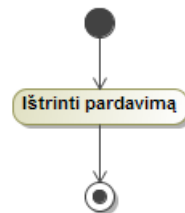


8.8.1 pav. Panaudojimo atvejo „Sukurti pardavimą“ ir „Redaguoti pardavimą“ veiklos diagrama

8.1 lentelė PA „Sukurti pardavimą“, „Redaguoti pardavimą“ formalus aprašymas

PA „Sukurti pardavimą“, „Redaguoti pardavimą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sukurti pardavimą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvedama dokumento data.	Sistema išsaugo duomenis.

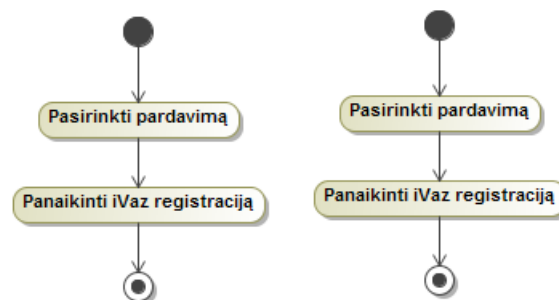
2. Pasirenkamas pirkėjas.	Sistema atidaro pirkėjų lentelę ir išsaugo duomenis.
3. Pasirenkamas padalinys.	Sistema atidaro padalinių lentelę ir išsaugo duomenis.
4. Keičiama valiuta.	Sistema išsaugo duomenis.
5. Keičiamas apmokėjimo dienų skaičius.	Sistema išsaugo duomenis.
6. Įvedamos prekės.	Sistema atidaro prekių įvedimo langą ir išsaugo duomenis.
7. Pritaikoma nuolaida.	Sistema perskaičiuoja kainas.
8. Pasirenkamas operacijos tipas.	Sistema išsaugo duomenis.
9. Įvedama papildoma informacija	Sistema išsaugo duomenis.
10. Įvedamas mokėtojas	Sistema išsaugo duomenis.
11. Patvirtinamas arba išsaugomas pardavimas	Sistema išsaugo arba patvirtina pardavimą.
Po sąlyga:	Sukurtas pardavimas



8.2 pav. Panaudojimo atvejo „Ištrinti pardavimą“ veiklos diagrama.

8.2 lentelė PA „Ištrinti pardavimą“ formalus aprašymas

PA „Ištrinti pardavimą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sistemoje ištrinti pardavimą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Ištrinamas pardavimas	Sistema ištrina duomenis.
Po sąlyga:	Ištrintas pardavimas.



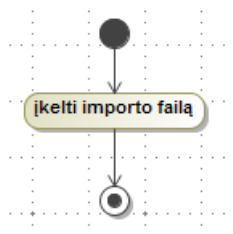
8.3 pav. PA „Registruoti iVaz“ ir „Panaikinti iVaz registraciją“ veiklos diagramos

8.3 lentelė PA „Registruoti iVaz“ formalus aprašymas

PA „Registruoti iVaz“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori pardavimą registruoti iVaz
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamas pardavimas	Sistema išsaugo pasirinkta pardavimą.
2. Registruojama iVaz	Registracijos API registruoja pardavimą.
Po sąlyga:	Pardavimas užregistruotas iVaz.

8.4 lentelė PA „Panaikinti iVaz registraciją“ formalus aprašymas

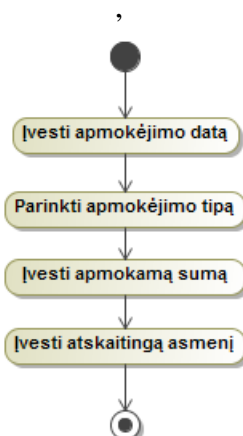
PA “ Panaikinti iVaz registraciją“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori panaikinti iVaz registraciją
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamas pardavimas	Sistema išsaugo pasirinkta pardavimą.
2. Naikinama iVaz registracijos	Registracijos API panaikina registraciją.
Po sąlyga:	Panaikinta iVaz registracija.



8.4 pav. PA „Importuoti pardavimą“ veiklos diagrama

8.5 lentelė PA „Importuoti pardavimą“ formalus aprašymas

PA “ Importuoti pardavimą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos. Vartotojas turi importuojama pardavimo failą.
Sužadinimo sąlyga	Vartotojas nori įkelti pardavimą į sistemą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įkeliamas importo failas.	Sistema išsaugo pardavimą.
Po sąlyga:	Į sistemą įkeltas pardavimas.



8.5 pav. PA „Sukurti apmokėjimą“ veiklos diagrama

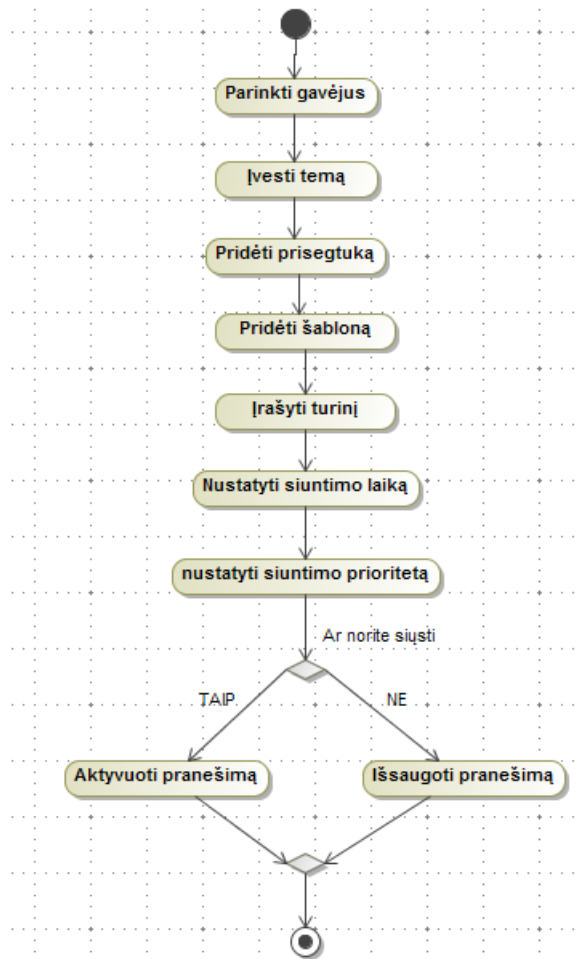
8.6 lentelė PA „Įvesti apmokėjimą“ formalus aprašymas

PA “ Įvesti apmokėjimą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori įvesti apmokėjimą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvedama apmokėjimo data.	Sistema išsaugo datą.
2. Parenkamas apmokėjimo tipas	Sistema išskleidžia tipų sąrašą.
3. Įvedama apmokama suma	Sistema išsaugo duomenis
4. Įvedamas atskaitingas asmuo	Sistema atidaro atskaitingų asmenų lentelę ir išsaugo duomenis.
Po sąlyga:	Suformuotas apmokėjimas

8.1.2. Projektas „Pranešimai“

8.7 lentelė Panaudojimo atvejo „Sukurti pranešimą“ ir „Redaguoti pranešimą“ formalus aprašymas

PA „Sukurti pranešimą“, „Redaguoti pranešimą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori išsiųsti laišką
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
12. Pasirenkamas gavėjas	Sistema pateikia gavėjų sąrašą arba leidžia įvesti tekstą.
13. Įrašoma tema	Sistema išsaugo duomenis
14. Įrašomas tekstas	Sistema išsaugo duomenis
15. Pridedamas prisegtukas	Sistema išsaugo duomenis
16. Pridedamas šablonas	Sistema pateikia šablonų sąrašą.
17. Nustatomas siuntimo laikas	Sistema išsaugo duomenis
18. Nustatomas siuntimo prioritetas	Sistema išsaugo duomenis
19. Aktyvuojamas pranešimas	Sistema išsaugo pranešimą ir paruošia siuntimui.
Po sąlyga:	Išsiųstas pranešimas
Alternatyvūs scenarijai	
A1. Jei nesukonfigūruotas elektroninis paštas	Sistema praneša, kad pranešimas neišsiųstas, nes nesukonfigūruotas elektroninis paštas.

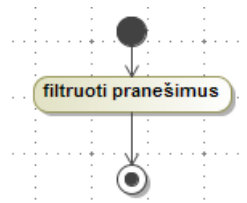


8.6 pav. Panaudojimo atvejo „Sukurti pranešimą“ veiklos diagrama

8.8 lentelė PA „Filtruoti pranešimus“ formalus aprašymas

PA „Filtruoti pranešimus“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori filtruoti pranešimus.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamas filtras	Sistema pateikia filtrų sąrašą
2. Filtruojama	Sistema atfiltruoja pranešimus

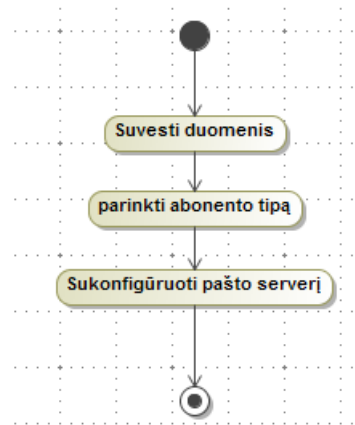
Po sąlyga:	Atfiltruoti pranešimai
------------	------------------------



8.7 pav. Panaudojimo atvejo „Filtruoti pranešimus“ veiklos diagrama

8.9 lentelė PA „Sukonfigūruoti elektroninį paštą“ formalus aprašymas

PA „Sukonfigūruoti elektroninį paštą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sukonfigūruoti elektroninį paštą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Suvedami vartotojo duomenys	Sistema išsaugo duomenis
2. Parenkamas abonentų tipas	Sistema duoda abonentų tipų sąrašą
3. Sukonfigūruojamas pašto serveris	Sistema sukonfigūruoja pašto serverį.
Po sąlyga:	Sukonfigūruotas elektroninis paštas.

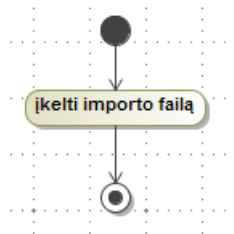


8.8 pav. Panaudojimo atvejo „Sukonfigūruoti elektroninį paštą“ veiklos diagrama

8.1.3. Projektas „Banko importas“

8.10 lentelė PA „Importuoti banką“ formalus aprašymas

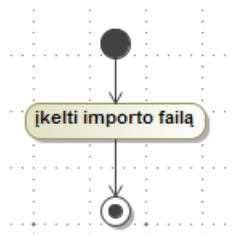
PA „Importuoti banką“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos. Vartotojas turi importuojama banko failą.
Sužadinimo sąlyga	Vartotojas nori įkelti bankines operacijas į sistemą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įkeliamas importo failas.	Sistema išsaugo bankines operacijas.
Po sąlyga:	Į sistemą įkeltas banko importo failas.
Alternatyvūs scenarijai	
A1. Jei netinkamas importo failas.	Sistema praneša, kad importo failas yra netinkamas.



8.9 pav. Panaudojimo atvejo „Importuoti banką“ veiklos diagrama

8.11 lentelė PA “ Importuoti PayPal“ formalus aprašymas

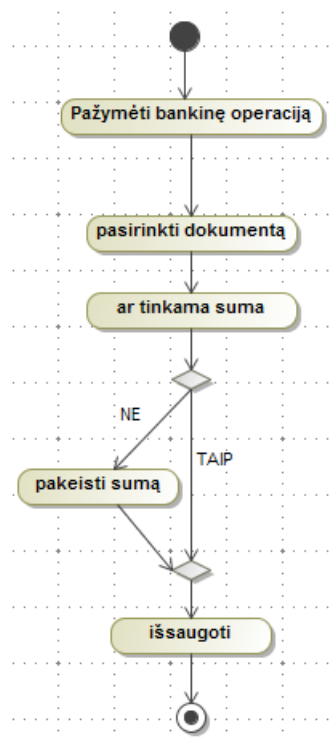
PA “ Importuoti PayPal“	
Prieš sąlygą	Vartotojas prisijungęs prie verslo valdymo sistemos. Vartotojas turi importuojamą <i>PayPal</i> failą.
Sužadinimo sąlyga	Vartotojas nori įkelti <i>PayPal</i> operacijos į sistemą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įkeliamas importo failas.	Sistema išsaugo <i>PayPal</i> operacijas.
Po sąlyga:	Į sistemą įkeltas banko importo failas.
Alternatyvūs scenarijai	
A1. Jei netinkamas importo failas.	Sistema praneša, kad importo failas yra netinkamas.



8.10 pav. Panaudojimo atvejo „Importuoti PayPal“ veiklos diagrama

8.12 lentelė PA “ Sudengti“ formalus aprašymas

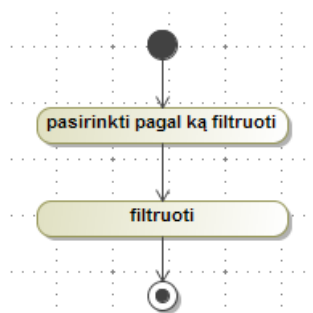
PA “ Sudengti“	
Prieš sąlygą	Vartotojas prisijungęs prie verslo valdymo sistemos. Banko/ <i>PayPal</i> importo failas sukeltas į sistemą.
Sužadinimo sąlyga	Vartotojas nori sudengti bankines operacijas su sistemoje esančiomis atsiskaitymų operacijomis.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkama bankinė operacija.	Sistema pažymi bankinę operaciją.
2. Pasirenkamas dokumentas su kuriuo bus sudengta bankinė operacija.	Sistema išsaugo duomenis.
3. Patikrinama ar bankinės operacijos suma sutampa su norima dengti suma, esant reikalui pakoreguojama.	Išsaugoma dengiama suma.
4. Išsaugomas sudengimas.	Sistema sudengia bankinę operaciją su sistemoje esančia atsiskaitymo operacija. Tiekėjo/pirkėjo skola esanti atsiskaitymuose yra pašalinama.
Po sąlyga:	Sudengta pirkėjo/tiekėjo skola su bankine operacija.



8.11 pav. Panaudojimo atvejo „Sudengti“ veiklos diagrama

8.13 lentelė PA „Filtruoti“ formalus aprašymas

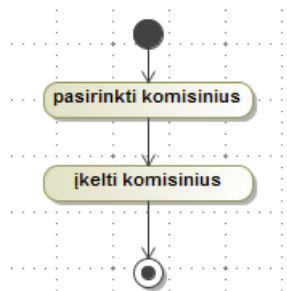
PA „Filtruoti“	
Prieš sąlygą	Vartotojas prisijungęs prie verslo valdymo sistemos. Banko/PayPal importo failas sukeltas į sistemą.
Sužadinimo sąlyga	Vartotojas nori atfiltruoti bankines operacijas.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamas filtras.	Sistema atfiltruoja bankines operacijas.
Po sąlyga:	Atfiltruotos bankinės operacijos.



8.12 pav. Panaudojimo atvejo „Filtruoti“ veiklos diagrama

8.14 lentelė PA „Įkelti komisinius“ formalus aprašymas

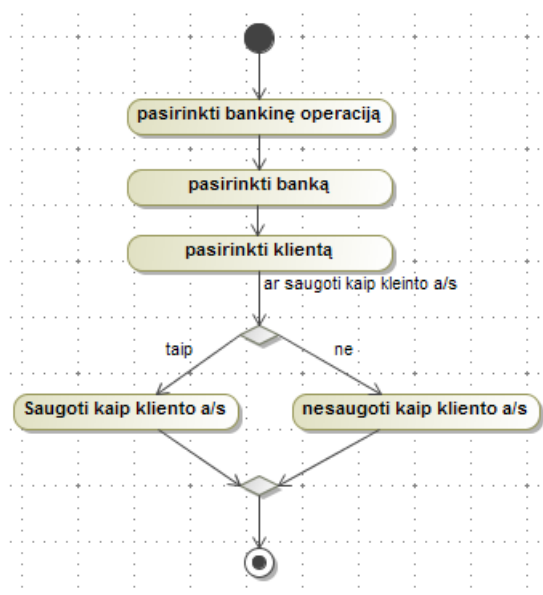
PA „Įkelti komisinius“	
Prieš sąlygą	Vartotojas prisijungęs prie verslo valdymo sistemos. Banko/PayPal importo failas sukeltas į sistemą.
Sužadinimo sąlyga	Vartotojas nori atfiltruoti bankines operacijas.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamos banko operacijos – komisiniai.	Sistema išsaugo duomenis.
2. Įkeliami komisiniai.	Sistema įkelia bankines operacijas – komisinius į sistemos atsiskaitymų modulį.
Po sąlyga:	Į sistemą įkelti komisiniai.



8.13 pav. Panaudojimo atvejo „Įkelti komisinius“ veiklos diagrama

8.15 lentelė PA „Priskirti A/S“ formalus aprašymas

PA „Priskirti A/S“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos. Vartotojas yra suimportavęs bankines/PayPal operacijas.
Sužadinimo sąlyga	Vartotojas nori priskirti atsiskaitomąją sąskaitą prie bankinės operacijos.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkama bankinė/PayPal operacija.	Sistema išsaugo duomenis.
2. Pasirinkti banką.	Sistema išsaugo duomenis.
3. Pasirinkti klientą.	Sistema išsaugo duomenis.
4. Saugoti/nesaugoti kaip kliento A/S.	Sistema išsaugo/neišsaugo kliento atsiskaitomosios sąskaitos.
Po sąlyga:	Prie bankinės operacijos priskirta atsiskaitomoji sąskaita.



8.14 pav. Panaudojimo atvejo „Priskirti A/S“ veiklos diagrama

8.1.4. Projektas „Valiutos“

8.16 lentelė PA „Sukurti valiutą“ formalus aprašymas

PA „Sukurti valiutą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sistemoje susikurti naują valiutą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvedamas valiutos kodas	Sistema išsaugo duomenis.
2. Įvedamas valiutos pavadinimas	Sistema išsaugo duomenis.
3. Įvedama pajamų sąskaitą	Sistema išsaugo duomenis.
4. Įvedama išlaidų sąskaitą	Sistema išsaugo duomenis.
5. Įvedamas išlaidų straipsnis	Sistema išsaugo duomenis.
6. Įvedamas valiutos kursas	Sistema išsaugo duomenis.
Po sąlyga:	Sukurtas valiutos kursas.

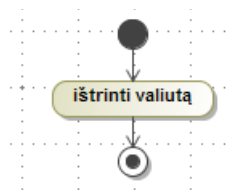
PA „Redaguoti valiutą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sistemoje sukurti naują valiutą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvedamas valiutos kodas	Sistema išsaugo duomenis.
2. Įvedamas valiutos pavadinimas	Sistema išsaugo duomenis.
3. Įvedama pajamų sąskaitą	Sistema išsaugo duomenis.
4. Įvedama išlaidų sąskaitą	Sistema išsaugo duomenis.
5. Įvedamas išlaidų straipsnis	Sistema išsaugo duomenis.
6. Įvedamas valiutos kursas	Sistema išsaugo duomenis.
Po sąlyga:	Sukurtas valiutos kursas.



8.15 pav. Panaudojimo atvejo „Sukurti valiutą“ veiklos diagrama

8.17 lentelė PA „Trinti valiutą“ formalus aprašymas

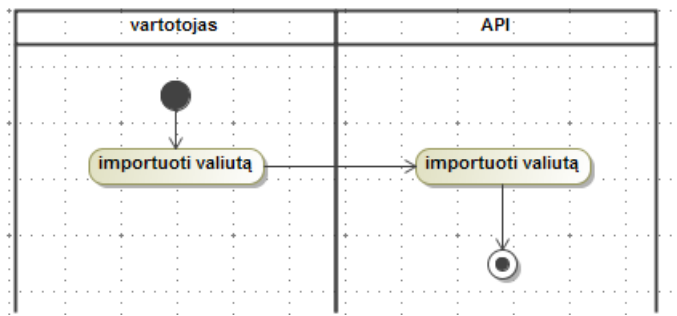
PA „Trinti valiutą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori sistemoje ištrinti valiutą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
5. Ištrinama valiuta	Sistema ištrina duomenis.
Po sąlyga:	Ištrinta valiuta.



8.16 pav. Panaudojimo atvejo „Trinti“ veiklos diagrama

8.18 lentelė PA „Importuoti valiutą“ formalus aprašymas

PA „Importuoti valiutą“	
Prieš sąlyga	Vartotojas prisijungęs prie verslo valdymo sistemos.
Sužadinimo sąlyga	Vartotojas nori importuoti valiutų kursus iš Lietuvos banko.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Paspaudžiamas mygtukas „Importuoti“	API importuoja valiutą.
Po sąlyga:	Suimportuotos valiutos iš LB.



8.17 pav. Panaudojimo atvejo „Importuoti“ veiklos diagrama