

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Džulija Leščinskaitė

**Informacinių sistemų projektų darbų apimties
nustatymo metodika**

Magistro darbas

Darbo vadovas

prof. dr. L. Nemuraitė

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Džulija Leščinskaitė

**Informacinių sistemų projektų darbų apimties
nustatymo metodika**

Magistro darbas

Recenzentas

doc. dr. Eimutis Karčiauskas

2010-05-

Vadovas

prof. dr. L. Nemuraitė

2010-05-

Atliko

IFM 4-4 gr. stud.

Džulija Leščinskaitė

2010-05-28

Kaunas, 2010

Turinys

SUMMARY	5
SANTRAUKA	6
1. ĮVADAS	7
2. IS PROJEKTŲ DARBŲ APIMTIES NUSTATYMO METODIKŲ ANALIZĖ	11
2.1. ANALIZĖS TIKSLAS	11
2.2. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA	11
2.3. TYRIMO OBJEKTO ANALIZĖ	12
2.3.1. <i>Esama IS projektų darbų apimties įvertinimo situacija</i>	12
2.4. VERTINIMO METODŲ ANALIZĖ	18
2.4.1. Algoritminis kainos modeliavimas	18
2.4.1.1. <i>COCOMO modelis</i>	20
2.4.1.2. <i>SLIM modelis</i>	23
2.4.1.3. <i>Funkcinių taškų modelis</i>	24
2.4.2. <i>Funkcinės apimties matavimo modelių standartai</i>	26
2.4.2.1. <i>Albrecht / IFPUG Function Point Analysis (Albrecht / IFPUG) standartas</i>	27
2.4.2.2. <i>Mark II Function Point Analysis (Mark II FPA) standartas</i>	28
2.4.2.3. <i>COSMIC Full Function Points (COSMIC FFP) standartas</i>	28
2.4.2.4. <i>Functional Size Measurement FSM Method (FiSMA) standartas</i>	29
2.4.2.5. <i>Bendras funkcinės apimties matavimo metodų procesas ir skirtumai</i>	30
2.5. DARBŲ APIMTIES NUSTATYMO TYRIMO UŽDAVINIO FORMULUOTĖ	33
2.5.1. <i>Vartotojų analizė</i>	34
2.5.1.1. <i>Vartotojų aibė, tipai ir savybės</i>	34
2.5.1.2. <i>Vartotojų tikslai ir problemos</i>	34
2.5.2. <i>Siekiamas sprendimas</i>	34
2.5.3. <i>Tyrimo tikslas ir uždaviniai</i>	36
2.5.4. <i>Rizikos faktorių analizė</i>	36
2.5.5. <i>Rezultato kokybės kriterijai</i>	37
2.6. ANALIZĖS IŠVADOS	37
3. FISMA METODIKOS REIKALAVIMŲ SPECIFIKACIJA IR ANALIZĖ	39
3.1. REIKALAVIMŲ SPECIFIKACIJA	39
3.2. SPRENDIMO TAIKYMO SRITIS	39
3.3. NEFUNKCINIAI REIKALAVIMAI	39
3.4. FUNKCINIAI REIKALAVIMAI	40
3.5. KOMPIUTERIZUOJAMŲ PANAUDOJIMO ATVEJŲ DIAGRAMA	40
3.6. REIKALAVIMŲ PANAUDOJIMO ATVEJŲ SPECIFIKACIJA	41
3.7. DALYKINĖS SRITIES MODELIS	54
3.8. REIKALAVIMŲ ANALIZĖS APIBENDRINIMAS	55
4. „FISMA 1.1“ METODIKOS APRAŠAS	56
4.1. FORMALIZUOTAS „FISMA 1.1“ METODIKOS APRAŠAS	57
4.1.1. <i>„FiSMA 1.1“ klasės ir tipai</i>	57
4.1.2. <i>Skaičiavimo taisyklės kiekvienam tipui</i>	59
4.1.3. <i>Funkcinės apimties apskaičiavimas programinės įrangos daliai</i>	62
4.1.4. <i>Funkcinės apimties įvertinimo taikymo procesas</i>	62
4.1.5. <i>Detalizuotas taikymo procesas</i>	65
4.1.5.1. <i>Vartotojo paslaugos</i>	65
4.1.5.2. <i>Loginė sistemos architektūra</i>	66
4.1.5.3. <i>Panaudojimo atvejų realizacijos klasių diagramos</i>	67
4.1.5.4. <i>Sistemos panaudojimo atvejus realizuojančios sekų diagramos</i>	72
4.1.6. <i>Sistemos klasių modelis</i>	79
4.1.7. <i>Duomenų bazės schema</i>	81
4.2. METODIKOS APIBENDRINIMAS	82
5. SISTEMOS REALIZACIJA PAGAL FISMA METODIKĄ	83
5.1. REALIZUOTA FUNKCINIŲ REIKALAVIMŲ ĮVERTINIMO IDEOLOGIJA	83
5.2. REALIZUOTAS FUNKCINIŲ REIKALAVIMŲ ĮVERTINIMO PROCESAS	83

5.3. NAUDOTOJO VADOVAS	85
5.4. TESTAVIMO MEDŽIAGA	87
5.5. SUKURTOS SISTEMOS APIBENDRINIMAS	88
6. EKSPERIMENTINIS SISTEMOS REALIZUOTOS PAGAL FISMA METODIKĄ TYRIMAS	90
6.1. EKSPERIMENTO TIKSLAS	90
6.2. EKSPERIMENTO APIBRĖŽIMAS	90
6.3. EKSPERIMENTO VYKDYMAS	91
6.4. EKSPERIMENTO INTERPRETAVIMAS IR PATEIKIMAS	91
6.5. EKSPERIMENTO IŠVADOS	98
7. IŠVADOS.....	100
8. SANTRUMPŲ IR TERMINŲ ŽODYNAS.....	101
9. LITERATŪRA.....	102
10. PRIEDAI	103
10.1. PRIEDAS. NAUDOTOJO VADOVAS.....	103
10.2. PRIEDAS. TESTAVIMO MEDŽIAGA.....	112
10.3. PRIEDAS. APKLAUSOS ANKETA	120
10.4. PRIEDAS. SKIRTUMAI TARP <i>IFPUG FPA</i> , <i>MK II FPA</i> , <i>COSMIC FFP</i> , <i>FISMA FSM</i> METODŲ..	126

Summary

Methodology for functional size measurement of Information system projects

The date of project realization is calculated in the beginning of project, but usually it is over optimistic. The final date can be calculated without considering to the project risks: complexity, constantly changing conditions in the market, the lack of workers' competence.

The aim of this project is to realize the system, according the *IS* project scope of task assessment methodology, called *FiSMA*, that makes easier to calculate the scope of project, to organize tasks, to observe deviations between the planned and factual time, and to store the statistics data.

The relevance of the *FiSMA* was evaluated during an experiment. The experiment with 5 real life projects has shown that the implementation of the method was well accepted by developers. They quickly understood how to calculate the number of function points and spent for this only 3 percents of the time needed for implementing their projects. The received average value of function point for 5 projects was equal to 1,16 hours.

After executed project, it can be assumed that the implemented system should help to evaluate the scope of *IS* project tasks. To find out whether the system is successful over a long period, there is a demand to gather more statistic data, to compare supposed and factual times, and to calculate more accurate functional point value.

Key words: Information system, software engineering, functional size measurement, *FiSMA*, functional point.

Santrauka

Informacinių sistemų projektų darbų apimties nustatymo metodika

Projekto pradžioje apskaičiuota projekto įgyvendinimo data paprastai yra pernelyg optimistinė. Pabaigos data gali būti apskaičiuota neįvertinus projekto rizikos: sudėtingumo, besikeičiančių rinkos sąlygų, darbuotojų kompetencijos stokos.

Šio darbo keliamas tikslas realizuoti sistemą pagal *IS* projektų darbų apimties įvertinimo metodiką *FiSMA*, siekiant palengvinti projekto apimties skaičiavimą, darbų organizavimą, stebėti spėjamo ir faktinio laiko nukrypimus, kaupti statistinius duomenis.

Eksperimento metu analizuojant 5 vykdomus projektus buvo nustatyta, kad sistemoje realizuotą metodą gerai suprato projektuotojai. Funkcinių taškų skaičiavimui jie sugaišo tik 3 procentus visos projekto darbų trukmės. Gauta 5 projektų bendra vieno funkcinio taško vertė yra lygi 1,16 valandos.

Iš atlikto eksperimento galima daryti prielaidą, jog naujai realizuota sistema turėtų padėti tiksliau vertinti *IS* projektų darbų apimtį. Siekiant sužinoti ar sistema pasiteisins ilgesniame laikotarpyje, reikalinga sukaupti daugiau statistinių duomenų. Lyginant projektų spėjamus ir faktinius laikus, bus gaunama vis tikslesnė faktinė funkcinio taško vertė.

Raktiniai žodžiai: Informacinė sistema, programinės įrangos inžinerija, funkcinės apimties nustatymas, *FiSMA*, funkcinis taškas.

1. Įvadas

Kiekvieno projekto tikslas yra per nustatytą laiką neviršijus skirto biudžeto pasiekti reikiamą rezultatą. Projekto inicijavimo fazėje visos projekto apimties detalės nėra žinomos. Pradžioje apskaičiuota projekto įgyvendinimo data paprastai yra pernelyg optimistinė. Projekto pabaigos data gali būti apskaičiuota neįvertinus projekto rizikos: sudėtingumo, besikeičiančių rinkos sąlygų, darbuotojų kompetencijos stokos.

Šio darbo **tyrimo sritis** yra projektų darbų apimties nustatymo metodai. **Tyrimo objektas** – projektų darbų apimties nustatymo procesas.

Tyrimo problema - trūkumas praktinių metodų, leidžiančių įvertinti projektų darbų apimtį. Tam skirtas naujas *FiSMA* standartas, tačiau apie jo taikymą kol kas nėra atsiliepiamų. Norint taikyti *FiSMA*, reikia žinoti funkcinių taško vertę, kuriai nėra nustatytų normų. Kyla abejonių dėl šio standarto pritaikomumo, kadangi jis gali pareikalauti per didelių projektuotojo laiko sąnaudų.

Šio tiriamojo darbo **tikslas** - ištirti *FiSMA* metodikos tinkamumą projekto darbų apimčiai įvertinti, realizuojant darbų apimties įvertinimo sistemą, leidžiančią nustatyti funkcinių taškų įvertinimus pagal kaupiamus faktinius duomenis, ir įvertinant projektuotojo darbo laiko sąnaudas šiai metodikai taikyti.

Darbo uždaviniai:

- Išanalizuoti esamas projektų darbų apimties įvertinimo metodikas.
- Atlikti *FSM* (funkcinės apimties matavimo) standartų analizę.
- Sukurti sistemą, kuri leistų skaičiuoti projekto darbų apimtį pagal *FiSMA* standartą ir kaupti faktinius duomenis funkcinių taškų vertei nustatyti bei patikslinti.
- Atlikti eksperimentą, kuris leistų įvertinti *FiSMA* metodikos efektyvumą, t.y. nustatyti žmogaus atliekamo funkcinių taškų įvertinimo laiką ir jo santykį su visa projekto trukme (ar ne per daug laiko sugaištų projektuotojai, taikydami *FiSMA* metodiką).

Tyrimo metodika. Darbe buvo taikoma literatūros analizė, *FiSMA* metodika, eksperimentinis tyrimas.

Kokybės kriterijai šiame darbe yra realizuotos metodikos tinkamumas projekto darbų apimčiai įvertinti ir jos efektyvumas. Metodikos tinkamumu suprantamas jos suprantamumas ir priimtinas kūrėjams, efektyvumu – darbų apimties skaičiavimo laiko santykis su visu projekto vykdymo laiku. Jei darbų apimčiai skaičiuoti užtrunkama per daug laiko, tokia

metodika neefektyvi. Efektyvumą padidino tai, kad buvo automatizuotas reikalavimų specifikacijų įkėlimas į projektą.

Kadangi projektų vadovų vienas iš darbo aspektų yra projekto darbų apimties nustatymas, kuris padeda planuojant projektą, rengiant biudžetą, vėliau ruošiant specifikacijas ir susijusius dokumentus, todėl buvo apžvelgta esama Lietuvos *IS* projektų darbų apimties įvertinimo situacija, su tikslu išsiaiškinti kylančias apimties nustatymo problemas ir ieškoti alternatyvių sprendimų.

Remiantis literatūros šaltiniu [2], projektai yra vertinami atliekant pirminį spėjimą, kuris naudojamas kaip priemonė pasiūlymų atrankai, o preliminarus vertinimas ir tikslus įvertinimas yra naudojami apimties įvertinimui projekto eigoje. Preliminarus įvertinimas pagrįstas vartotojo poreikių analize ir statistine informacija apie panašių projektų apimtį. Lietuvoje nėra išsamios *IT* projektų statistikos, todėl *IT* kompanijos priverstos vadovautis tik savo patirtimi.

Praktikoje dažniausiai taikomi du pagrindiniai netikslaus apimties įvertinimo rizikos valdymo būdai: valandinio įkainio kontraktai ir fazinis apimties įvertinimas fiksuotos kainos kontraktuose [2]. Valandiniai įkainio kontraktai pasižymi tuo, kad esant neaiškiems reikalavimams, projekto biudžetas ir trukmė apibrėžiami labai abstrakčiai ir užsakovas moka vykdytojui už darbo laiką pagal sutartą valandinį įkainį. Faziniam apimties įvertinimui fiksuotos kainos kontraktuose yra būdingas preliminarus apimties įvertinimas visam projektui ir tikslų įvertinimą artimiausiai fazei, po kiekvienos fazės patikslinamas preliminarus įvertinimas likusiai projekto daliai ir pateikiamas tikslus būsimos fazės įvertinimas.

Šie preliminarūs įvertinimai dažnai sunkiai prognozuojami ir mažai patikimi, kadangi tai – subjektyvūs metodai, kurie yra taikomi remiantis praeities patirtimi ar paremti intuicija. Todėl reikalingas tikslesnis metodikos pasirinkimas.

Metodikos pasirinkimas priklauso nuo įvairių aspektų, tokių kaip: kuriamos sistemos, jos apimties, keliamų funkcinių reikalavimų, reikalaujamo apimties tikslumo, vartotojo ir panašių aspektų.

Siekiant išsirinkti tinkamiausią metodiką buvo remtasi literatūros šaltiniais [4], [5], [9], [11], [12], [13] palygintos *IS* projektų darbų apimties įvertinimo metodikos, tokios kaip algoritminis kainos modeliavimas ir funkcinės apimties matavimo standartai.

Kadangi daugumos algoritminių kainų modeliuose matavimai remiasi kodo eilučių skaičiumi, tikslus kodo apimties įvertinimas yra sudėtingas projekto ankstyvoje stadijoje, ypač jei projektų reikalavimai pastoviai kinta, todėl šių metodikų buvo atsisakyta. Funkcinės apimties matavimas yra naudingesnis, kadangi funkcinės apimties metodai leidžia išmatuoti

programinės įrangos funkcionalumo apimtį [9]. Ši metrika įvertina programinės įrangos apimtį iš vartotojo pozicijų [11].

Nors visi funkcinės apimtys matavimo metodai įvertina apimtį, remiantis funkcionalumu, pagrindinis skirtumas tarp šių technikų kilo iš to, kas yra skaičiuojama ir kaip tai atliekama [13].

Išanalizavus šias metodikas, realizacijai priimtas sprendimas kurti sistemą pagal *FiSMA* standartą. Kadangi šis standartas remiasi objektyviais kriterijais (vertinami labai aiškūs projekto elementai), klasės išskirstytos į itin detalius tipus, kas įtakoja tikslesnį funkcinio reikalavimo įvertinimą. *COSMIC FFP* standartas pasižymi taip pat detaliu įvertinimu, tačiau pritaikomumas yra sudėtingas, kadangi metodas apima ne tik funkcinio reikalavimo apimtys įvertinimą, tačiau ir kainos variavimą, vykdytojų motyvavimą. Standartai *IFPUG FPA* ir *Mk II FPA* yra apribojami vertinant informacines sistemas, t.y šie standartai skirti vertinti verslo sistemas, priešingai *FiSMA* kuri yra vienas naujausių standartų (2008 m.), ir buvo suprojektuotas pritaikyti visoms programinės įrangos tipams [12]. Tai metodika projektuota/koreguota remiantis vartotojų poreikiais, iškilusiomis problemomis.

Remiantis *FiSMA* metodika, sukurta sistema, kurioje projektų vadovas įkėlęs reikalavimus ir sistemai apskaičiavus jų trukmę, galės priskirti juos vykdytojams, kurie juos atlikę fiksuos faktinius funkcinį reikalavimų laikus. Užsibrėžtas tikslas, kad projektų vadovas galėtų efektyviau koordinuoti vykdytojus, mažiau apkrauti save matematiniais apskaičiavimais, mažiau užtrukti laiko keliant reikalavimus į sistemą. Sistemoje esanti galimybė kaupti statistinius duomenis apie prieš tai įvertintus projektus šioje sistemoje, bus pagalba projektų vadovui naujų projektų rengime.

Siekiant išsiaiškinti ar realizuota sistema pagal pasirinktą metodiką padės tiksliau įvertinti projektų apimtį ir bus pagalba projektų vadovui buvo atliktas eksperimentas. Eksperimento metu nustatytos spėjamos ir faktinės programos realizavimo trukmės, įvertintas užtrukimo laikas skaičiuojant projektų elementus. Rezultatai pabrėžė, kad tikslesniam įvertinimui metodika yra reikalinga, eksperimento apklaustųjų projektų faktinis laikas buvo didesnis už spėjamąjį. Gauta 5 projektų bendra faktinė 1 funkcinio vertė yra lygi 1,16 valandos. Vertinant projektų elementus pagal užtrukimo trukmę, procentinė dalis gaunama nedidelė, vertinimo laikas neviršija 3 procentų viso projektų faktinio laiko.

Iš atlikto eksperimento galima daryti prielaidą, jog naujai realizuota sistema turėtų padėti tiksliau ir su nedidelėmis laiko sąnaudomis vertinti IS projektų darbų apimtį. Siekiant sužinoti ar sistema pasiteisins ilgesniame laikotarpyje, reikalinga sukaupti daugiau statistinių duomenų, lyginant projektų spėjamus ir faktinius laikų nuokrypius bus gaunama vis tikslesnė faktinė 1 funkcinio taško vertė.

Šio darbo struktūra:

- Skyriuje „IS projekto darbų apimties nustatymo metodikų analizė“ yra pateikiama esama IS projektų darbų apimties įvertinimo situacija, pateikiamos praktiškai naudojamos metodikos, išskiriamos esamos apimties įvertinimo taisyklės ir jų problemos, nusakoma ko reikia tikslesniam projekto įvertinimui. Pateikiama algoritminio kainos modeliavimo metodų (*Boehm COCOMO*, *Putmano SLIM* ir *Albrecht* funkcinių taškų) apžvalga. Atliekama funkcinės apimties matavimo standartų (4 ISO standartai: *IFPUG Function Point Analysis (ISO/IEC 20926, 2003)*, *Mark II Function Point Analysis (ISO/IEC 20968, 2002)*, *COSMIC Full Function Points (COSMIC FFP, ISO/IEC 19761, 2003)* ir *FiSMA (ISO/IEC 29881, 2008)*) analizė.
- Skyriuje „*FiSMA* metodikos reikalavimų specifikacija ir analizė“ pateikiamos naujai kuriamos sistemos pagal *FiSMA* metodiką funkciniai ir nefunkciniai reikalavimai, suformuojama panaudojimo atvejų diagrama ir panaudojimo atvejų specifikacijos.
- Skyriuje „*FiSMA* metodikos aprašas“ yra pateikiamas funkcinės apimties įvertinimo taikymo procesas, formalizuotas funkcinės apimties įvertinimo metodikos aprašas, kuriame pateikiamos *FiSMA* klasės ir tipai bei skaičiavimo taisyklės kiekvienai klasei. Aprašomas bendras funkcinės apimties apskaičiavimas programinės įrangos daliai. Šiame skyriuje taip pat aprašomos vartotojo paslaugos, loginė sistemos architektūra, panaudojimo atvejų realizacijos klasių diagramos, panaudojimo atvejų analizės modelis, suprojektuota duomenų bazės schema.
- Skyriuje „Sistemos realizacija pagal *FiSMA* metodiką“ pateikiamas realizuotas funkcinių reikalavimų įkėlimo procesas, įvertinimas, naudotojo vadovas, atliekamas testavimas.
- Skyriuje „Eksperimentinis sistemos realizuotos pagal *FiSMA* metodiką tyrimas“ aprašomas atliktas eksperimentas, nustatomas spėjamo, faktinio projekto vykdymo laikas, funkcinio taško vertės pagal tipus, fiksuojamas projektuotojų užgaištas laikas skaičiuojant projekto elementus.

Išvadosse pateikiami šio darbo apibendrinti rezultatai.

2. IS projektų darbų apimties nustatymo metodikų analizė

2.1. Analizės tikslas

Keliamas *tikslas* yra išanalizuoti esamas projektų kūrimo stadijų problemas, susijusias su terminų atlikimo apibrėžimu. Kiekvieno projekto tikslas yra per nustatytą laiką neviršijus skirto biudžeto pasiekti reikiamą rezultatą. Priešingu atveju, išskyla rizika, kad atskiros projekto dalys arba visas projektas truks ilgiau, nei iš pradžių planuota, atsiranda papildomų išlaidų rizika. Kas įtakos į planą neįneinančius nenumatytus darbus ar suplanuotus, kurie truks ilgiau nei nustatyta.

Kadangi, ypač ankstyvoje stadijoje, darbų apimtį yra sudėtinga nustatyti, reikalinga pasirinkti metodiką, kuri padėtų tai įvertinti. Todėl darbe yra siekiama palyginti darbų apimties įvertinimo metodikas, tokias kaip algoritminis kainos modeliavimas ir *FSM* (funkcinės apimties matavimo) standartus, su tikslu rinktis tinkamiausią.

Pagal atlikta analizę ir pasirinktą realizuoti metodiką vėliau bus modeliuojami įvairūs sistemos aspektai, projektuojami sistemos komponentai.

2.2. Tyrimo sritis, objektas ir problema

Viena iš opiausių IT projektų *problemų* yra neplanuotai didėjantys projektų biudžetai ir vėlavimas. Užsakovai bando apsisaugoti fiksuotos kainos sutartimis bei baudomis už vėlavimą. Vykdytojai bando išvengti apimties didėjimo tobulindami reikalavimų pasikeitimų valdymą, reikalaudami papildomo laiko ir apmokėjimo pakeitimams atlikti. Deja, šios priemonės dažniausiai problemos neišsprendžia, bet padidina užsakovo ir vykdytojo konfrontaciją. Norint kovoti su projektų trukmės ir biudžeto didėjimo problema reikalinga suprasti ir pašalinti jos priežastis [2].

Kadangi projekto inicijavimo fazėje visos projekto apimties detalės nėra žinomos, esant tokioms sąlygoms atsiranda tikimybė per klaidą ką nors užmiršti. Nepaisant šio bendro projekto apimties nustatymo, egzistuoja kiti pavojai, tokie kaip pradžioje apskaičiuota projekto įgyvendinimo data paprastai yra pernelyg optimistinė, projekto pabaigos data gali būti apskaičiuota neįvertinus projekto rizikos, taip pat egzistuoja užsakovų spaudimas. Pradinis projekto biudžetas gali neleisti atsitiktinių nukrypimų nuo apimties, suklystų elementų pakartotinio atlikimo ar projekto vėlavimo. Nustatyta projekto apimtis dėl techninių problemų, tvarkaraščio terminų ar biudžeto ribojimo gali būti keičiama, tačiau tai remiasi į kainą.

Projektų vadovų darbo pobūdžio vienas iš aspektų yra projekto darbų apimties nustatymas, kuris padeda planuojant projektą, rengiant biudžetą, organizuojant darbus, vėliau ruošiant specifikacijas, instrukcijas ir susijusius dokumentus.

Apimtis – svarbiausia projektų vadybos grandis. Apimtis yra suprantama kaip kas turės būti padaryta. Į apimtį (plačiąja prasme) įeina ir kokybė. Projektų vadovas, *IBM Lietuva* (Kęstutis Lašinskas) savo pristatyme „Geros ir blogos projektų valdymo praktikos“ išskiria vieną iš svarbiausių aspektų dirbant projektuose - darbo apimtį [3]. Iš to galima teigti, kad laiku ir tinkamai įvertinta darbo apimtis gali padėti pasiekti pageidaujamą kokybę.

Šio darbo *tyrimo sritis* – yra projektu darbu apimties nustatymo metodai. *Tyrimo objektas* – projektu darbu apimties nustatymo procesas.

Kuriant sistemas yra sudėtinga įvertinti jų apimtį, kadangi trūksta praktinių metodų, leidžiančių įvertinti projektų darbų apimtį.

Sprendžiant *problemą* bus siekiama surasti metodiką, kuri būtų tinkamiausia. Bus realizuota sistema pagal šią metodiką ir eksperimentu patikrintas jos pritaikomumas.

2.3. Tyrimo objekto analizė

2.3.1. Esama IS projektų darbų apimties įvertinimo situacija

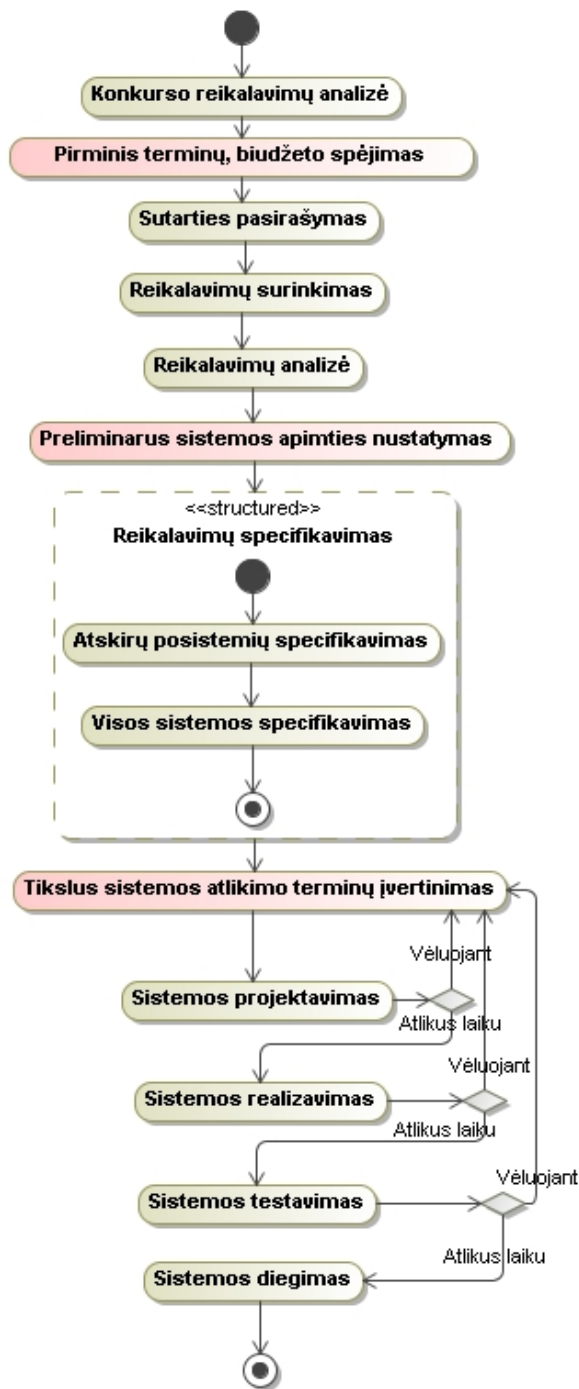
Ilgalaikė *IT* projektų patirtis rodo, kad fiksuotos projekto trukmės ir apimties įvertinimas neturint profesionaliai išanalizuotų ir suformuluotų reikalavimų yra vienas iš pagrindinių *IT* projektų rizikos šaltinių, sukeltis projekto biudžeto viršijimo ir trukmės pailgėjimo rizikas [2].

Pirmoji problema, su kuria susiduria projekto vadovas ir projekto užsakovas yra nesutarimas dėl projekto pradžioje atliekamo projekto apimties įvertinimo tikslumo.

Įvertinimo metodikos apibrėžia tris įvertinimo tikslumo lygius:

- pirminį spėjimą,
- preliminarų įvertinimą,
- tikslų įvertinimą.

Įvertinimo lygiai išdėstyti veiklų diagramos 1 paveiksle:



1 pav. Bendras esamų IS projekto darbų apimties nustatymas

Pirminis spėjimas yra naudojamas kaip priemonė pasiūlymų atrankai, o preliminarus vertinimas ir tikslus įvertinimas yra naudojami apimties įvertinimui projekto eigoje.

Preliminarus įvertinimas pagrįstas vartotojo poreikių analize ir statistine informacija apie panašių projektų apimtį. Lietuvoje nėra išsamios IT projektų statistikos, todėl IT kompanijos priverstos vadovautis tik savo patirtimi. Kai statistinė informacija nekaupiama įmonės viduje – lieka pasikliauti tik projekto vadovo patirtimi. Dėl šių priežasčių preliminarių įvertinimų patikimumas Lietuvoje yra sunkiai prognozuojamas ir mažai patikimas [2].

Tiksliam projekto įvertinimui reikia:

Parengti *IT* sistemos reikalavimus pagal turimus vartotojo poreikius, paruošti užduočių sąrašą ir įvertinti kiekvienos užduoties apimtį.

Kai reikalavimai parengti korektiškai - įvertinimo tikslumas siekia 80 procentų. Tačiau vartotojo poreikių analizė ir sistemos reikalavimų parengimas paprastai užima 20-30 procentų viso *IT* projekto laiko, todėl atlikti šį darbą per preliminarų sutarties sąlygų aptarimą yra neįmanoma.

Užsakovai siekdami kuo tiksliau suplanuoti projekto biudžetą ir trukmę pageidauja sudaryti fiksuotos kainos kontraktus. Sutartyje nurodytą projekto apimtį jie traktuoja kaip tikslų projekto apimtį įvertinimą. Tuo tarpu projekto vadovas ir vykdytojas, turėdamas tik pirminius vartotojo poreikius, pačiu geriausiu atveju gali pateikti tik preliminarų projekto apimtį įvertinimą. Dėl šio užsakovo ir vykdytojo nesusikalbėjimo projekto biudžetas ir detalus projekto planas ruošiamas pagal preliminarų (netikslų) įvertinimą. Toks biudžetas ir tvarkaraštis yra vadinamas nerealistiniu.

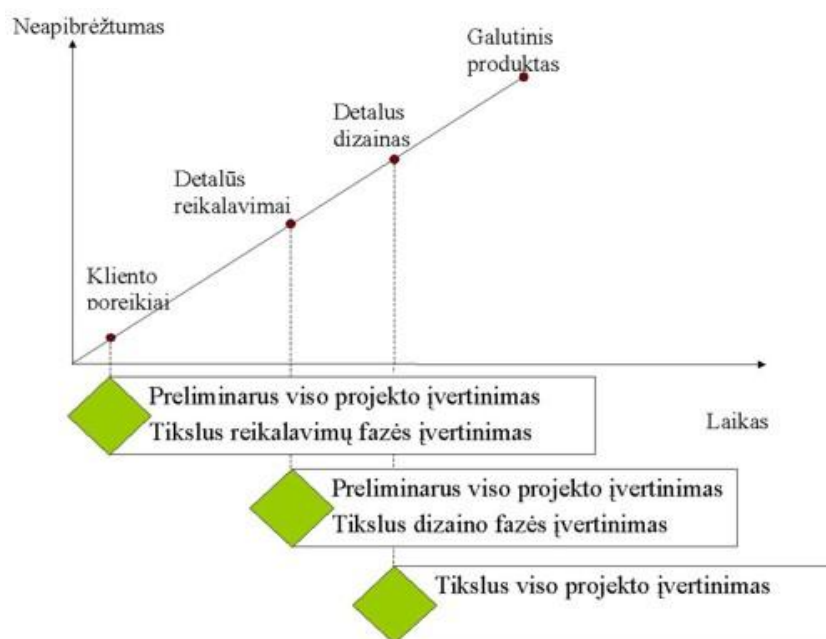
Barry Boehm, vieno iš programinės įrangos rizikų valdymo pradininkų vertinimu, būtent nerealistiniai biudžetai ir tvarkaraščiai yra antroji priežastis pagal dažnį ir svarbą 10 svarbiausių programinės įrangos rizikų sąrašė. Jie turi įtakos ne tik projekto vėlavimui ir biudžeto augimui, bei ir su tuo susijusiam *IT* projekto žlugimui.

Pagrindinė programinės įrangos projekto vadovo užduotis – identifikuoti šią, netikslaus apimtį įvertinimo riziką, supažindinti užsakovą ir pateikti rizikos valdymo pasiūlymus [2].

Praktikoje dažniausiai taikomi du pagrindiniai netikslaus apimtį įvertinimo rizikos valdymo būdai:

- Valandinio įkainio kontraktai. Esant neiškiems reikalavimams, projekto biudžetas ir trukmė apibrėžiami labai abstrakčiai ir užsakovas moka vykdytojui už darbo laiką pagal sutartą valandinį įkainį. Tokiuose kontraktuose užsakovas dažniausiai skiria projekto vadovą, kuris planuoja ir kontroliuoja projekto komandos darbą.
- Fazinis apimtį įvertinimas fiksuotos kainos kontraktuose. Projekto pradžioje projekto vadovas pateikia preliminarų apimtį įvertinimą visam projektui ir tikslų įvertinimą artimiausiai fazei. Po kiekvienos fazės patikslinamas preliminarus įvertinimas likusiai projekto daliai ir pateikiamas tikslus būsimo

fazės įvertinimas. Klasikinis programinės įrangos kūrimo projekto fazinio įvertinimo pavyzdys pateiktas 2 paveiksle.



2 pav. Fazinio įvertinimo pavyzdys programinės įrangos projektui

Programinės įrangos užsakovas turi atsižvelgti į tai, kokia informacija disponuoja projekto vadovas konkrečiame projekto etape, realistiškai vertinti projekto vadovo galimybes pateikti tikslus įvertinimus ir pasirinkti jam priimtinus netikslaus įvertinimo rizikos valdymo būdus.

Kita, aktuali apimties įvertinimo problema yra įvertinimo patikimumas. Kadangi užsakovai, pateikę pirkimo pasiūlymus keletui potencialių vykdytojų, gauna skirtingus preliminarinius įvertinimus. Šiuo atveju, vertintojams gavus vienodą pirminę informaciją, vertinimų skirtumams turi įtakos vertintojo kvalifikacija.

Esamos apimties įvertinimo taisyklės ir jų problemos

Vertinimo patikimumas bus mažas, kai turimi pradiniai duomenys netinka taikomai metodikai. Pavyzdžiui, projekto vadovas gauna pirminį užsakovo poreikių dokumentą, parengia užduočių sąrašą ir pritaiko tikslaus vertinimo metodiką. Gautas apimties įvertinimas yra labai netikslus, kadangi nebuvo atlikta užsakovo poreikių analizė ir programinės įrangos reikalavimų parengimas, kurio metu atrandama 30 - 50 proc. paslėptų reikalavimų. Deja, jų įgyvendinimui nenumatytas laikas pateiktame apimties įvertinime.

Maksimalus įvertinimo tikslumas nebus pasiektas, kai pasirinkta vertinimo metodika nepakankamai panaudoja turimus pradinius duomenis. Pavyzdžiui, užsakovas pateikia

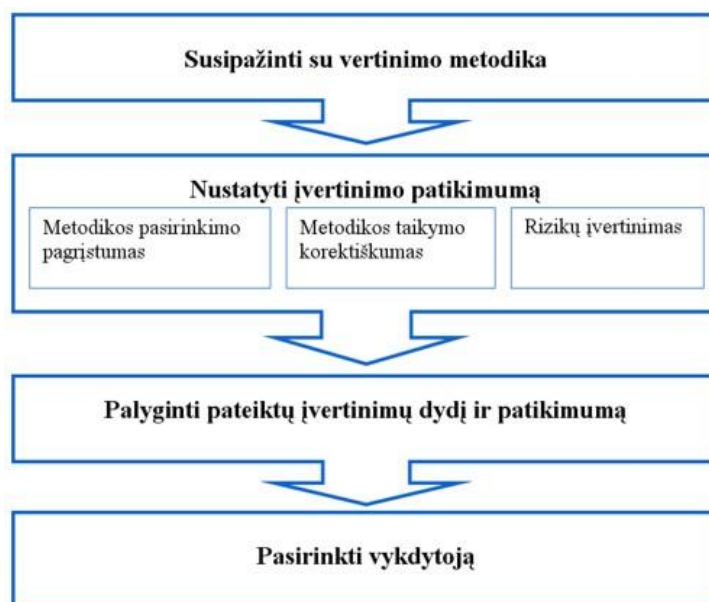
parengtą programinės įrangos reikalavimų dokumentą, tačiau vykdytojas, taupydamas vertinimo laiką, nedaro tikslaus apimties įvertinimo kiekvienam reikalavimui. Jis pasirenka vertinimo iš viršaus būdą ir pateikia preliminarų vertinimą, paremtą analogiškų projektų patirtimi.

Įvertinimo patikimumas priklauso nuo pradinių duomenų patikimumo. Didžioji vertinimo metodikų dalis pagrįsta statistiniais duomenimis arba iš statistinių duomenų gautomis empirinėmis formulėmis, kurių patikimumas tiesiogiai priklauso nuo turimų statistinių duomenų patikimumo. Pavyzdžiui, projekto vadovo nuolatinė veikla - interneto svetainių kūrimo projektai. Iš patirties jis žino, kad tokio tipo projektuose apie 60 proc. programavimo laiko užima vartotojo sąsajos sukūrimas, o 40 proc. laiko užima serverio programavimo darbai. Vertindamas interneto pardavimų sistemos sukūrimo projektą, projektų vadovas peržiūri vartotojo sąsajos langus, įvertina kiek reikės laiko jų sukūrimui ir pritaikęs 40/60 paskirstymo formulę, įvertina serverio dalies programavimo laiką. Šiuo atveju, įvertinimo paklaida bus didelė, kadangi pritaikyta paskirstymo formulė netinka interneto pardavimų sistemos įvertinimui. Formulėje 40/60 neįtraukta pardavimų sistemoje reikalinga verslo logika, dėl kurios serverio programavimo laikas yra ilgesnis nei paprastos interneto svetainės [2].

Siekiant didesnio įvertinimo tikslumo labiau pasireiškia įvertinimą atliekančio asmens patirties ir kvalifikacijos įtaka. Preliminarų įvertinimą galima gauti panaudojus statistinių duomenų bazę ir automatizuotus įrankius. Kai atliekamas tikslus užduoties apimties įvertinimas, vertintojas privalo suprasti užduotį, įvertinti jos sudėtingumą, turėti užduočiai atlikti reikalingą kvalifikaciją. Projekto vadovas tikslų vertinimui turėtų pasitelkti tinkamus žmones ir pasirinkti tinkamą darbų apimties įvertinimui skirtą metodiką, siekiant projekto įvertinimo pagrįstumo.

Projekto apimties įvertinimas apima ne tik projekto įgyvendinimo trukmę ir kaštus, bet ir projekto rizikos valdymo kaštus. Rizikos valdymo įvertinimas yra numatytas iš kaštų ir laiko, skirto įvertinti neįtrauktų projekto problemų sprendimui.

Lyginant skirtingų vykdytojų pateiktus projekto apimties įvertinimus turi būti svarstomas ne tik galutinis valandų skaičius ar kaina, bet ir pasirinktos vertinimo metodikos tinkamumas, pateikto įvertinimo patikimumas, rizikos įvertinimas. Sprendimo priėmimo procesas, kurį reiktų atlikti analizuojant skirtingų vykdytojų pateiktus įvertinimus, parodytas 3 paveiksle [2].



3 pav. Sprendimo priėmimo procesas renkantis projekto vykdytoją pagal pateiktus projekto apimties įvertinimus

Apibendrinant galima teigti, jog projekto apimties įvertinimo tikslumas turi didelę įtaką projekto sėkmei, todėl abu svarbiausi projekto dalyviai – projekto vadovas ir užsakovas – turi siekti maksimaliai tikslių įvertinimų. Siekiant tiksliau apskaičiuoti reikalinga ne tik projekto vadovo kvalifikacija susipažįstant su esamomis vertinimo metodikomis, bet ir sugebėjimas jas taikyti konkrečiose situacijose. Taip pat be asmeninių projekto vadovo pastangų būtinas kompanijos vadovybės palaikymas ir užtikrinimas, kad kompanijoje būtų renkama, apdorojama ir kaupiama reikalinga statistinė informacija, išigyjami reikalingi darbo įrankiai. Taipogi, norint efektyvumo įvertinime užsakovas turi kontroliuoti įvertinimo procesą, reikalauti iš vertintojo pagrįstų įvertinimo dokumentų.

Siekiant šių tikslų įgyvendinimo yra tikslinga kurti sistemą pagal darbų apimties įvertinimo metodiką, kuri įgalintų projektų vadovus įverti kiek įmanoma tiksliau darbų apimtį, įvertinant kiekvieną reikalavimą, kas įtakotų didesnę pasitikėjimą iš užsakovo pozicijų. Tokios sistemos sukūrimas padėtų ne tik stebėti, bet ir kontroliuoti veiklas, tai padėtų palaikyti glaudų ryšį tarp vykdymo komandos ir projekto vadovo, kas didintų darbų efektyvumą. Atliekant pakartotinį panaudojimą, galima būtų stebėti planuotus terminus su faktiniais.

2.4. Vertinimo metodų analizė

Nėra paprasto būdo tiksliai įvertinti programų sistemos reikalingas kūrimo pastangas, jos apimtį, kadangi pradiniai vertinimai remiasi neadekvačia reikalavimų apibrėžimo informacija, programos gali naudoti naujas technologijas, projekte dirbantys žmonės nepakankamai pažįstami [6].

Apimties įvertinimas yra svarbus, kai nėra pakankamai informacijos. Kuomet galima apskaičiuoti darbų apimtį su dideliu tikslumu, pastangų ir trukmės informacija nėra reikalinga [11].

Kiekvienas metodas turi savo silpnybes ir stiprybes. Įvertinimas turi remtis keliais metodais. Jeigu jie neduoda panašių rezultatų, tai reiškia, kad vertinimui informacija yra nepakankama. Tokiu atveju reikia papildomų veiksmų, norint gauti daugiau informacijos.

Įvertinimo metodai yra klasifikuojami į dvi pagrindines kategorijas, priklausomai nuo jų pobūdžio į tiesioginius įvertinimo metodus ir įgytus įvertinimo/matavimo metodus [11].

Tiesioginiai įvertinimo metodai yra paremti ekspertų nuomone, tai – subjektyvūs metodai, kurie yra taikomi remiantis praeities patirtimi ar paremti intuicija, kurie dažniausiai taikomi išanalizuotoje esamoje IS darbų apimties įvertinimo situacijoje. Vis dėlto, šie metodai nėra tikslūs, todėl sekantys apžvelgtieji metodai yra priskiriami įgytiems įvertinimo/matavimo metodams. Šie metodai dar yra žinomi kaip algoritminiai modeliavimo metodai.

2.4.1. Algoritminis kainos modeliavimas

Algoritminis kainos modeliavimas naudoja matematinės formules nustatyti projekto kainas, paremtas projekto apimties įvertinimu, projekto vykdytojų skaičiumi ir kitus procesus ir produkto faktorius. Paprastai nustatomos tam tikros buvusių projektų charakteristikos: trukmė, kaina, projekto komandos dydis, konkretūs kiekybiniai programų sistemos rodikliai (programos eilučių, operatorių ar kitų programinių objektų skaičius). Dideliems projektams, paprastai tokiems, kuriuos rengia daugiau kaip trys žmonės, sąnaudų ir programų sistemos dydžio priklausomybė yra eksponentinė (kaina priklausomai nuo projekto dydžio didėja netiesiškai, nes didėjant projektui reikia papildomų sąnaudų dėl didėjančių ryšių kiekio, sudėtingesnio konfigūracijos valdymo, sunkesnės integracijos ir pan.):

$$\text{Effort} = A \cdot \text{Size}^B \cdot M$$

A yra konstanta, kuri priklauso nuo vietinės organizacinės praktikos ir nuo programinės įrangos plėtojimo. Apimtis gali būti kaip įvestis kodo apimties, funkcionalumu, išreikštų funkcijų ir objektų taškais. Eksponentės B vertė visada būna nuo 1 iki 1.5, ji atspindi

pastangų neproporcingumą dideliems projektams. M yra daugiklis atspindintis produkto, proceso ir projektavimo atributus, tokius kaip reikalavimų patikimumas ir patirtis projektavimo komandos. Daugiausia algoritminės kainos įvertinimo modelių turi eksponentinį komponentą (B daugiau negu vienas lygtyje), tai yra susiję su apimties nustatymu. Tai atspindi faktą, kad kaina išauga priklausomai nuo projekto apimties. Kai programinės įrangos kaina išauga, reikalingas papildomos kainos indėlis, dėl organizacinių sąlygų sudarymo didesnei komandai, sudėtingesnis konfigūracijos valdymas, sunkesnė sistemos integracija ir t.t. Kuo didesnė sistema, tuo didesnė vertė šios eksponentės.

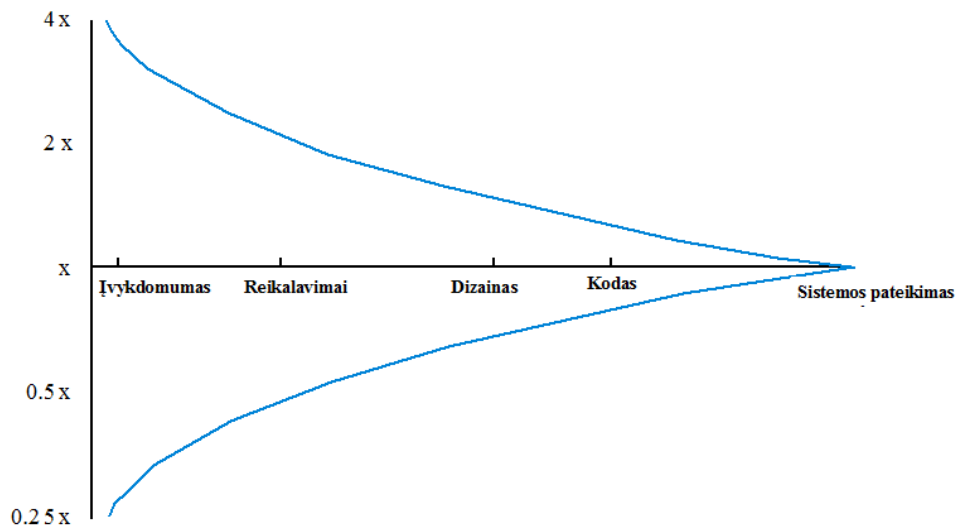
Algoritminis kainos modelis leidžia vykdyti planavimą ir palyginti alternatyvias strategijas, tačiau esama trūkumų:

- Dažnai yra sudėtinga įvertinti sistemos apimtį ankstyvoje projekto stadijoje, kuomet sudaryta tik specifikacija. Funkcijos ir objektų taškų įvertinimą yra lengviau pateikti nei kodo skaičių, bet vis tiek tai išlieka netikslu.
- Įvertinimas faktorių, įtakančių B ir M yra subjektyvus. Įvertinimas gali kisti, nuo vieno asmens, prie kito, priklausomai nuo jų patirties darbu su sistema.

Kodo eilučių skaičius sistemoje yra pagrindinės metrikos daugumoje algoritmų kainų modeliuose. Tikslus kodo apimties įvertinimas yra sudėtingas projekto ankstyvoje stadijoje, nes kodo apimtis yra įtakojamas dizaino, kuris dar nėra pagamintas. Be to, kodo apimtis ne visada parodo sistemos apimtį (pasirinkta programavimo kalba įtakoja kodo eilučių skaičių, pavyzdžiui programuojant Java gali reikėti daugiau kodo eilučių, nei C kalba).

Naudojant šiuos modelius naudinga sukurti grupę įvertinimui (blogiausia, tikėtina ir geriausia), tai geriau nei vienas įvertinimas, ir pritaikyti kainų formules visai šiai grupei. Įvertinimas yra labiausiai tikėtinas, kuomet yra suprantamas programinės įrangos tipas, kuomet naudojamas patikrintas kainos modelis, pasirinkta žinoma programavimo kalba.

Tikslumas atliekant apskaičiavimus naudojant algoritminės kainos modelį, priklauso nuo informacijos apie turimą sistemą. Kuo toliau pažengęs kūrimo procesas tuo apimties vertinimas tampa tikslesnis. Šis apskaičiavimo neapibrėžtumo procesas pateikiamas 4 pav.:



4 pav. Vertinimo neapibrėžtumas

Jei pirminiu įvertinimu reikia x mėnesių pastangų, šis diapazonas gali būti nuo $0,25x$ iki $4x$ (priklausomai nuo jos sudėtingumo), kai sistema tik pasiūlyta kurti. Siaurėjimas vyksta projektavimo metu, paveikslas atspindi stambių programinės įrangos projektų patirtį, beabejo, prieš pateikiant užsakovui programinę įrangą, įvertinimas yra tiksliausias.

Yra plati įvairovė skirtingų algoritminių modelių, labiausiai paplitę ir naudojami yra *Boehm COCOMO* [BOEHM-81], *Putmano SLIM* ir *Albrecht* funkcinių taškų [ALBR-83] [4]. Abstrakčiai vertinant visi algoritminiai modeliai yra panašūs, bet naudojama skirtingas parametrų vertes [5].

2.4.1.1. COCOMO modelis

Vienas iš algoritminių modelių – *COCOMO* modelis.

Tai empirinis modelis paremtas apibendrinta informacija apie praktiškai įvykdytus projektus. Šie duomenys yra analizuojami, siekiant atrasti formules, kurios geriausiai tiktų stebėjimui. Šios formulės sujungia sistemos apimtį ir produktą, projekto ir komandos faktorių pastangas projektuojant sistemą.

Galima pasirinkti naudoti *COCOMO* modelį dėl keleto priežasčių:

- Gerai dokumentuotas, prieinamas ir palaikomas viešam naudojimui, taip pat palaikomas komercinių įrankių.
- Plačiai naudotas ir skaičiavimai atlikti daugelyje sričių organizacijose.
- Ilgas egzistavimas, nuo pirmo paruošimo 1981 metais iki paskutinės versijos *COCOMO II*, išleistos 2000 metais.

COCOMO modeliai yra visapusiški, su daugybe parametrų. Modeliai yra kompleksiniai, todėl bus pateiktos tik pagrindinės charakteristikos.

Pirma versija *COCOMO* modelio (*COCOMO 81*) yra trijų lygių modelis, kur lygiai atitinka analizės detales skaičiuojant kainą. 1 lentelėje yra pateikiama bendra *COCOMO 81* formulė skirtingų tipų projektams.

1 lentelė. *COCOMO 81* modelis

Projekto sudėtingumas	Formulė	Aprašymas (kada pritaikyti)
Paprastas	$PM = 2.4(KDSI)^{1.05} \cdot M$	Gerai suprantamos programos, kuriamos mažų komandų.
Vidutinis	$PM = 3.0(KDSI)^{1.12} \cdot M$	Daugiau kompleksinių projektų, kur komandos nariai gali turėti ribotą patirtį su susijusiomis sistemomis.
Įterptinis	$PM = 3.6(KDSI)^{1.20} \cdot M$	Kompleksiniai projektai, kur programinė įranga yra dalis labai susietų aparatūrinės, programinės įrangos, reguliavimo ir operacinių procedūrų.

Pirmas lygis pateikia apytikrį įvertinimą. Antras – modifikuotą, priklausomai nuo projektų, procesų koeficientų. Trečias yra detaliausias lygis, apskaičiuojant skirtingas projekto fazes. Koeficientas M įtakoja produkto, projekto ir komandos charakteristikas.

COCOMO 81 sudarytas remiantis prielaida, kad naudotas kaskadinis proceso modelis ir kad programos bus kuriamos nuo pat pradžios.

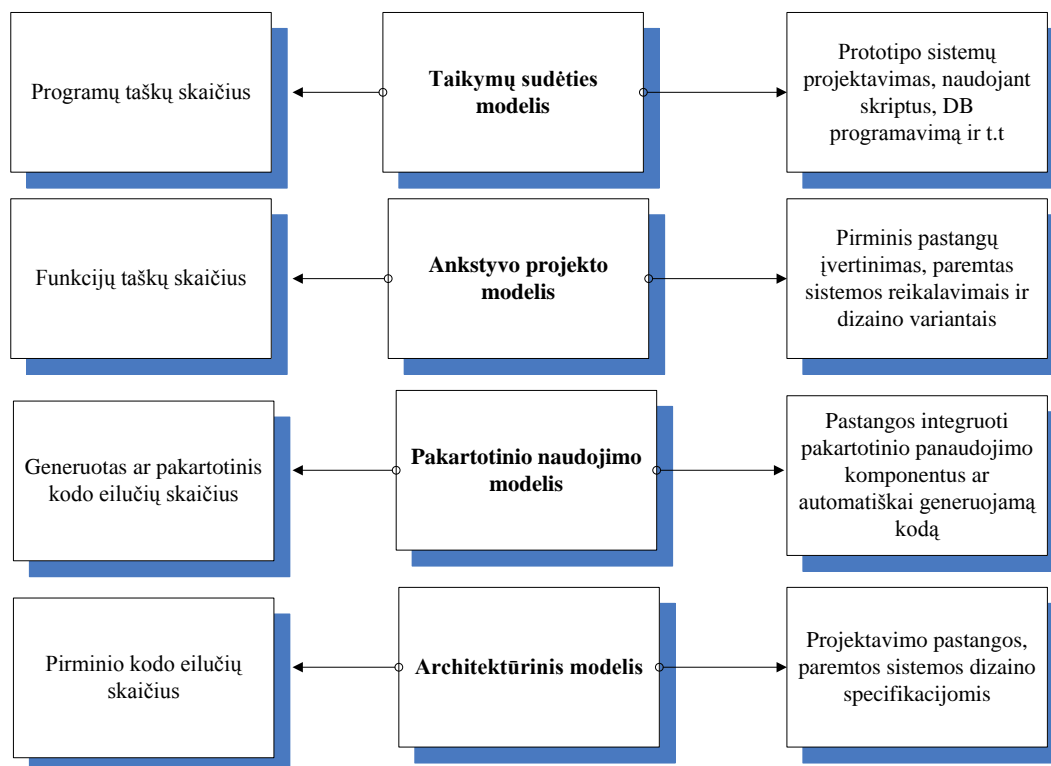
Ilgai tobulintas nuo pradinės versijos (*COCOMO 81*) per tarpines iki *COCOMO II*. Jis apima programinės įrangos projektavimo prototipą, projektavimo komponentų sudarymą, naudoja duomenų bazės programavimą. *COCOMO II* integruoja skirtingus programų kūrimo būdus, kurie atsirado programų inžinerijoje. *COCOMO II* įkomponuoja keletą dalinių modelių, kurie detaliau vertina programas, tai pateikta 2 lentelėje:

2 lentelė. Daliniai modeliai *COCOMO II*

Modelis	Formulė	Aprašymas
Taikymų sudėties modelis	Formulė $PM = \frac{NAP \cdot \left(1 - \frac{\%reuse}{100}\right)}{PROD}$	Naudojamas, kai programos sudaromos iš jau egzistuojančių dalių. Modelis supaprastintai įvertina, jog nėra papildomų pastangų susijusių su pakartotinu panaudojamumu.
	pastangų apskaičiavimo sistemos prototipams yra: PM – pastangų įvertinimas asmens – mėnesiais. NAP - pateikiamos sistemos programų taškų suma. %reuse – suminis įvertinimas pakartotinio kodo projektavime. PROD – objekto taško produktyvumas.	

<p>Ankstyvo projekto modelis</p>	$\text{Effort} = A \cdot \text{Size}^B \cdot M$ <p>Boehm siūlo koeficientą A laikyti lygiu 2.94. Komponentė B atspindi padidėjusių pastangų reikalingumą išaugus projekto apimčiai, gali varijuoti nuo 1.1 iki 1.24, priklausomai nuo projekto naujoviškumo, projektavimo lankstumo, rizikos, tarpusavio ryšio tarp projektavimo komandos, procesų brandos lygio organizacijoje. Koeficientas M paremtas 7 projektų rinkinio supaprastinimu ir procesų charakteristikomis, kurie įtakoja įvertinimą.</p> <p>$M = \text{PERS} \cdot \text{RCPX} \cdot \text{RUSE} \cdot \text{PDIF} \cdot \text{PREX} \cdot \text{FCIL} \cdot \text{SCED}$</p> <p>PERS – asmeniniai gebėjimai. RCPX – projekto patikimumas ir sudėtingumas. RUSE – reikalingumas pakartotinio panaudojamumo. PDIF – platformos sudėtingumas. PREX – asmeninė patirtis. FCIL – palaikymo priemonės. SCED – tvarkaraštis.</p>	<p>Naudojamas, kai turimi reikalavimai, bet projektavimas dar neprasidėjo. Įvertinimas šiame etape yra patiekiamas algoritminių modelių standartine formule.</p>
<p>Pakartotinio naudojimo modelis</p>	$PM_{\text{Auto}} = \frac{\left(\text{ASLOC} \cdot \frac{\text{AT}}{100} \right)}{\text{ATPROD}}$ <p>AT – procentas adaptuoto kodo, kuris yra automatiškai generuojamas. ATPROD – inžinierių produktyvumas, generuojant tokį kodą. ASLOC – kodo eilučių skaičius, komponentuose, kurie buvo adaptuoti.</p>	<p>Naudojamas pastangų skaičiavimui integruojamų pakartotinio naudojimo komponentų. Formulė pastangų įvertinimo (įvertinimas generuojamo kodo).</p>
<p>Architektūrinis modelis</p>	$PM = A \cdot \text{Size}^B \cdot M$ <p>Koeficientų reikšmės pateiktos ankstyvojo projekto modelyje</p>	<p>Naudojamas, kai suprojektuota architektūra ir yra daugiau informacijos apie sistemą. Įvertinimas šiame etape yra patiekiamas algoritminių modelių standartine formule.</p>

COCOMO II modelių naudojimas:



5 pav. COCOMO II modeliai

Projekto vadybininkai taip pat turi vertinti projekto trukmę ir personalo apkrovimą. Kalendorinis laikas gali būti vertinamas *COCOMO II* pagal formulę:

$$TDEV = 3 \cdot (PM)^{[0.33 + 0.2 \cdot (B - 1.01)]}$$

PM – apskaičiuotos pastangos ir B – eksponentinis faktorius, aptartas anksčiau. Tai prognozuoja nominalią projekto trukmę.

COCOMO II modelis prognozuodamas reikiamas pastangas įvertina projekto, produkto, personalo ir aparatūros atributus, taip pat įtraukia projektavimo tvarkaraščių įvertinimą.

2.4.1.2. *SLIM* modelis

SLIM (Programinės įrangos gyvavimo raidos ciklo valdymas) - vienas iš anksčiau taikytų algoritminių metodų didelių projektų sąnaudoms apskaičiuoti. Jis remiasi *Norden / Reyleigh* funkcija ir vadinamąja technologine konstanta *C*, kuri priklauso nuo naudojamų programavimo priemonių, projekto unikalumo, didumo, komandos patirties, kokybės standartų ir kt. *SLIM* įgalina programinės įrangos kainos vertintoją įvykdyti šias funkcijas [10]:

- Patikra: puikus modelio suderinimas, kad būtų galima reprezentuoti vietinės programinės įrangos išsivystymo aplinką, interpretuojant buvusių projektų duomenų bazę.
- Sukūrimas: programinės įrangos sistemos informacinis modelis, programinės įrangos charakteristikų rinkimas, asmeniniai, kompiuterio atributai ir t.t.
- Programinės įrangos apimties nustatymas: *SLIM* naudoja automatizuotą kodo eilučių įvertinimo techniką.

SLIM modelio naudoja formulę [8]:

$$K = \left[\frac{LOC}{\left(\frac{4}{C \cdot t^3} \right)} \right] \cdot 3$$

LOC – kodo eilučių skaičius, K – suma gyvavimo ciklo pastangų (darbo metais), t – projektavimo laikas (metais). C – technologijos konstanta, sujungianti įrankius, kalbas, metodologijas, procedūras ir t.t. Technologijos konstantos vertė gali varijuoti nuo 610 iki 57314. Putman rekomenduoja skaitmenis skirtingiems tipams projektų:

- Realus laiko integravimas (*Real-Time Embedded*) 1500
- Paketų projektavimas (*Batch Development*) 4894
- Palaikymas ir organizavimas (*Supported and Organised*) 10040

Naudojant šį modelį dažnai gaunama, kad vykdant mažesnius projektus bendra sąnaudų suma gaunasi mažesnė, nei vykdant vieną didelį projektą. Todėl reikia atsargiai jį taikyti, kai didelis projektas yra padalinamas į mažesnius struktūrinius vienetus.

Jis yra tinkamas programinės įrangos projektavimui, kurie atitinka tokius reikalavimus kaip:

- Programinės įrangos apimtis didesnė nei 5000 eilučių.
- Patirtis didesnė nei 1.5 metų.
- Daugiau kaip 6 mėnesiai projektavimo laiko.

2.4.1.3. Funkcinių taškų modelis

A. Albrecht suprojektavo funkcinių taškų metrikas ir funkcinių taškų analizės metodą, kad įvertinti programinės įrangos apimtį kaip alternatyvą kodo eilučių skaičiui. Tai metodas, įvertinantis pastangas, apskaičiuoja sistemos funkcionalumą. Buvo išskirti funkcinių taškų privalumai prieš kodo eilučių skaičių [11]:

- Ankstesnis įvertinimas (programinės įrangos reikalavimų analizės ir parengiamojo projektavimo metu).
- Įvertinimas atliekamas projekto komandos, kuri nesusijusi su technine dalimi.

- Nepriklausomas nuo vykdomos kalbos ir projektuotojų patirties.

Metodas turi identifikuoti ir apskaičiuoti unikalios funkcijos tipus [4]:

- Įvestis.
- Išvestis.
- Užklausa.
- Failai.

Susitelkdamas ties reikalavimų specifikacijos dokumentu, vertintojas gali apskaičiuoti sistemos funkcionalumą, kuri buvo projektuota, identifikuojant funkcijos tipus. Funkciniai taškai skaičiuojami dauginant kiekvienos charakteristikos kiekį iš svorio ir viską sumuojant.

Funkcinių taškų skaičiavimas modifikuojamas priklausomai nuo projekto sudėtingumo. Pagal funkcinis taškus gali būti apskaičiuotas programos eilučių kiekis [5]:

- Eilučių kiekis = Nuo naudojamos kalbos priklausantis koeficientas * funkcinių taškų kiekio;
- Nuo naudojamos kalbos priklausantis koeficientas gali kisti nuo 200-300 assembleriui iki 2-40 ketvirtos kartos kalboms.

Bendras techninis sudėtingumo faktorius gali būti apskaičiuojamas naudojant formulę:

$$TCF = 0.65 + \frac{(\text{sum of factors})}{100}$$

Galutinis funkcijos taškų skaitmuo gali būti apskaičiuojamas:

$$FP = UFC \cdot TCF$$

TCF gali kisti nuo 0.65 iki 1.35, kadangi skaitmuo 0.65 duos rezultata, kuomet sudėtingumo faktorius neturi įtakos, o 1.35 - priešingai, pabrėš įtaką. Dėl to, jei specifikacijos reikalavimai parodys, kad sistema yra palyginti paprasta, TCF bus žemesnis negu 1, todėl, kad sudėtingumo faktoriai turės mažą įtaką, priešingu atveju – daugiau nei 1.

Funkcinių taškų skaičiavimas labai subjektyvus ir priklauso nuo vertintojo. Automatinis funkcinių taškų skaičiavimas labai sudėtingas.

3 lentelė. Algoritminių kainų modeliavimo modelių įvertinimas

Modelis	Privalumai	Trūkumai
<i>COCOMO</i>	<ul style="list-style-type: none"> • <i>COCOMO</i> yra aiškus, galima matyti kaip jis veikia, skirtingai nei kiti modeliai, tokie kaip <i>SLIM</i>. • Tvarkyklės (drivers) yra ypatingai padedančios įvertinimui, suprasti skirtingų faktorių poveikį, kurie 	<ul style="list-style-type: none"> • Sudėtinga tiksliai įvertinti apimtį <i>KDSI</i> (pateikiamos pirminės instrukcijos) projekto pradžioje, kai daugiausia įvertinimo pastangų reikalaujama.

	įtakoja projekto kainas.	<ul style="list-style-type: none"> • Ypatingai jautri vieta – praradimas klasifikacijos projektavime. • Sėkmė didžia dalimi priklauso suderint modelį su organizacijos poreikiais, naudojant praeities duomenis, kurie ne visada yra prieinami.
<i>SLIM</i>	<ul style="list-style-type: none"> • Naudoja linijinį programavimą, kad atsižvelgti į kainos ir pastangų apribojimus projektavime. 	<ul style="list-style-type: none"> • <i>SLIM</i> įvertinimas yra ypatingai jautrus technologijos faktoriui. • Nepritaikomas smulkiems projektams.
Funkcinių taškų	<ul style="list-style-type: none"> • Svarbūs duomenys yra pasiekiami ankstyvoje projekto stadijoje, tereikia detalios specifikacijos. • Savarankiška kalba. • Savarankiškas išdėstymas. • Gana tiksli. 	<ul style="list-style-type: none"> • Subjektyvus apskaičiavimas. • Sudėtinga automatizuoti. • Ignoruojama išvesties kokybė. • Orientuota į tradicines programas.

Buvo apžvelgtos klasikinės metodikos, skirtos projektų kainos apskaičiavimui. Kadangi daugumos algoritminių kainų modeliuose pagrindinė metrika yra kodo eilučių skaičius (kaip nagrinėtas *SLIM* modelis), tikslus kodo apimties įvertinimas yra sudėtingas projekto ankstyvoje stadijoje, be to tokios metrikos nevertina kokybės. Todėl apskaičiuojant programinės įrangos apimtį yra naudinga turėti metrikas, kurios nepriklauso nuo naudojamos technologijos, kadangi skirtingos kalbos turi skirtingas išraiškas (pvz.: C++ eilučių skaičius nebus lygus Asemblerio eilučių skaičiui). Be to, jeigu reikalavimai pastoviai keičiasi tai eilučių kiekio skaičiavimas nėra prasmingas, kadangi programos nėra statinės. Šiuo aspektu funkcinės apimties matavimas yra naudingesnis, kadangi funkcinės apimties metodai leidžia išmatuoti programinės įrangos funkcionalumo apimtį [9].

Antra pagal dažnumą naudojama metrika yra funkcionalumas. Ši metrika įvertina programinės įrangos apimtį iš vartotojo pozicijų, vartotojo pozicijos yra gerokai svarbesnės už projektuotojo pozicijas (kodo eilučių skaičiaus metrikas). Kaip pradininkas šio funkcinės taškų metodo yra minėtas Albrecht, kurio originalaus metodo variantai buvo vėliau tobulinami, pritaikant naujas funkcinės taškų įvertinimo technikas, tačiau pagrindinė metodo idėja išliko, tai yra įvertinimas apimties funkcionalumas, kuri pristatoma vartotojui [11].

2.4.2. Funkcinės apimties matavimo modelių standartai

Per praėjusius 3 dešimtmečius įvairūs programinės įrangos apimties įvertinimo metodai buvo tobulinti, projektuoti ir pritaikyti skirtingose projektavimo fazėse [11].

Dėl pakitimų metoduose, kurie paremti funkcionalumo metrikomis, be bendro fundamentalių koncepcijų nustatymo, natūraliai išsivystė nesuderinamumai ir *ISO* (tarptautinė standartų organizacija) pradėjo darbą, kad funkcinės apimties įvertinime nustatyti bendrus principus šiuose metoduose. Standartas prisidėjo prie nuoseklaus funkcinės apimties matavimo interpretavimo principų. Funkcinės apimties matavimo metodai yra sukurti, siekiant patikimai įvertinti funkcinę apimtį, apibrėžus funkcinis reikalavimus, kuomet programinės įrangos reikalavimų specifikacija yra paruošta.

Toliau bus apžvelgti 4 *ISO* standartai: *IFPUG Function Point Analysis* (ISO/IEC 20926, 2003), *Mark II Function Point Analysis* (ISO/IEC 20968, 2002), *COSMIC Full Function Points* (*COSMIC FFP*, ISO/IEC 19761, 2003) ir *FiSMA* (ISO/IEC 29881, 2008), apibrėžiant metodo plėtojimą, esmę, kiekvieno iš jų siūlomą įvertinimo procesą.

2.4.2.1. Albrecht / IFPUG Function Point Analysis (Albrecht / IFPUG) standartas

Metodo plėtojimas

Pirminis funkcinų taškų analizės metodas pasiūlytas 1979 metais buvo palyginti paprastas. Aprašytos funkcijos (žiūrėti 24 - 25 psl.) 1983 metais papildytos modifikavus metodo versiją, įvestas 3 lygių sudėtingumo funkcijos, taisyklės sudėtingumo įvertinimui pagal funkcijos tipą ir atitinkamai svorių lentelė. Failų tipai suskaidyti į du potipius: vidinis loginis failas ir išorinis sąsajos failas. Pagrindiniai funkciniai komponentai šioje versijoje vadinami [11]:

- Išorinės įvestis (failo vardai).
- Išorinės išvestis (pranešimai, žinutės).
- Išorinės užklauskos (interaktyvios įvestys, reikalaujančios atsako).
- Vidiniai sąsajų failus (failai bendrai naudojami su kitomis programinės įrangos sistemomis)
- Vidiniai loginiai failai (nematomi sistemos išorėje).

1985 metais *IBM* grupė (*GUIDE*) pataisė Albrecht pagrindinius apibrėžimus, siekiant nustatyti aiškumą ir apibrėžti funkcinio taško analizės taisykles, nustatant funkcinio sudėtingumo vertinimą (žemas, vidutinis, aukštas) ir 5 funkcinis tipus, pagal kuriuos sukurtos matricos (viena – loginiams failams, ir dvi – transakcijoms su iš anksto numatytais intervalo vertėmis, skirtomis vertinimui. Tai leidžia atlikti ekspertams nuoseklų matavimą.

1986 metais tarptautinė funkcinų taškų vartotojų grupė (*IFPUG*) tapo projektavimo autoritetas, nuo tada *IFPUG* išplėtė funkcinio taško skaičiavimo taisykles ir Albrecht aprašymą. *ISO/IEC* standartu tampa 2003 metais.

Metodo esmė

IFPUG FPA pagrindiniai funkciniai komponentai yra klasifikuojami iš galutinio vartotojo pozicijos, tokie kaip: išorinės išvestis, įvestis, užklausos, sąsajos failai ir loginiai vidiniai failai. Priskiriami svoriai kiekvienam funkciniai komponentui, šie svoriai sumuojami ir rezultatų vertės koreguojamos, naudojant vertės koregavimo faktorių, kad pateikti pakoreguotą funkcinio taško dydį. Jis yra paremtas 14 pagrindinių sistemos charakteristikų, kurios įvertina pagrindinį apskaičiuojamos programos funkcionalumą.

2.4.2.2. Mark II Function Point Analysis (Mark II FPA) standartas

Metodo plėtojimas

Britiškas *Mk II FPA* metodas projektuotas nuo 1988 metų C. Symons, siekiant išspręsti trūkumus įprastinių *FPA* metodų. Dabar metrikų praktikos komitetas (*Metrics Practices Committee - MPC*) iš Didžiosios Britanijos programinės įrangos metrikų asociacijos (*UK Software Metrics Association - UKSMA*) yra projektavimo metodo autoritetas (1998 m.). ISO/IEC standartu tampa 2002 metais.

Metodo esmė

Mk II FPA siekia įvertinti informacijos procesą. Šis metodas mato sistemą kaip rinkinį loginių transakcijų ir skaičiuoja programinės įrangos funkcinę apimtį, apskaičiuojant: įvedimo duomenų elementų tipus, duomenų objektų tipų nuorodas ir išvedimo elementų duomenų tipus, kiekvienai loginei transakcijai. Buvo suprojektuota siekiant įvertinti verslo informacines sistemas, kaip ir *Albrecht/IFPUG FPA*. Kitų sričių programinė įranga, kokia kaip mokslinė ar realaus laiko įmanoma, tačiau gali reikėti tam tikrų metodo modifikacijų.

2.4.2.3. COSMIC Full Function Points (COSMIC FFP) standartas

Metodo plėtojimas

1999 metais *COSMIC FFP* metodo antroji versija buvo išleista bendro programinės įrangos tarptautinio įvertinimo konsorciumo (*Common Software Measurement International Consortium - COSMIC*). (Ankstesnių metodo versijų apžvalgą žiūrėti 20 - 23 psl.) Buvo nusistatytas tikslas suprojektuoti naują metodą, kuris įvertintų programinės įrangos funkcinę apimtį abejoms: verslo ir realaus laiko ir jų mišrioms programinėms įrangoms (2003 metais). ISO/IEC standartu tampa 2002 metais.

Metodo esmė

COSMIC FFP metodas buvo suprojektuotas įvertinti programinės įrangos funkcinę apimtį, paremtą funkciniais vartotojų reikalavimais. Į šiuos funkcinis reikalavimus

neįtraukiami kokybės ir techniniai reikalavimai. Programinės įrangos funkcinė apimtis yra išmatuojama remiantis keturiais pagrindiniais funkcinių komponentų tipais, tokiais kaip:

- Įėjimas.
- Išėjimas.
- Skaitymas.
- Rašymas.

2.4.2.4. Functional Size Measurement FSM Method (FiSMA) standartas

Metodo plėtojimas

Tai yra brandus ir kruopštus *FSM* metodas. Nuo jo pirmo pristatymo kaip „*Laturi* metodas“, prieš 15 metų, *FiSMA* plėtojosi, remiantis moksliniais tyrimais ir vartotojų atsiliepimais. Šis metodas išvystytas Suomijos įvertinimo asociacijos (*Finnish Software Measurement Association - FiSMA*, 2004). Tai yra pagrindinis parametrų apimtį įvertinimo metodas, kuris buvo suprojektuotas pritaikyti visoms programinės įrangos tipams. Šis metodas buvo numatytas projektuoti vietoj ankstesnio *FSM Experience 2.0 Function Point Analysis* metodo. ISO/IEC standartu tampa 2008 metais.

Metodo esmė

Panašiai kaip kiti metodai, paremti „funkcionalumu“, *FiSMA FSM* taip pat paremtas funkciniais vartotojų poreikiais. *FiSMA* metodas veikia geriausiai, kuomet yra baigtas funkcinių vartotojo reikalavimų ir paslaugų sąrašas, metodas atlieka apimtį valdymą ir jį efektyviai keičia, patikimai ir palyginti lengvai suprantamas galiniam vartotojui. Pagrindinis panaudojamumas yra įvertinimas ir produktyvumo analizė, įrodytas naudingumas projektų planavime, stebėjime, kontroliavime [12]. Skirtumas tarp apžvelgtų metodų yra toks, jog *FiSMA* yra orientuota į paslaugas, o ne į procesą. Į procesą orientuotų metodų, visi funkciniai procesai, palaikomi programinės įrangos, turi būti identifikuojami. Į paslaugas orientuotame metode, visos skirtingos paslaugos, aprūpinamos programine įranga, turi būti taip pat identifikuojamos. *FiSMA* identifikuoja 7 skirtingas klases:

- Interaktyvios galinio vartotojo navigacijos ir užklausų (q).
- Interaktyvios galinio vartotojo įvesties (i).
- Neinteraktyvios galinio vartotojo išvesties (o).
- Sąsajos paslaugos kitoms programoms (t).
- Sąsajos paslaugos iš kitų programų (f).
- Duomenų saugyklos (d).
- Algoritminės ir apdorojimo (a).

Po kiekvienos paslaugos identifikavimo skaičiavimo taisyklės yra pritaikomos, siekiant nustatyti kiekvienos paslaugos apimtį. Po to bendra funkcinė apimtis yra suskaičiuojama, sumuojant visų paslaugų apimtį.

Kadangi kiekvienas metodas turi skirtingus tipus, funkcinė apimtis yra skaičiuojama pagal kiekvieno metodo skirtingus vienetus.

2.4.2.5. Bendras funkcinės apimties matavimo metodų procesas ir skirtumai

Iš kitų apimties išmatavimo/įvertinimo metodų, *FSM* metodai yra plačiai naudojami. Nors visi *FSM* metodai įvertina apimtį, remiantis funkcionalumu, pagrindinis skirtumas tarp šių technikų kilo iš to, kas yra skaičiuojama ir kaip tai atliekama.

„*IFPUG 4.1*“ pagrindiniai funkciniai procesai, yra klasifikuojami iš galinio vartotojo pozicijos kaip transakcijų ir duomenų funkcijų tipai. Transakcijų funkcijų tipai yra taip pat suskirstyti į kategorijas: išorinę įvestį, išorinę išvestį, išorines užklausas. Datos funkcijos tipai: išorinius sąsajos failus, vidinius loginius failus. Priklausomai nuo skaičiaus duomenų elementų tipų ir įrašų elementų tipų, kiekvienas tipas yra suklasifikuojamas ir priskiriamas į „paprastą“, „vidutinį“, „sudėtinį“. Šios vertės yra sumuojamos gauti visą funkcinę apimtį.

„*Mk II FPA 1.3.1*“ tikslas apskaičiuoti informacijos proceso apimtį. Šio metodo pagrindiniai funkciniai komponentai yra loginės transakcijos. Jos apibrėžiamos kaip „žemiausias“ lygis verslo procesų, kurie palaikomi programinės įrangos. Loginė transakcija identifikuojama suskaidant kiekvieną funkcinį reikalavimą į elementarius komponentus. Kiekviena loginė transakcija turi 3 dedamąsias: įvestį, procesą ir išvesties komponentus. Funkcinė apimtis kiekvienai loginei transakcijai yra vertinama dauginant kiekvieno komponento dydį iš svorio faktoriaus, kuris yra nustatomas pagal rinkos vidurkį, atitinkamai įvertinant analizės, projektavimo, programavimo ir testavimo pastangas. Tuomet funkcinė apimtis kiekvienos loginės transakcijos yra susumuojama siekiant apskaičiuoti visos sistemos funkcinę apimtį.

COSMIC FFP metode kiekvieni funkciniai reikalavimai yra suskaidyti į elementarius komponentus, vadinamus funkciniais procesais. Kiekvienas iš funkcinų procesų apima subprocesų rinkinį, kuris atlieka taip pat duomenų judėjimą ar apdorojimą. Šis metodas nėra sukurtas apskaičiuoti sistemas, kurių duomenų tipai orientuoti į platų duomenų apdorojimą, tokį kaip mokslinė programinė įranga. Egzistuoja 4 duomenų judėjimo rūšys, jų tipai: įvestis, išvestis, išėjimas, skaitymas ir rašymas. Funkcinė apimtis kiekvieno funkcinio proceso yra nustatoma apskaičiuojant įvestį, išvestį, išėjimą, skaitymą ir rašymą kiekvienam funkciniam procesui. Tuomet sistemos visų funkcinų procesų bendras apskaičiuotas skaičius yra funkcinė apimtis.

„*FiSMA 1.1*“ skaičiavimo parametrų teisingumas gali būti apskaičiuotas remiantis koreliacija tarp funkcinės apimties ir pastangų pagal panašią aplinką, technines aplinkybes ir kokybės reikalavimus. Vartotojo reikalavimai gali būti apibrėžti kaip funkciniai – ką sistema atlieka ir nefunkciniai – kaip sistema turi atlikti (įtraukiant kokybės reikalavimus). Priešingai nei kiti metodai, kurie orientuoti į procesą, *FiSMA* orientuota į funkcinės paslaugas. Išskiriami tokie 7 skirtingos duomenų klasės (pateikiami trumpiniai, plačiau žiūrėti 29 psl.): q, i, o, t, f, d, a. Kiekviena klasė skaidoma į tipus. Iš viso yra 28 duomenų tipai. Priešingai nei prieš tai aptarti standartai „*FiSMA 1.1*“ yra bendras, parametrinis funkcinės apimties apskaičiavimo metodas visiems programinės įrangos tipams.

Skirtumai tarp *FSM* metodų yra apibendrinti **10.4. Priede. Skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų**, atsižvelgus į sekančius kriterijus:

- Funkcinės srities pritaikomumas. *ISO* standarte funkcinė sritis yra apibrėžiama kaip programinės įrangos klasė, paremta funkcinų reikalavimų charakteristikomis. Standartas reikalauja, kad *FSM* metodas turi aprašyti funkcinę sritį (-is), kuriai *FSM* gali būti pritaikytas.
- Matavimo vienetas. *ISO* standartas reikalauja, kad vienetai, kuriais išreikšta funkcinė apimtis, turi būti apibrėžti.
- Matavimo pozicija. Iš kieno pozicijos išmatuojama funkcionalumo apimtis.
- Pagrindiniai funkciniai komponentai. Tai elementarus vienetas funkcinų reikalavimų, apibrėžtų *FSM* metodo išmatavimo tikslais.
- Tipų dedamoji. Siekiant priskirti skaitines vertes kiekvienam funkciniam komponentui, kai kurie *FSM* metodai identifikuoja ir apskaičiuoja dedamąsias, iš kurių susideda tipai.
- Kiekvienos dedamosios funkcionalumas. Apibrėžimai paimti iš išmatavimų metodų vadovų.
- Pagrindinio apskaičiavimo išvedimas. Kriterijai, kurie kiekvieno metodo gali būti apskaičiuojami gauti funkcinę apimtį.
- Funkcinio sudėtingumo svoris.
- Susijusios pagrindinių skaičiavimų funkcinės apimties įvertinimo vertės. Kai kuriems metodams suteikiami svoriai, arba ne, skaičiuojant funkcinę apimtį.

Apžvelgus darbo **10.4. Priedo. Skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų** aprašus, galima matyti jų skirtumus, šiems metodams būdingas charakteristikas. Kaip galima pastebėti, metodo pasirinkimas priklauso nuo įvairių aspektų,

tokių kaip: kuriamos sistemos, jos apimties, keliamų funkcinių reikalavimų, reikalaujamo apimties tikslumo, vartotojo ir panašių aspektų.

Pasirinkus realizuoti bet kurią iš apžvelgtų metodų, jie bus pakankamai geri skaičiuojant darbų apimtį. Tačiau apimties skaičiavimas nėra vienintelė dimensija įvertinimui, reikalingas taip pat įvertinimas kitų metrikų, kurios padėtų suvokti faktorius, kurie įtakoja projektavimą. Situacijos analizė, siekiant suprasti aplinkos faktorius, kurie galėtų įtakoti komandos produktyvumą, taip pat gali padėti atlikti tikslesnius apskaičiavimus. Be to, yra kitos problemos, tokios kaip reikalavimų valdymas. Yra sudėtinga atlikti įvertinimus, kai reikalavimai suformuoti blogai. Funkcinės apimties įvertinimas ir funkciniai taškai yra dalis apimties įvertinimo, jie užtikrina, kad reikalavimai tikslumą. Jei neįmanoma įvertinti funkcinės apimties programinei įrangai, negalima jos projektuoti [13].

Literatūroje, analizuojant metodo pasirinkimą, siūloma atsižvelgiant į situacijos analizę įvertinti 4 produktyvumo faktorius. Pirmasis apima projekto organizuotumą ir vartotojo įtraukimą į tai. Antrasis faktorius apima viso projekto valdymą kaip visumą. Trečiasis – kokie faktoriai svarbūs siekiant tikslo. Ketvirta grupė apima komandos faktorių, projekto valdymo patirtį.

4 lentelėje yra pateikiama apibendrinta *FSM* standartų lyginamoji analizė.

4 lentelė. *FSM* apibendrinta standartų lyginamoji lentelė

FSM standartai	Funkciniai komponentai	Įvertinimo apskaičiavimo sudėtingumas	Apribojimai IS sistemų įvertinimui	Laiko įvertinimas pagrindinis kriterijus	Standarto prieinamumas vartotojui	Standarto detalumas ir suprantamumas	Naujumas
<i>IFPUG FPA</i>	+	-	+	-	-	-/-	2004 m.
<i>Mk II FPA</i>	+	-	+	-	-	-/+	2002 m.
<i>COSMIC FFP</i>	+	+	-	-	+	+/-	2002 m.
<i>FiSMA FSM</i>	+	-	-	+	+	+/+	2008 m.

Kiekvieno nagrinėto standarto metodika remiasi funkciniais komponentais (klasėmis ar tipais, priklausomai nuo konkretaus standarto). Šių standartų klasės skirtos vartotojų funkcinių reikalavimų sisteminiui. Šios klasės smulkiau yra aprašomos 2.5.3 skyrelyje „*FSM* (funkcinės apimties matavimo) modelių standartai“, detalesnis klasių skirstymas pateiktas priedo 1 lentelėje. Iš pateiktų duomenų galima teigti, jog *FiSMA FSM* standartas turintis 7 klases, kurios suskirstytos į 28 tipus (tuo tarpu *IFPUG FPA* – 2 klasės, 5 tipai, *Mk II FPA* - 1 klasė, 1 tipas, *COSMIC FFP* – 1 klasė, 4 tipai) yra itin detalus, kas įtakoja tikslesnį

funkcinio reikalavimo įvertinimą. Detalesnis vertinimas gali padėti tiksliau įvertinti, mažiau nukrypstant nuo realaus atlikimo laiko, kadangi reikalavimas išskaidomas į smulkesnes dalis (tipus).

COSMIC FFP standartas lyginant su nagrinėtaisiais standartais pasižymi detaliu įvertinimo mechanizmu, tačiau pateikiamos formulės, pritaikomumas yra sudėtingas, kadangi metodas apima ne tik funkcinio reikalavimo apimties įvertinimą, tačiau ir kainos variavimą, vykdytojų motyvavimą. *FiSMA FSM* standartas orientuojasi tik į funkcinio reikalavimo apimties vertinimą, standarto siauresnė metodika įtakoja mažiau painų įvertinimo apskaičiavimą. Taip pat *FiSMA FSM* detalesni tipai labiau struktūrizuoja įvertinimo apskaičiavimą, kadangi sumuojama kiekviena programinės įrangos dalis, ir visi funkciniai dydžiai sudedami kartu.

Standartai *IFPUG FPA* ir *Mk II FPA* yra apribojami vertinant informacines sistemas, t.y šie standartai skirti vertinti verslo sistemos (*CRM* sprendimai / Verslo valdymo sistemos). *COSMIC FFP* skirtas verslo ir realaus laiko ir jų mišrioms programinėms įrangoms vertinti, tuo tarpu *FiSMA FSM* yra vienas naujausių standartų (2008 m.), buvo suprojektuotas pritaikyti visoms programinės įrangos tipams (7 klasės apimančios platų spektrą *IS*).

Kadangi *FiSMA FSM* standartas yra vienas naujausių, standartas yra prieinamas visiems vartotojams, priešingai nei yra su *IFPUG FPA*, *Mk II FPA*, kuriems trūksta informatyvumo, standartų atsisivertinimui reikalingos specialios registracijos, metodikos nėra detalios, suprantamos. *FiSMA FSM* kūrėjai siekdami platinti metodiką pateikia standarto detalius tipų įvertinimo pavyzdžius, ko trūksta *COSMIC FFP* standartui, norint jį realizuoti.

Taigi remiantis atlinkta analize *FiSMA FFP* metodika pasirinkta realizacijai, kadangi ji remiasi objektyviais kriterijais (vertinami labai aiškūs projekto elementai), plati klasių grupė, kuri įtakoja įvairių informacinių sistemų vertinimą, klasės išskirstytos į itin detalius tipus, kas įtakoja tikslesnį funkcinio reikalavimo įvertinimą. Standartas yra naujas, projektuotas/koreguotas remiantis vartotojų poreikiais, iškilusiomis problemomis.

2.5. Darbų apimties nustatymo tyrimo uždavinio formuluotė

Šiame skyriuje yra pateikiama vartotojų analizė, siekiamas sprendimas, tyrimo tikslai ir uždaviniai, rizikos faktorių analizė, rezultato kokybės kriterijai.

2.5.1. Vartotojų analizė

2.5.1.1. Vartotojų aibė, tipai ir savybės

Naujai kuriamoje projekto darbų apimties apskaičiavimo sistemoje numatomi tokie vartotojų tipai: projektų vadovas, darbų vykdytojas ir administratorius. Vartotojai bus suskirstyti į tipus, pagal kuriuos bus priskirtos tam tikros funkcijos. Projektų vadovas įkėlęs reikalavimus ir sistemai apskaičiavus jų trukmę, galės priskirti juos vykdytojams, kurie juos atliks vos faktinius funkcinių reikalavimų laikus. Administratorius valdys vartotojų duomenis, sukurdamas, šalindamas vartotoją, taip pat sukurs vartotojų teises. Siekiama, kad projektų vadovas galėtų efektyviau koordinuoti vykdytojus, stebint sistemoje atliekamų reikalavimų teorinius/faktinius, mažiau apkrauti save matematiniais apskaičiavimais. Taip pat galimybė kaupti statistinius duomenis apie prieš tai įvertintus projektus šioje sistemoje, bus pagalba projektų vadovui naujų projektų rengime.

2.5.1.2. Vartotojų tikslai ir problemos

Kuriant sistemas problema yra apskaičiuoti jų apimtį, kiek konkrečiai užtruks reikalavimų atlikimas, vertinant darbų pradžią ir įvykdymo trukmę, kadangi tai priklauso nuo įvairių aspektų.

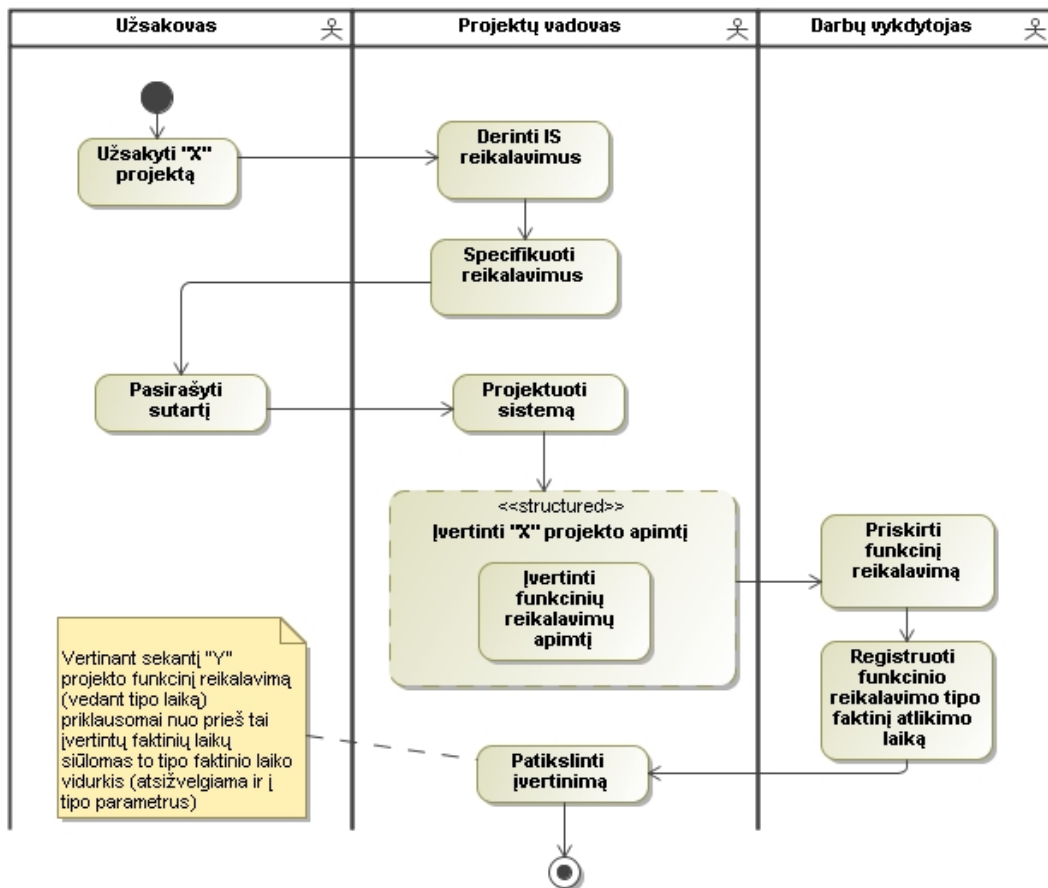
Projektų vadovų vienas iš tikslų apskaičiuoti kuo ankstyvesnėje projektavimo stadijoje projektų darbų apimtį ir tai atlikti kuo įmanoma tiksliau. Projektai yra dažniausiai įvertinami remiantis prieš tai projektuotų, panašaus pobūdžio projektų istoriniais duomenimis, tačiau apimties įvertinimas yra svarbus, kai nėra pakankamai informacijos. Pritaikius darbų apimties įvertinimo metodiką siekiama kardinalesnių pokyčių įvertinant reikalavimus.

2.5.2. Siekiamas sprendimas

Trūksta praktinių metodų, leidžiančių įvertinti projektų darbų apimtį. Analizės rezultate buvo pasirinktas *FiSMA* standartas, kuris paremtas funkciniais vartotojo reikalavimais. Tačiau apie šio standarto pritaikymą kol kas nėra atsiliepimų.

Siekiamo realizuoti standartą, kuriame bus įgyvendintas funkcinių reikalavimų įkėlimas į sistemą, taip pat funkcinių taškų automatinis paskaičiavimas pagal *FiSMA* klases, taip bus sutrumpintas projektuotojų laikas.

Taikant *FiSMA* reikia žinoti funkcinio taško vertę, kuriai nėra nustatytų normų. Todėl norint padidinti *FiSMA* patikimumą, sistemą siekiama papildyti spėjamų ir faktinių laikų kaupimu ir nuolat patikslinti darbo imlumo įvertinimus. Toliau pateiktame 6 paveiksle yra vaizduojamas bendras „X“ projekto *FiSMA* veiklos procesas, kuriame užsakovas užsako projektą, su projektu vadovu derinami reikalavimai, reikalavimai specifikuojami, pasirašoma sutartis ir projektuojama sistema.



6 pav. „X“ projekto *FiSMA* taikymo veiklos procesas

Norint įvertinti bendrą projekto apimtį, reikia įvertinti kiekvieną funkcinį reikalavimą kaip atskirą vienetą. Funkciniam reikalavimui pasirenkamas klasė, tinkamas tipas, pagal kurį suvedami atitinkami to tipo parametrai ir kiek bus teoriškai užtrukta prie reikalavimo. Įvestos funkcinių reikalavimų tipų parametrų reikšmės yra apskaičiuojamos pagal matematinės formules, sumuojant atskiras funkcinių reikalavimų reikšmes į bendrą sumą.

Kiekvienas įvertintas funkcinis reikalavimas yra priskiriamas vykdytojui, nurodant terminą, iki kada reikalavimas turi būti atliktas. Prisijungęs vykdytojas įveda faktinį laiką užgaištą prie reikalavimo tipo. Faktinius tipų laikus projektų vadovas gali stebėti „X“ projekto generuojamoje ataskaitoje, t.y galima stebėti pvz.: tipo „Prisijungimo ir atsijungimo langų“ projektų vadovo priskirtą teorinį laiką ir teorinę kainą, bei realų šio tipo faktinį laiką, faktinę

kainą, taip pat matyti funkcinis taškus. Ši informacija yra naudinga siekiant kuo tiksliau įvertinti funkcinio reikalavimo tipo teorinį laiką. Todėl kai projektų vadovas įves „Y“ projekto funkcinio reikalavimo tipo „Prisijungimo ir atsijungimo langų“ teorinį laiką, priklausomai nuo to koks buvo „X“ projekto faktinis laikas, bus siūlomas pasirenkamas preliminarus užtrukimo laikas, artimas prieš tai faktiškai įvertintam laikui. Taigi, *FiSMA* metodika bus papildyta statistinių duomenų kaupimu, t. y kuo daugiau bus registruojama sistemų, tuo tiksliau galima bus teoriškai įvertinti funkcinio reikalavimo tipo laiką.

2.5.3. Tyrimo tikslas ir uždaviniai

Tyrimo tikslas - ištirti *FiSMA* metodikos tinkamumą projekto darbų apimčiai įvertinti, realizuojant darbų apimties įvertinimo sistemą, leidžiančią nustatyti funkcinį taškų įvertinimus pagal kaupiamus faktinius duomenis, ir įvertinant projektuotojo darbo laiko sąnaudas šiai metodikai taikyti.

Tyrimo uždaviniai:

- Išanalizuoti esamas projektų darbų apimties įvertinimo metodikas.
- Atlikti *FSM* (funkcinės apimties matavimo) standartų analizę.
- Sukurti sistemą, kuri leistų skaičiuoti projekto darbų apimtį pagal *FiSMA* standartą ir kaupti faktinius duomenis funkcinį taškų vertei nustatyti bei patikslinti.
- Atlikti eksperimentą, kuris leistų įvertinti *FiSMA* metodikos efektyvumą, t.y. nustatyti žmogaus atliekamo funkcinį taškų įvertinimo laiką ir jo santykį su visa projekto trukme (ar ne per daug laiko sugaištų projektuotojai, taikydami *FiSMA* metodiką).

2.5.4. Rizikos faktorių analizė

Pagrindinis rizikos faktorius yra blogai apskaičiuoti funkciniai reikalavimai. Siekiant išvengti šios rizikos bus remiamasi daugeliu literatūros šaltinių, kurie išskiria pasirinkto standarto pranašumą. Taip pat bus atliekamas eksperimentas realioms projektams, kad patikrinti gautus rezultatus.

Kiti egzistuojantys rizikos faktoriai:

- Egzistuoja rizika dėl metodo naujumo.

- Vartotojui sukurtoji sistema gali būti nepriimtina, reikalaujanti daug pastangų ją naudojantis. Bus stengiamasi sukurti sistemą, kad projektuotojas neužgaištų daug laiko vertindamas projektų elementus.
- Rizika, jog tiriamojo darbo atsakymas gali būti neigiamas, tai parodys atliktas eksperimentas.

2.5.5. Rezultato kokybės kriterijai

Išskiriami tokie rezultato kokybės kriterijai:

- Tinkamumas darbų apimčiai įvertinti. Sistemos realizacija tikslesniam funkcinių reikalavimų įvertinimui.
- Suprantamumas. Eksperimento metu greitas perpratimas projektų elementų skaičiavimo.
- Efektyvumas. Projektuotojai, lyginant su projektų visa trukme, neužgaišta daug laiko taikydami *FiSMA* metodiką.
- Funkcinių reikalavimų įkėlimas. Įkeliami funkciniai reikalavimai į sistemą, tuo palengvinama reikalavimų analizė, t.y sutrumpinamas jos laikas.

2.6. Analizės išvados

1. Analizės dalyje išskirti dažniausiai praktikoje taikomi IS projektų darbų apimties nustatymo sprendimai: valandinio įkainio kontraktai ir fazinis apimties įvertinimas fiksuotos kainos kontraktuose. Apibrėžtos tokios esamų sprendimų problemos: išanalizuotų ir suformuluotų reikalavimų nebuvimas, nesutarimas dėl projekto pradžioje apimties įvertinimo tikslumo, projektų vadovų kvalifikacijos stoka, kuri remiasi vadovavimusi savo patirtimi, statistinės informacijos nebuvimas. Dėl šių priežasčių darbų apimties įvertinimų patikimumas yra sunkiai prognozuojamas ir mažai patikimas.
2. Atlikta algoritminio kainos modeliavimo metodų analizė, apžvelgti *COCOMO [BOEHM-81]*, *Putmano SLIM* ir *Albrecht* funkcinių taškų (*FP*) metodai. Tačiau daugumos algoritminių kainų metodų pagrindinė metrika yra kodo eilučių skaičius, įvertint programinės įrangos apimtį šiais metodais yra sudėtinga. Be to, jeigu reikalavimai pastoviai keičiasi, tai eilučių kiekio skaičiavimas nėra

prasingas, kadangi programos nėra statinės. Šiuo aspektu funkcinės apimties matavimas yra naudingesnis.

3. Antra pagal dažnumą naudojama metrika yra funkcionalumas, todėl atlikta *FSM* (funkcinės apimties matavimo) modelių standartų apžvalga. Apžvelgti tokie 4 *ISO* standartai: *IFPUG Function Point Analysis*, *Mark II Function Point Analysis*, *COSMIC Full Function Points* ir *FiSMA*, apibrėžiant metodo plėtojimą, esmę, kiekvieno iš jų siūlomą įvertinimo procesą. Metodų apžvalga padėjo apsisprendimui, kokią metodiką pritaikyti būtų efektyviausia naujos sistemos realizacijai.
4. Analizės rezultate buvo pasirinktas universaliausias *FiSMA* standartas, kuris remiasi objektyviais kriterijais (vertinami labai aiškūs projekto elementai), plati klasių grupė, kuri įtakoja įvairių informacinių sistemų vertinimą, klasės išskirstytos į itin detalius tipus, kas įtakoja tikslesnį funkcinio reikalavimo įvertinimą. Standartas yra naujas, projektuotas ir koreguotas remiantis vartotojų poreikiais, iškilusiomis problemomis.
5. Siekiama realizuoti pasirinktą sistemą pagal *FiSMA* standartą, įvertinant ir apskaičiuojant projekto funkcinius reikalavimus, pakartotinio panaudojimo vykdymą. Šia realizacija siekiama, kad projektų vadovas būtų kuo mažiau apkrautas matematiniais skaičiavimais ir galėtų sutaupyti laiko, turėdamas galimybę įkelti į sistemą funkcinius reikalavimus, ir kaupdamas projektų statistinius duomenis tiksliau įvertintų spėjamą funkcinių reikalavimų realizavimo laiką.

3. FiSMA metodikos reikalavimų specifikacija ir analizė

Šiame skyriuje yra pateikiama standarto *FiSMA* reikalavimų specifikacija.

3.1. Reikalavimų specifikacija

Šiame poskyryje yra pateikiama *FiSMA* metodikos taikymo sritis, funkciniai, nefunkciniai reikalavimai, apimantys panaudojimo atvejų diagramas ir jų specifikacijas, sekų diagramas.

3.2. Sprendimo taikymo sritis

Kuriant sistemas yra sudėtinga apskaičiuoti kiek konkrečiai užtruks reikalavimų atlikimas. Kyla rizika, kad atskiros projekto dalys arba visas projektas truks ilgiau, nei iš pradžių planuota, atsiranda papildomų išlaidų rizika. Siekiant išspręsti šią problemą pasirinktas standartas *FiSMA*, kurio taikymas padės tiksliau apskaičiuoti funkcinų reikalavimų įvykdymo trukmę.

Šiame poskyryje išanalizuota *FiSMA* standarto reikalavimų specifikacija. Priešingai nei esamas IS darbų apimtį įvertinimas, šiame metode surinkus reikalavimus, reikalavimų specifikacijos bus sukeltos į sistemą. Skaičiuojant funkcinų reikalavimų apimtį, pasirenkamas tipas, kuriam priskirti atitinkami parametrai. Įvestos funkcinų reikalavimų tipų reikšmės apskaičiuojamos pagal matematinės formules, sumuojant atskiras funkcinų reikalavimų reikšmes į bendrą sumą. Kiekvienas įvertintas funkcinis reikalavimas priskiriamas vykdytojui, nurodant terminą, iki kada reikalavimas turi būti atliktas, nurodomas atitinkamas įkainis už atliktą veiklą. Priskyrus darbus, stebimas ir kontroliuojamas jų atlikimas pagal pateiktus projekto terminus, kaupiant statistinius projektų duomenis (funkcinų reikalavimų tipų spėjamus ir faktinius laikus). Taip didinama tikimybė nurodyti darbų vykdytojui spėjamą atlikimo laiką, artimiausią faktiniam atlikimo laikui.

3.3. Nefunkciniai reikalavimai

Nefunkciniai reikalavimai, keliami informacinei sistemai:

Bendrieji reikalavimai IS:

- *FiSMA* standarto atitikimas. Šio standarto laikymasis kuriant naują sistemą padės užtikrinti tikslesnį darbų apimties apskaičiavimą.
- Sistemos panaudojimo paprastumas. Sistema kuriama siekiant automatizuoti funkcinių reikalavimų apimties apskaičiavimą, kad vartotojai nebūtų apkrauti skaičiavimais.
- Reikalavimų įkėlimo šablonas turi būti vartotojui tikslus ir aiškus.
- Turi būti sukurtas vartotojo vadovas, aprašantis, kaip atlikti funkcinių reikalavimų įvertinimą.

Vartotojo sąsajai:

- Vartotojo sąsaja turi turėti minimalų veiksmų skaičių, gerą išdėstymą, kad būtų suprantama ir vartotojui patogi naudoti.
- Naudojimosi patogumas. Susijusi informacija turi būti išdėstoma šalia, vartotojo sąsaja neturi būti perkrauta informacija, kad būtų lengviau įsisavinamas skaičiavimų procesas.
- Vartotojo sąsajos aiškumas. Jei atliekami ilgesni skaičiavimai turėtų matytis sistemos būseną, išvesti pranešimai apie sistemoje vykstančius veiksmus.

Vartotojo duomenims:

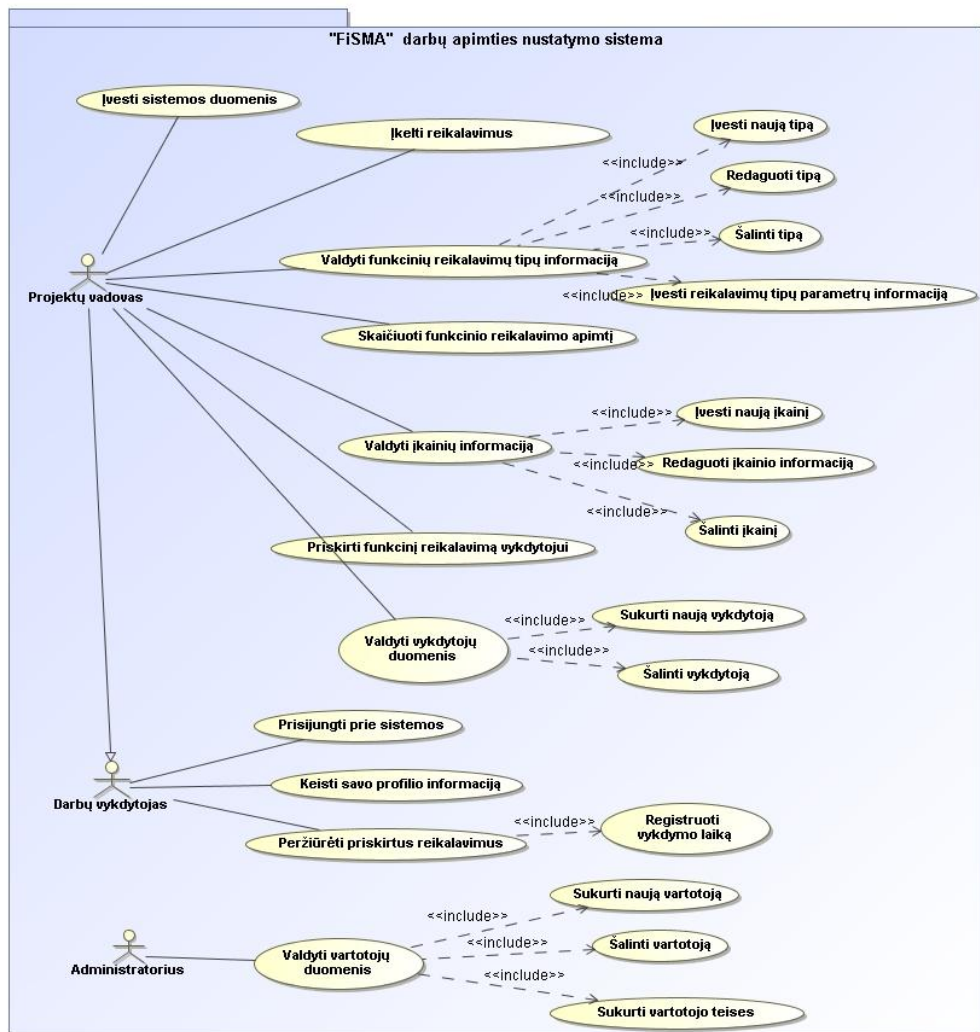
- Sistemai keliami tikslumo reikalavimai, todėl turi būti tikrinama duomenų sintaksė, įvedami reikalavimų parametrai turi būti realūs skaičiai.
- Saugumo priemonė vartotojams. Į sistemą įkelti reikalavimus, juos priskirti ir kontroliuoti gali tik sistemoje registruoti ir atitinkamas teises turintys vartotojai; peržiūrėti reikalavimus gali tik registruoti vartotojai, kuriems tie reikalavimai yra priskirti.

3.4. Funkciniai reikalavimai

Nustatant kuriamos sistemos funkcinius reikalavimus, reikia apibrėžti, kokios veiklos funkcijos turi būti kompiuterizuojamos.

3.5. Kompiuterizuojamų panaudojimo atvejų diagrama

Šioje sistemoje yra 3 tipų vartotojai: projektų vadovas, darbų vykdytojas ir administratorius. Šių vartotojų atliekamos funkcijos matomos bendroje kompiuterizuojamų panaudojimo atvejų diagramoje (7 pav.):



7 pav. Kompiuterizuojamų panaudojimo atvejų modelis

Projektų vadovas: didžiausias teises turintis registruotas vartotojas. Tik jis gali įkelti į sistemą naujus reikalavimus, įvesti duomenis, susijusius su projektuojama sistema, valdyti funkcinių reikalavimų tipų informaciją, įvedant reikalavimų tipų parametrų reikšmes. Tai pat jis gali peržiūrėti ir kontroliuoti reikalavimų sąrašą, priskirti vykdytojui funkcinius reikalavimus, įkainius ir atlikimo datas, valdyti šių vykdytojų informaciją.

Vykdytojas: tai projektų vadovo patvirtintas vartotojas. Prisijungęs jis gali kuri peržiūrėti jam priskirtus vykdyti reikalavimus. Taip pat gali redaguoti savo asmeninę informaciją.

Administratorius: tai vartotojas, kuris valdo vartotojų duomenis, sukurdamas, šalindamas vartotoją, taip pat šis vartotojas kuria vartotojų teises.

3.6. Reikalavimų panaudojimo atvejų specifikacija

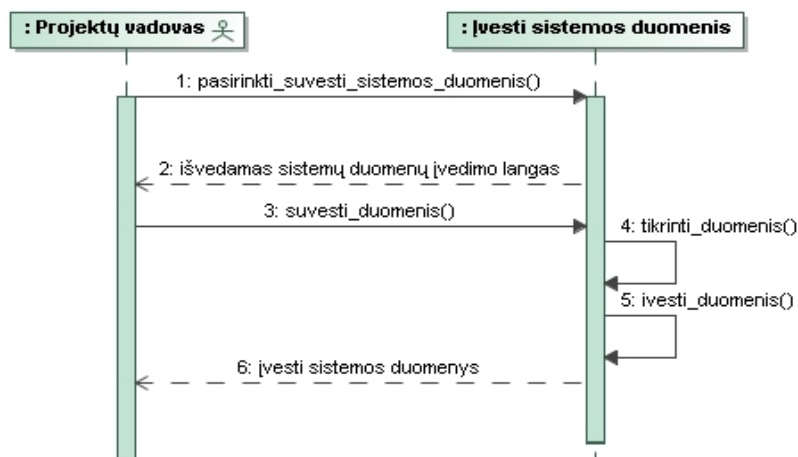
Naujai kuriamoje sistemoje turi būti kompiuterizuotos tokios funkcijos:

- Įvesti sistemos duomenis (8 pav.), identiškas panaudojimo atvejis: „Įvesti reikalavimų tipų parametrų informaciją“

- Įkelti reikalavimus (9 pav.)
- Valdyti funkcinių reikalavimų tipų informaciją (10 pav.)
- Įvesti naują tipą (11 pav.), identiški panaudojimo atvejai: „Įvesti naują įkainį“, „Sukurti naują vykdytoją“, „Sukurti naują vartotoją“, „Sukurti vartotojo teises“.
- Redaguoti tipą (12 pav.), identiški panaudojimo atvejai: „Redaguoti įkainio informaciją“, „Keisti savo profilio informaciją“.
- Šalinti tipą (13 pav.), identiški panaudojimo atvejai: „Šalinti įkainį“, „Šalinti vykdytoją“, „Šalinti vartotoją“.
- Skaičiuoti funkcinio reikalavimo apimtį (15 pav.)
- Valdyti įkainių informaciją (16 pav.)
- Valdyti vykdytojų duomenis (22 pav.)
- Priskirti funkcinį reikalavimą vykdytojui (23 pav.)
- Prisijungti prie sistemos (24 pav.)
- Peržiūrėti priskirtus reikalavimus (26 pav.)
- Valdyti vartotojų duomenis (27 pav.)
- Valdyti vykdytojų duomenis (31 pav.)

Pateikiamos projektų vadovo, darbų vykdytojo ir administratoriaus specifikacijos:

Panaudojimo atvejo „Įvesti sistemos duomenis“ diagrama pavaizduota 8 pav. Identiškas panaudojimo atvejis „Įvesti reikalavimų tipų parametrų informaciją“, todėl jis nepateikiamas atskirai.



8 pav. Panaudojimo atvejo „Įvesti sistemos duomenis“ diagrama

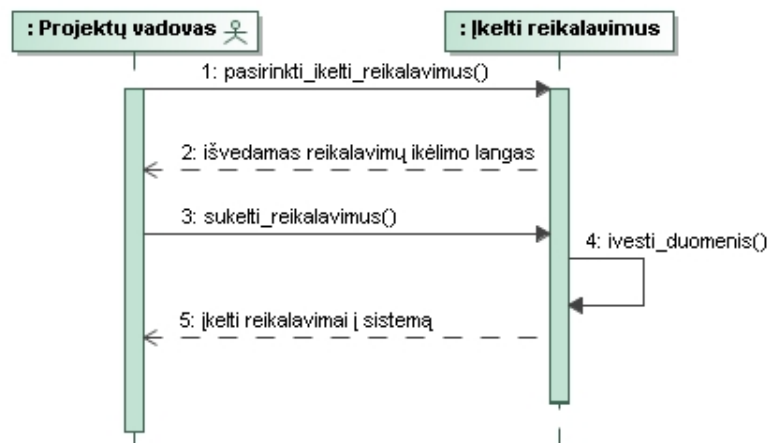
Panaudojimo atvejo „Įvesti sistemos duomenis“ specifikacija.

5 lentelė. Panaudojimo atvejo „Įvesti sistemos duomenis“ specifikacija

PA „Įvesti sistemos duomenis“

Prieš sąlyga		Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos
Aktorius		Projektų vadovas
Sužadinimo sąlyga		Vartotojas nori suvesti duomenis į sistemą
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda sistemos duomenis.		1.1. Sistema išveda sistemų duomenų įvedimo formą. 1.2. Tikrinami duomenys. 1.3. Įvedami į sistemą duomenys.
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Suvesti duomenys neteisingi.		A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.

Panaudojimo atvejo „Įkelti reikalavimus“ diagrama pavaizduota 9 pav.



9 pav. Panaudojimo atvejo „Įkelti reikalavimus“ diagrama

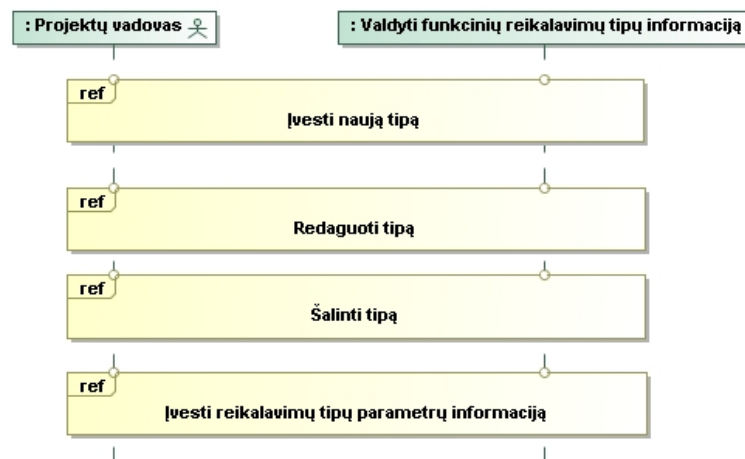
Panaudojimo atvejo „Įkelti reikalavimus“ specifikacija.

6 lentelė. Panaudojimo atvejo „Įkelti reikalavimus“ specifikacija

PA „Įkelti reikalavimus“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadinimo sąlyga	Vartotojas nori įkelti reikalavimus į sistemą	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas įkelia reikalavimus į sistemą.	1.1. Sistema išveda reikalavimų įkėlimo formą. 1.2. Funkciniai reikalavimai įkeliami.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Įkeliamų reikalavimų netinkamas formatas.	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.	

Panaudojimo atvejo „Valdyti funkcinį reikalavimų tipų informaciją“ diagrama pavaizduota 10 pav.



10 pav. Panaudojimo atvejo „Valdyti funkcinį reikalavimų tipų informaciją“ diagrama

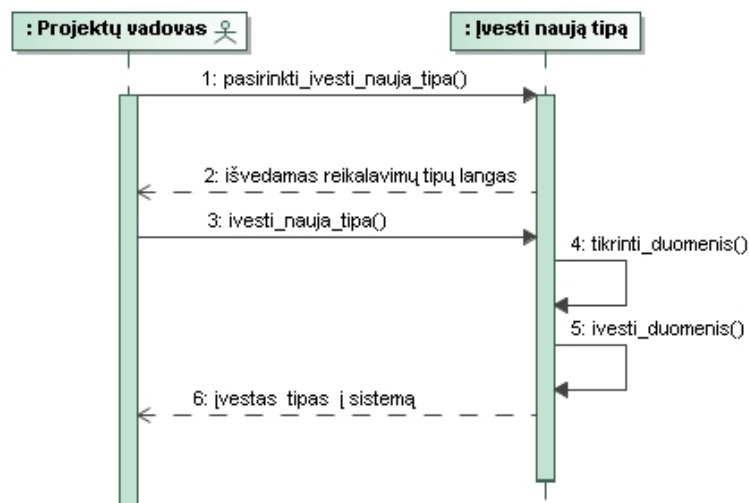
Panaudojimo atvejo „Valdyti funkcinį reikalavimų tipų informaciją“ specifikacija.

7 lentelė. Panaudojimo atvejo „Valdyti funkcinį reikalavimų tipų informaciją“ specifikacija

PA „Valdyti funkcinį reikalavimų tipų informaciją“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadavimo sąlyga	Vartotojas nori valdyti funkcinių reikalavimų tipų informaciją.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	„Įvesti naują tipą“ „Redaguoti tipą“ „Šalinti tipą“ „Įvesti funkcinio reikalavimo tipų parametrų reikšmes“
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas valdo funkcinių reikalavimų tipų informaciją.	1.1. Sistema išveda funkcinių reikalavimų tipų informacijos formą.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas funkcinių reikalavimų tipų meniu punktas	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.	

Panaudojimo atvejo „Įvesti naują tipą“ diagrama pavaizduota 11 pav. Identiški panaudojimo atvejai yra: „Įvesti naują įkainį“, „Sukurti naują vykdytoją“, „Sukurti naują vartotoją“, „Sukurti vartotojo teises“, todėl jie nepateikiami atskirai.



11 pav. Panaudojimo atvejo „Įvesti naują tipą“ diagrama

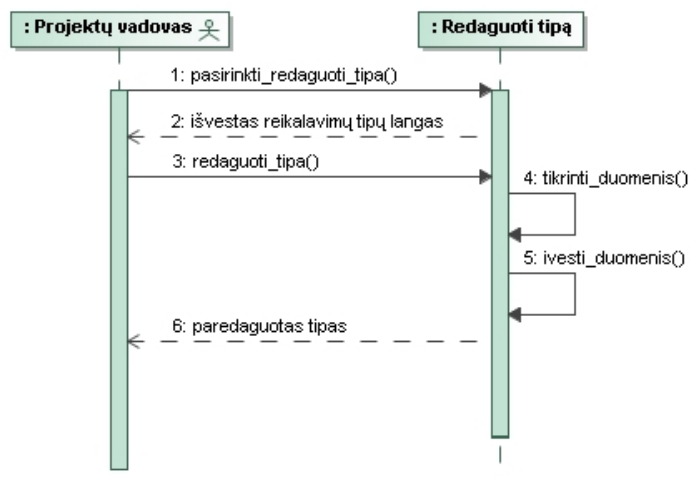
Panaudojimo atvejo „Įvesti naują tipą“ specifikacija.

8 lentelė. Panaudojimo atvejo „Įvesti naują tipą“ specifikacija

PA „Įvesti naują tipą“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadinimo sąlyga	Vartotojas nori įvesti naują tipą.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas įveda naują tipą.	1.1. Sistema išveda reikalavimų tipų langą. 1.2. Sistema tikrina duomenis. 1.3. Sistema įveda duomenis.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Neteisingai įvestas naujas tipas.	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.	

Panaudojimo atvejo „Redaguoti tipą“ diagrama pavaizduota 12 pav. Identiškas panaudojimo atvejis yra: „Redaguoti įkainio informaciją“, „Keisti savo profilio informaciją“ todėl jis atskirai nėra pateikiamas.



12 pav. Panaudojimo atvejo „Redaguoti tipą“ diagrama

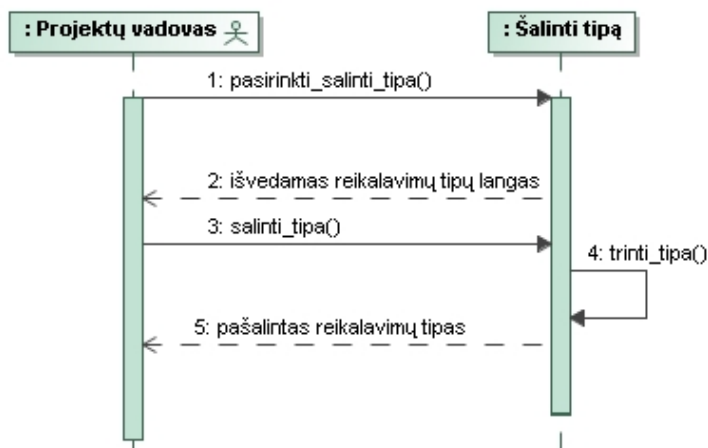
Panaudojimo atvejo „Redaguoti tipą“ specifikacija.

9 lentelė. Panaudojimo atvejo „Redaguoti tipą“ specifikacija

PA „Redaguoti tipą“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadinimo sąlyga	Vartotojas nori pagedaguti reikalavimų tipą.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas redaguoja reikalavimų tipą.	1.1. Sistema išveda reikalavimų tipų langą. 1.2. Sistema tikrina duomenis. 1.3. Sistema įveda duomenis.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Neteisingai įvestas reikalavimų tipas.	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.	

Panaudojimo atvejo „Šalinti tipą“ diagrama pavaizduota 13 pav. Identiški panaudojimo atvejai yra: „Šalinti įkainį“, „Šalinti vykdytoją“, „Šalinti vartotoją“, todėl jų diagramos neperbraižomos atskirai.



13 pav. Panaudojimo atvejo „Šalinti tipą“ diagrama

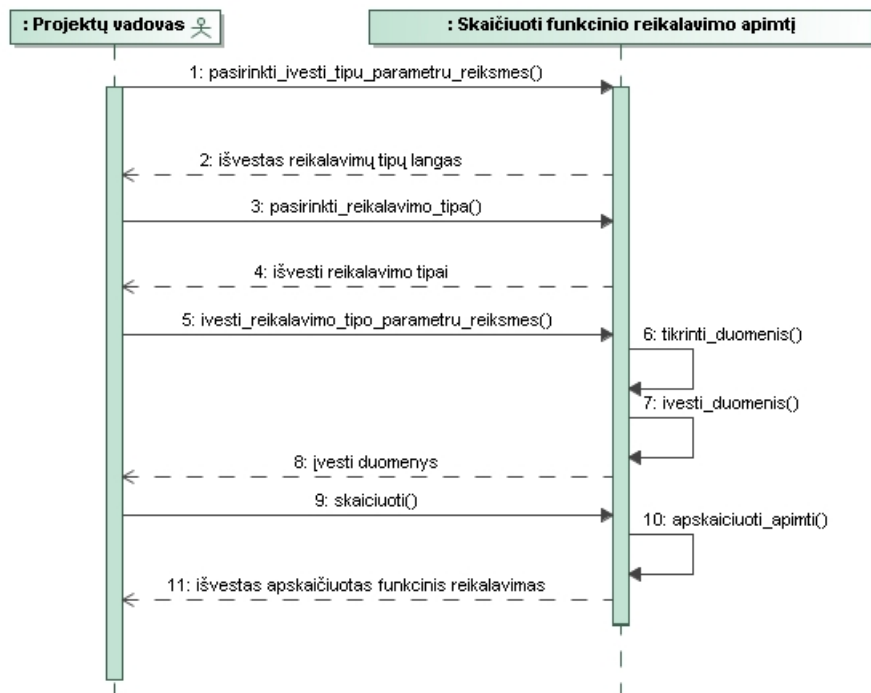
Panaudojimo atvejo „Šalinti tipą“ specifikacija.

10 lentelė. Panaudojimo atvejo „Šalinti tipą“ specifikacija

PA „Šalinti tipą“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadinimo sąlyga	Vartotojas nori pašalinti reikalavimų tipą.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas šalina reikalavimų tipą.	1.1. Sistema išveda reikalavimų tipų langą. 1.2. Sistema šalina reikalavimų tipą.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas reikalavimų tipas.	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.	

Panaudojimo atvejo „Skaičiuoti funkcinio reikalavimo apimtį“ diagrama pavaizduota 14 pav.



14 pav. Panaudojimo atvejo „Skaičiuoti funkcinio reikalavimo apimtį“ diagrama

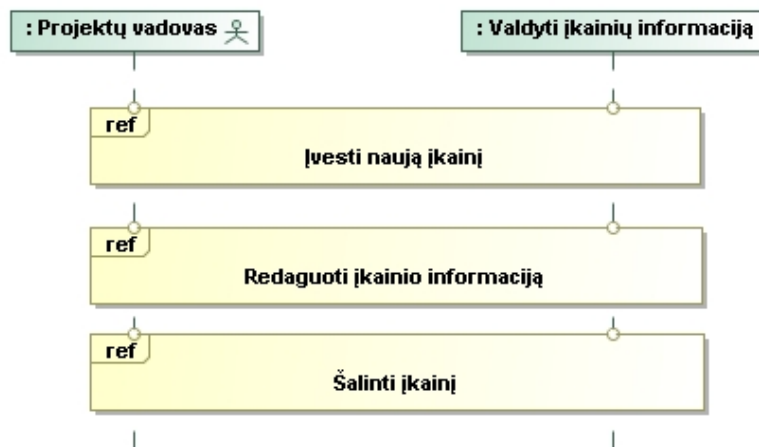
Panaudojimo atvejo „Įvesti funkcinio reikalavimo tipų parametrų reikšmes“ specifikacija.

11 lentelė. Panaudojimo atvejo „Skaičiuoti funkcinio reikalavimo apimtį“ specifikacija

PA „Skaičiuoti funkcinio reikalavimo apimtį“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius		Projektų vadovas
Sužadinimo sąlyga		Vartotojas apskaičiuoti funkcinio reikalavimo apimtį.
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda reikalavimų tipų parametrų reikšmes.		1.1. Sistema išveda reikalavimų tipų parametrų langą. 1.2. Sistema tikrina duomenis. 1.3. Sistema įveda duomenis.
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Neteisingai įvesta reikalavimų tipų parametrų reikšmė.		A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.

Panaudojimo atvejo „Valdyti įkainių informaciją“ diagrama pavaizduota 15 pav.



15 pav. Panaudojimo atvejo „Valdyti įkainių informaciją“ diagrama

Panaudojimo atvejo „Valdyti įkainių informaciją“ specifikacija.

12 lentelė. Panaudojimo atvejo „Valdyti įkainių informaciją“ specifikacija

PA „Valdyti įkainių informaciją“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius		Projektų vadovas
Sužadinimo sąlyga		Vartotojas nori valdyti funkcinių įkainių informaciją.
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	„Įvesti naują įkainį“ „Redaguoti įkainio informaciją“ „Šalinti įkainį“
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas valdo funkcinių reikalavimų įkainių informaciją.		1.1. Sistema išveda funkcinių reikalavimų įkainių informacijos formą.
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas funkcinių reikalavimų įkainių meniu punktas		A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.

Panaudojimo atvejo „Valdyti vykdytojų duomenis“ diagrama pavaizduota 16 pav.



16 pav. Panaudojimo atvejo „Valdyti vykdytojų duomenis“ diagrama

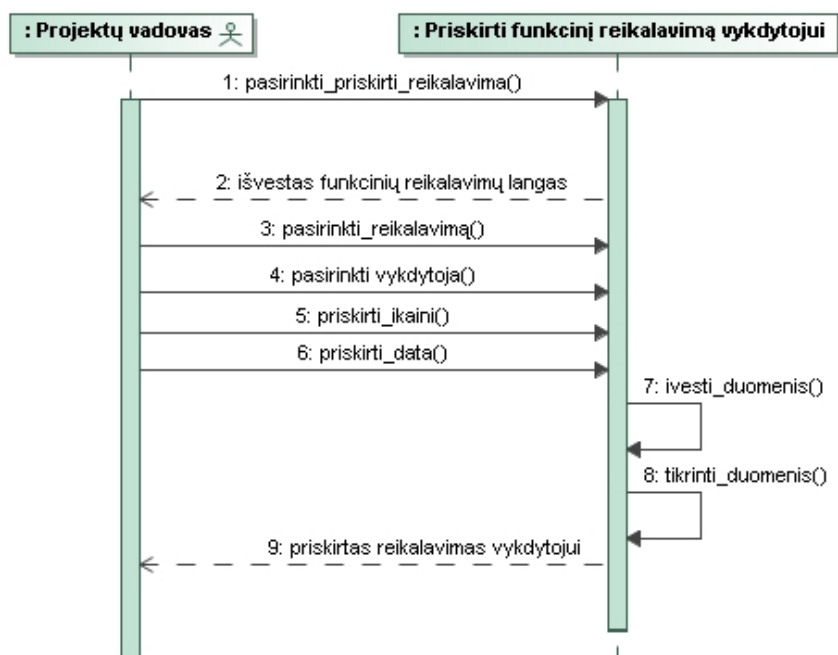
Panaudojimo atvejo „Valdyti vartotojų duomenis“ specifikacija.

13 lentelė. Panaudojimo atvejo „Valdyti vartotojų duomenis“ specifikacija

PA „Valdyti vartotojų duomenis“	
Prieš sąlyga	Vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius	Projektų vadovas	
Sužadinimo sąlyga	Vartotojas nori valdyti vykdytojų duomenis.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	„Sukurti naują vykdytoją“ „Šalinti vykdytoją“
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas valdo vykdytojų duomenis.	1.1. Sistema išveda vykdytojų duomenų formą.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas vykdytojų meniu punktas	A1.1. PA nevykdomas.	

Panaudojimo atvejo „Priskirti funkcinių reikalavimą vykdytojui“ diagrama pavaizduota 17 pav.



17 pav. Panaudojimo atvejo „Priskirti funkcinių reikalavimą vykdytojui“ diagrama

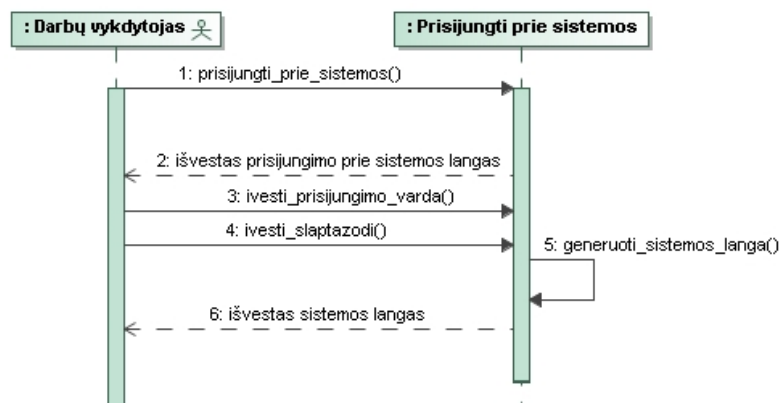
Panaudojimo atvejo „Priskirti funkcinių reikalavimą vykdytojui“ specifikacija.

14 lentelė. Panaudojimo atvejo „Priskirti funkcinių reikalavimą vykdytojui“ specifikacija

PA „Priskirti funkcinių reikalavimą vykdytojui“	
Prieš sąlyga	Projektų vadovas yra įvestas į duomenų bazę; vartotojas turi projektų vadovo teises; vartotojas prisijungęs prie sistemos

Aktorius		Projektų vadovas
Sužadinimo sąlyga		Vartotojas nori priskirti funkcinį reikalavimą vykdytojui
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka reikalavimą. 2. Vartotojas pasirenka vykdytoją. 3. Vartotojas priskiria įkainį. 4. Vartotojas priskiria datą.		1.1. Sistema išveda funkcinių reikalavimų formą. 1.2. Sistema tikrina duomenis. 1.3. Sistema įveda duomenis.
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas funkcinis reikalavimas ir/ar vykdytojas, ir/ar įkainis, ir/ar data.		A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.

Panaudojimo atvejo „Prisijungti prie sistemos“ diagrama pavaizduota 18 pav.



18 pav. Panaudojimo atvejo „Prisijungti prie sistemos“ diagrama

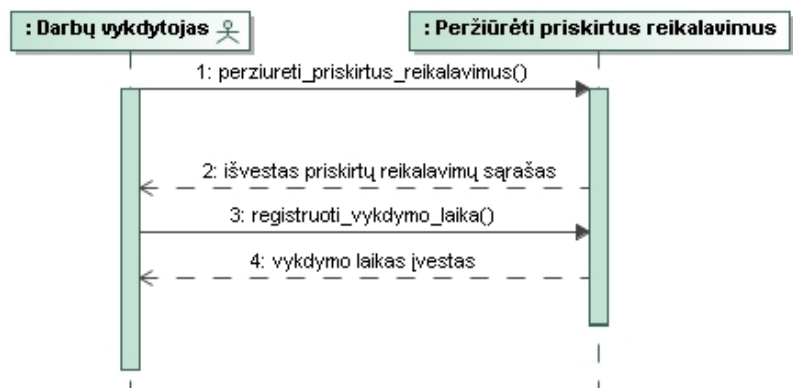
Panaudojimo atvejo „Prisijungti prie sistemos“ specifikacija.

15 lentelė. Panaudojimo atvejo „Prisijungti prie sistemos“ specifikacija

PA „Prisijungti prie sistemos“		
Prieš sąlyga		Darbų vykdytojas yra įvestas į duomenų bazę; vartotojas turi vykdytojo teises; vartotojas prisijungęs prie sistemos
Aktorius		Darbų vykdytojas
Sužadinimo sąlyga		Vartotojas nori prisijungti prie sistemos
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda prisijungimo vardą.		1.1. Sistema išveda prisijungimo prie sistemos langą.

2. Vartotojas įveda slaptažodį.	1.2. Sistema tikrina duomenis. 1.3. Sistema generuoja sistemos formą.
2. Vartotojas baigia PA	
Po sąlyga:	
Alternatyvūs scenarijai	
A1. Blogai įvestas prisijungimo vardas ir/ar slaptažodis.	A1.1. Vartotojui pateikiamas pranešimas, PA vykdymas nutraukiamas.

Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ diagrama pavaizduota 19 pav.



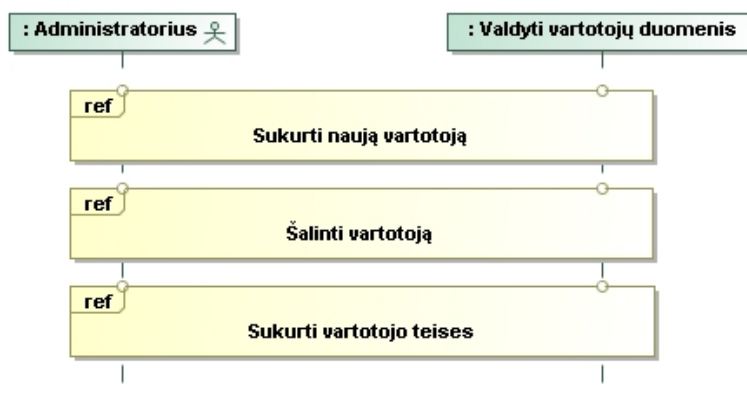
19 pav. Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ diagrama

Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ specifikacija.

16 lentelė. Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ specifikacija

PA „Peržiūrėti priskirtus reikalavimus“		
Prieš sąlyga		Darbų vykdytojas yra įvestas į duomenų bazę; vartotojas turi vykdytojo teises; vartotojas prisijungęs prie sistemos
Aktorius		Darbų vykdytojas
Sužadinimo sąlyga		Vartotojas nori peržiūrėti priskirtus reikalavimus, įvesti vykdymo laiką.
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas peržiūri priskirtus funkcinis reikalavimus, įveda vykdymo laiką.		1.1. Sistema išveda priskirtų reikalavimų formą
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas reikalavimų meniu punktas.		A1.1. PA nevykdomas.

Panaudojimo atvejo „Valdyti vartotojų duomenis“ diagrama pavaizduota 20 pav.



20 pav. Panaudojimo atvejo „Valdyti vartotojų duomenis“ diagrama

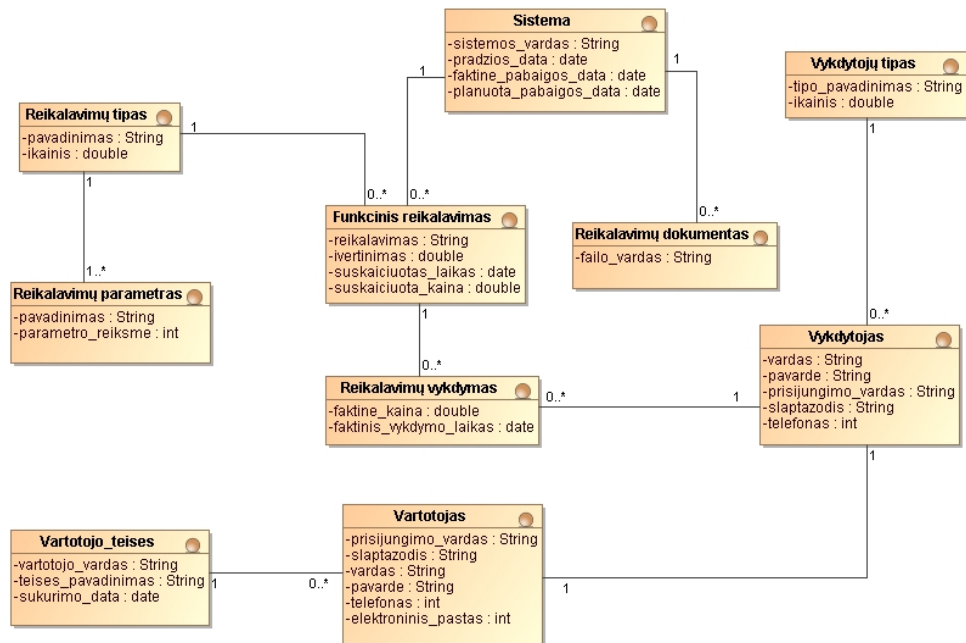
Panaudojimo atvejo „Valdyti vartotojų duomenis“ specifikacija.

17 lentelė. Panaudojimo atvejo „Valdyti vartotojų duomenis“ specifikacija

PA „Valdyti vartotojų duomenis“		
Prieš sąlyga	Vartotojas turi administratoriaus teises; vartotojas prisijungęs prie sistemos	
Aktorius	Administratorius	
Sužadavimo sąlyga	Vartotojas nori valdyti vartotojų duomenis.	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	„Sukurti naują vartotoją“ „Šalinti vartotoją“ „Sukurti vartotojo teises“
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas valdo vartotojų duomenis.	1.1. Sistema išveda vartotojų duomenų formą.	
2. Vartotojas baigia PA		
Po sąlyga:		
Alternatyvūs scenarijai		
A1. Nepasirinktas vartotojų meniu punktas	A1.1. PA nevykdomas.	

3.7. Dalykinės srities modelis

Išanalizavus dalykinės srities esybes, buvo sudaryta dalykinės srities esybių klasių diagrama (21 pav.). Išskirtos tokios esybės: „Sistema“, „Reikalavimų tipas“, „Funkcinis reikalavimas“, „Reikalavimų dokumentas“, „Reikalavimų parametras“, „Reikalavimų vykdymas“, „Vykdytojų tipas“, „Vykdytojas“.



21 pav. Dalykinės srities esybių modelis

3.8. Reikalavimų analizės apibendrinimas

1. Reikalavimų analizė apima pagal *FiSMA* standartą kuriamos sistemos panaudojimo atvejų diagramas ir jų specifikacijas, sekų, veiklos diagramas. Tai padėjo išskirti veiklos procesus, kuriuos turi kompiuterizuoti naujai kuriama sistema ir su jais susijusius veiklos dalyvius (projektų vadovą ir darbų vykdytojus).
2. Sudaryti kuriamai sistemai keliami nefunkciniai reikalavimai, apibrėžiantys, kokiomis savybėmis pasižymės naujai kuriama sistema. Realizuotas sistemos dalykinės srities modelis padėjo suprasti, kokie bus gaunami ir išvedami duomenys.
3. Taigi, standartą *FiSMA* realizuojančios sistemos reikalavimų specifikacija tiksliau nustatė sprendimo taikymo sritį, realizuotos diagramos padėjo detaliau apžvelgti standarto pritaikymą sistemai.

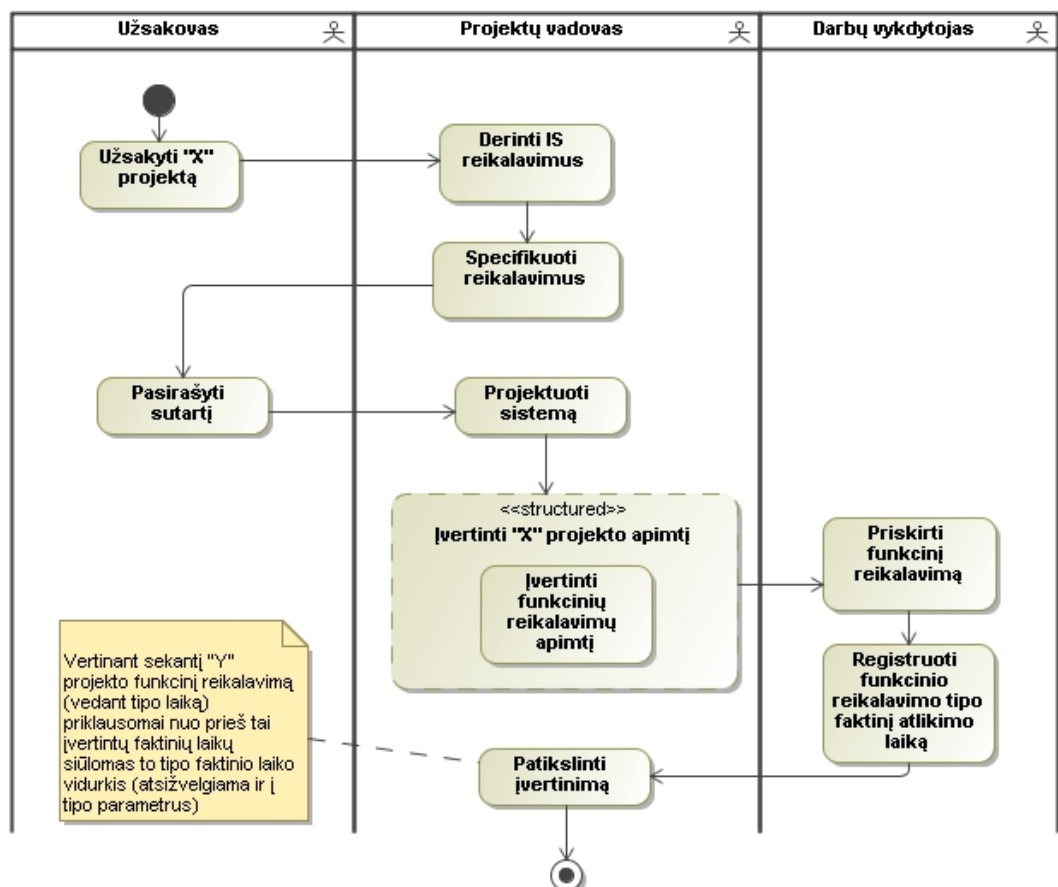
4. „FiSMA 1.1“ metodikos aprašas

Funkcinė apimtis yra būtinas matas visiems programinės įrangos palyginimams. Įrodyta, kad jis yra naudingas projektų planavime, sekime, kontrolėje ir sutarties sudaryme. Funkcinės apimties matavimas veikia geriausiai, kai yra užbaigtas funkcinų vartotojų reikalavimų ir paslaugų sąrašas, jis padaro apimties ir pakeitimų valdymą efektyvesnį, patikimesnį ir palyginti lengviau suprantamą galiniam vartotojui.

„FiSMA 1.1“ yra bendras, parametrizuotas funkcinės apimties matavimo metodas visiems programinės įrangos tipams. 22 - 23 paveiksluose pateikti sistemos veikimo procesai realizavus FiSMA.



22 pav. Apibendrintas FiSMA realizavimo procesas



23 pav. Detalesnis sistemos veikimo procesas realizavus „FiSMA“

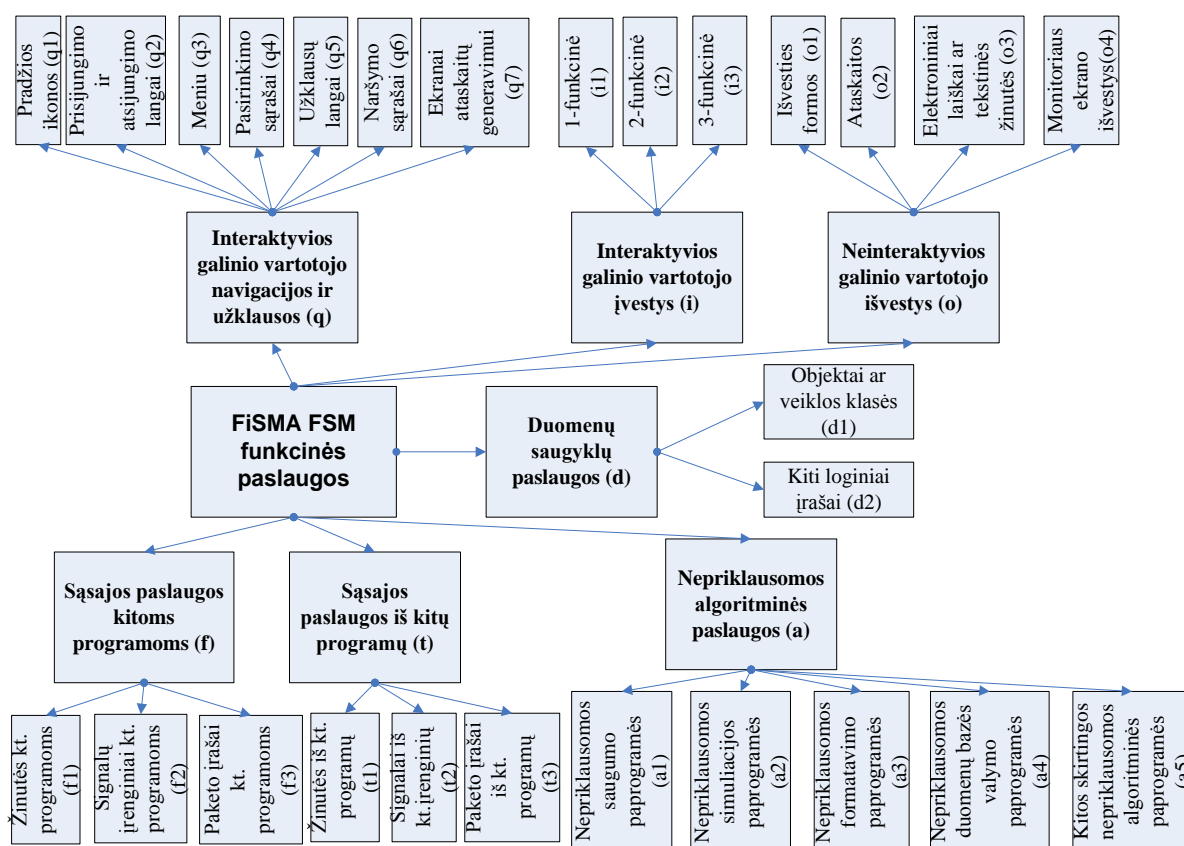
4.1. Formalizuotas „FiSMA 1.1“ funkcinės apimties įvertinimo metodikos aprašas

Šiame skyrelyje yra pateikiamas *FiSMA* metodikos aprašas.

4.1.1 „FiSMA 1.1“ klasės ir tipai

FiSMA funkciniai reikalavimai yra suskirstyti į 7 pagrindines funkcinės klases. Naujai kuriamos sistemos funkciniai reikalavimai yra priskiriami šioms klasėms. Sistemoje turi būti realizuojamas reikalavimų ir jam tinkamos klasės pasirinkimas, kad išvesti parametrus ir apskaičiuoti būtent tokios klasės reikalavimą.

Kiekviena iš klasių skirstoma į kelis tipus. Iš viso kartu yra 28 tipai. 24 paveikslas parodomas klasių ir jų sudedamųjų tipų santykis.



24 pav. *FiSMA 1.1* klasės ir tipai

Pagal funkciniam reikalavimui priskirtą klasę yra išvedami šiai klasei priklausantys tipų parametrai, kuriuos įvedęs projektų vadovas gaus reikalavimo įgyvendinimo trukmę.

Realizuojant informacinę sistemą funkcinių reikalavimų sąrašas bus sudarytas klausimų pavidalu, siekiant sužinoti funkcinis komponentus ir įvesti tinkamas reikšmes.

- **Interaktyvaus vartotojo navigatoriaus ir užklausų baziniai funkciniai komponentai** (visos interaktyvios funkcijos be redagavimo galimybių):
 - Kiek **pradžios ikonų** turite savo programoje?_____
 - Kiek skirtingų **prisijungimo ir atsijungimo langų** turite?_____
 - Kiek skirtingų **menu** turite savo programoje?_____
 - Kiek **pasirinkimo sąrašų (išplečiamų sąrašų)** turite savo programoje?_____
 - Kiek **užklausų langų** turite (rodančius duomenų bazės turinį vartotojams ekrane, neredaguojant nieko)?_____
 - Kiek **naršymo sąrašų** langų (rodančių keletą įvykių su tais pačiais duomenų tipais) turite?_____
 - Kiek skirtingų **ekranų pradėti ataskaitų generavimui** turite?_____
- **Interaktyvaus vartotojo informacijos įvedimo baziniai funkciniai komponentai** (kiekvienas tipas turi skirtingas redagavimo galimybes):
 - Kiek skirtingų **1-funkcinių (įvesti ar redaguoti ar trinti) vartotojo įvedimo langų** turite?_____
 - Kiek skirtingų **2-funkcinių (įvesti ir/ar redaguoti ir/ar trinti) vartotojo įvedimo langų** turite?_____
 - Kiek skirtingų **3-funkcinių (įvesti, redaguoti arba trinti) vartotojo įvedimo langų** turite?_____
- **Neinteraktyvaus vartotojo informacijos išvedimo baziniai funkciniai komponentai:**
 - Kiek skirtingų **išvedimo formų** numatyta programoje?_____
 - Kiek skirtingų **ataskaitų** numatyta programoje?_____
 - Kiek skirtingų **elektroninių laiškų ar tekstinių žinučių** Jūsų programoje yra rašoma ar siunčiama?_____
 - Kiek skirtingų **monitoriaus ekrano išvedimų** Jūs turite?_____
- **Sąsajų su kitomis programomis baziniai funkciniai komponentai:**
 - Kiek skirtingų **žinučių kitoms programoms** išsiunčiate?_____
 - Kiek skirtingų **signalų įrenginiams** išsiunčiate?_____
 - Kiek skirtingų **paketo įrašų kitai programai** išsiunčiate?_____
- **Sąsajų iš kitų programų baziniai funkciniai komponentai:**
 - Kiek skirtingų **žinučių iš kitų programų** gaunate?_____

- Kiek skirtingų **signalų iš įrenginių** gaunate?_____
- Kiek skirtingų **paketo įrašų iš kitų programų** gaunate?_____
- **Duomenų saugojimo paslaugos:**
 - Kiek **skirtingų objektų ar veiklos klasių** turite?_____
 - Kiek **kitų loginių įrašų tipų** turite?_____
- **Algoritmų ir manipuliavimo paslaugos:**
 - Kiek skirtingų **nepriklausomų saugumo paprogramių** Jūsų programa įtraukia?_____
 - Kiek skirtingų **nepriklausomų skaičiavimo paprogramių** Jūsų programa įtraukia?_____
 - Kiek skirtingų **nepriklausomų simuliacijos paprogramių** Jūsų programa įtraukia?_____
 - Kiek skirtingų **nepriklausomų formatavimo paprogramių** Jūsų programa įtraukia?_____
 - Kiek skirtingų **nepriklausomų duomenų bazės valymo paprogramių** Jūsų programa įtraukia?_____
 - Kiek **kitų** skirtingų **nepriklausomų algoritminių paprogramių** Jūsų programa įtraukia?_____

4.1.2. Skaičiavimo taisyklės kiekvienam tipui

1. *Interaktyvaus vartotojo navigatoriaus ir užklausų baziniai funkciniai komponentai (q):*

$$S_q = a_q + n / d_q + r_r / c_q$$

kur

S_q = užklausos (dialogo, meniu ir t. t. dydis)

n = duomenų elementų skaičius, laukai

r_r = esybių skaitymų nuorodų skaičius

d_q = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_q = klasei būdingas skaitymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

a_q = nustatymo konstanta, 0,2 Ffp

Ffp = funkcinio vieneto *FiSMA* funkciniai taškai, $d_q = 7,00$ $c_q = 2,00$

2. *Interaktyvaus vartotojo informacijos įvedimo baziniai funkciniai komponentai (i):*

$$S_i = m * (a_i + n/ d_i + r_w/ c_i + r_r/ b_i)$$

kur

S_i = įvedamos informacijos apimtis

m = funkcionalumo daugiklis, kurio reikšmė 1, 2 ar 3, priklausomai nuo to, kiek sukūrimo, įterpimo ir naikinimo funkcijų apima baziniai funkciniai komponentai.

n = duomenų elementų, laukų skaičius

r_w = esybių rašymo nuorodų skaičius

r_r = esybių tik skaitymo nuorodų skaičius

d_i = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_i = klasei būdingas rašymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

b_i = klasei būdingas skaitymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

a_i = Nustatymo konstanta, 0,2 Ffp

Ffp = funkcinio vieneto *FiSMA* funkciniai taškai, $d_i = 5,00$ $c_i = 1,50$, $b_i = 2,00$

3. *Neinteraktyvaus vartotojo informacijos išvedimo baziniai funkciniai komponentai*

(o):

$$S_o = a_o + n/ d_o + r_r/ c_o$$

kur

S_o = išvedamos informacijos apimtis

n = duomenų elementų, laukų skaičius

r_r = esybių skaitymo nuorodų skaičius

d_o = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_o = klasei būdingas skaitymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

a_o = nustatymo koeficientas 1,0 Ffp

Ffp = funkcinio vieneto *FiSMA* funkciniai taškai, $d_o = 5,00$ $c_o = 2,00$, „FiSMA“ funkcinės apimties matavimo vienetas, vertė d_o yra 5,00 ir vertė c_o yra 2,00

4. *Sąsajų su kitomis programomis baziniai funkciniai komponentai (t):*

$$S_t = a_t + n/d_t + r_r/ c_t$$

kur

S_t = sąsajų su kitomis programomis apimtis

n = duomenų elementų (atributų) skaičius

r_r = esybių skaitymo nuorodų skaičius

d_t = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_t = klasei būdingas skaitymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

a_t = Nustatymo koeficientas 0,5 Ffp

Ffp = funkcinio vieneto FiSMA funkciniai taškai, $dt = 7,00$ $ct = 2,00$

5. *Sąsajų iš kitų programų baziniai funkciniai komponentai (f):*

$$S_f = a_f + n/d_f + r_w/c_f + r_r/b_f$$

kur

S_f = sąsajų iš kitų programų apimtis

n = duomenų elementų, laukų skaičius

r_w = esybių rašymo nuorodų skaičius

r_r = esybių skaitymo nuorodų skaičius

d_f = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_f = klasei būdingas rašymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

b_f = klasei būdingas rašymo nuorodų skaičius, duodantis 1 funkcinį vienetą Ffp

a_f = nustatymo koeficientas 0,2 Ffp

Ffp = funkcinio vieneto FiSMA funkciniai taškai, $dt = 7,00$ $cf = 2,00$, $Bf = 2,00$

6. *Duomenų saugojimo paslaugos (d):*

$$S_d = a_d + n/d_d$$

kur

S_d = esybės ar įrašo dydis

n = duomenų elementų (atributų) skaičius

d_d = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

a_d = nustatymo koeficientas 1,5 Ffp

Ffp = funkcinio vieneto FiSMA funkciniai taškai, $dd = 4,00$

7. *Algoritmų ir manipuliavimo paslaugos (a):*

$$S_a = a_a + n/d_a + r_c/c_a$$

kur

S_a = algoritmo apimtis

n = duomenų elementų (kintamųjų, operandų) skaičius

r_c = operacijų skaičius

d_a = klasei būdingas duomenų elementų skaičius, duodantis 1 funkcinį vienetą Ffp

c_a = klasei būdingas skaičiavimo taisyklių skaičius, duodantis 1 funkcinį vienetą Ffp

a_a = nustatymo koeficientas 0,1 Ffp

Ffp = funkcinio vieneto *FiSMA* funkciniai taškai, $d_a = 5,00$, $c_a = 3,00$

4.1.3. Funkcinės apimties apskaičiavimas programinės įrangos daliai

Kiekviena programinės įrangos dalis turi būti išmatuota atskirai, ir visi funkciniai dydžiai sudėti kartu. Visas funkcinis programinės įrangos sistemos dydis yra suma funkcinų jos komponentų dydžių, t.y. programinės įrangos dalys įtrauktos į sistemą.

Programinės įrangos dalies funkcinė apimtis (S) yra suma dydžių (S_x) klasių. Bazinių funkcinų komponentų apimtis priklauso nuo jo tipo ir klasės konkrečių elementų kiekio.

Pateikta formule yra suskaičiuojamas funkcinis programinės įrangos dydis, kuris yra funkcinų komponentų dydžių suma:

$$S = S_q + S_i + S_o + S_f + S_t + S_d + S_a$$

4.1.4. Funkcinės apimties įvertinimo taikymo procesas

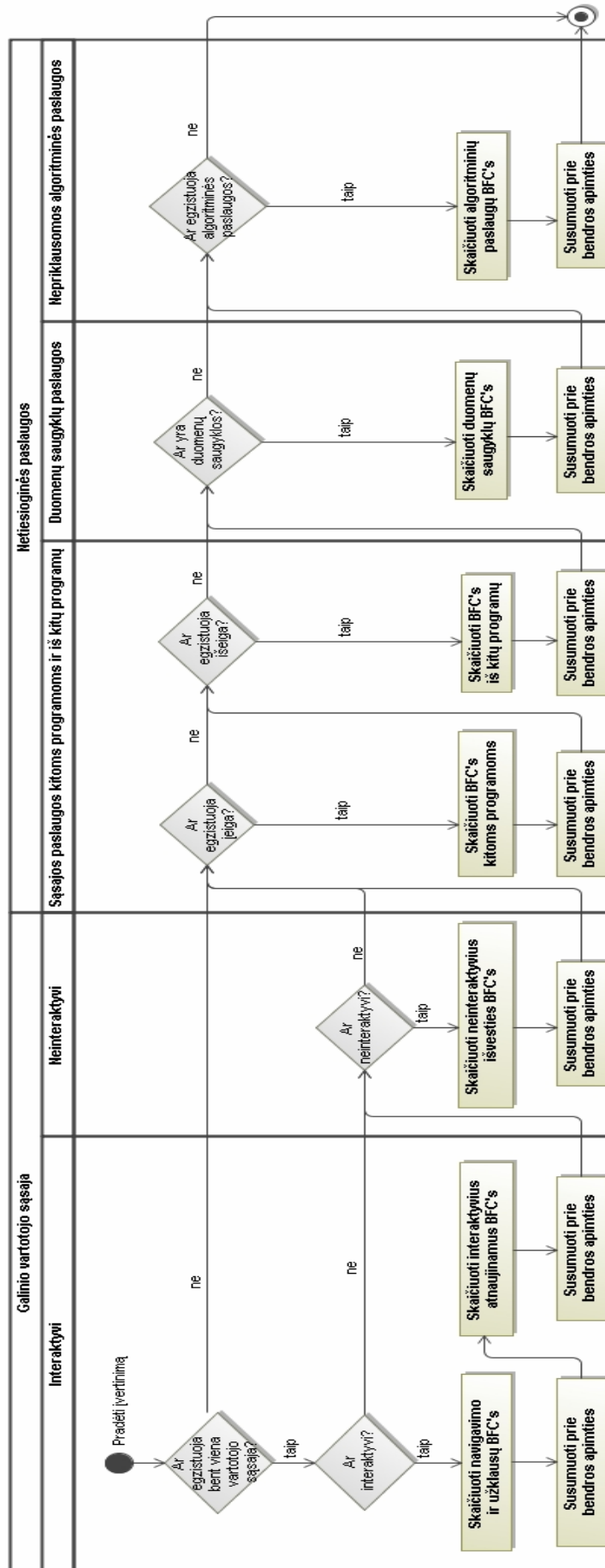
„*FiSMA 1.1*“ įvertinimo procesas susideda iš sekančių žingsnių:

1. Turi būti surinkta dokumentacija, aprašanti informacinės sistemos funkcinis reikalavimus. Tai gali apimti preliminarūs parengtus vartotojo reikalavimus, dalykinės srities esybių modelį, panaudojimo atvejus, ataskaitas, duomenų bazės modelius, duomenų srautų diagramas ir taip toliau – tas, kas apibūdina tai, ką programinė įranga atliks išreiškiant užduotimis ar paslaugomis, nepriklausomai nuo kokybės ar techninių reikalavimų.
2. Nustatoma *FSM* apimtis. „*FiSMA 1.1*“ apimtis yra nustatomas apimant funkcinis reikalavimus, kurie bus suprojektuoti ar patobulinti informacinėje sistemoje.
3. Reikalinga nustatyti ir apibrėžti, kurie yra funkciniai vartotojų reikalavimai (šie reikalavimai apibūdina tai, ką programinė įranga turi atlikti išreiškiant užduotimis ir paslaugomis), jie bus įvertinti „*FiSMA 1.1*“, nustatant apimtį.
4. Identifikuoti funkcinų reikalavimų pagrindiniai funkciniai komponentai į 2 dalis:
 - įvertinant galutinio vartotojo sąsajos paslaugas;
 - įvertinant netiesiogines paslaugas.

Jei viena iš šitų dviejų dalių neegzistuoja programinės įrangos daliai, tai procesas matavimas susideda tikrai iš paslaugų, kurios dalyvauja.

5. Klasifikuoti pagrindinius funkcinius komponentus į tinkamą tipą. Identifikuojant identišką funkcionalumą, jis turi būti įskaičiuojamas tiktai vieną kartą. Tipai yra identiški, jei jie turi tas pačias savybes (t.y., identiški duomenų elementai).
6. Priskiriama tinkama skaitmeninė vertė kiekvienam tipui.
7. Skaičiuojama funkcinė apimtis.

Šie žingsniai yra pateikiami 25 paveiksle:



25 pav. „FiSMA“ taikymo proceso schema pagal 7 tipus (klases)

Visų funkcinių reikalavimų iteracijos yra sumuojamos ir gaunamas bendras projekto įvertinimas.

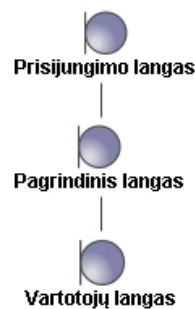
Atliktas apskaičiavimas pagal *FiSMA* bus lyginamas su prieš tai numatytu (spėtu). Bus daromos išvados, kaupiami istoriniai duomenys. Tuomet projektų vadovas priskirs reikalavimą darbų vykdytojui, nuroydamas pagal atliktus apskaičiavimus iki kada tai turi būti atlikta, priskirs įkainį.

4.1.5. Detalizuotas taikymo procesas

4.1.5.1. Vartotojo paslaugos

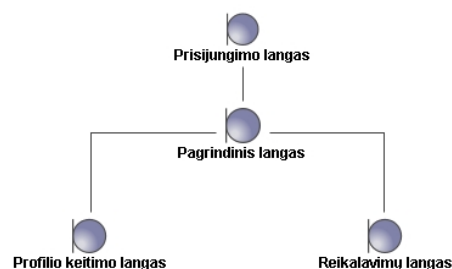
Sudarius navigavimo planą (26 - 28 pav.) galima matyti, kokios formos bus sistemoje.

Prisijungus, administratoriui atveriamas prisijungimo langas per kurį prisijungiama prie pagrindinio, iš kurio galima pasiekti vartotojų langą ir valdyti sistemą, sukuriant vartotoją, priskiriant jam teises, jį šalinant.



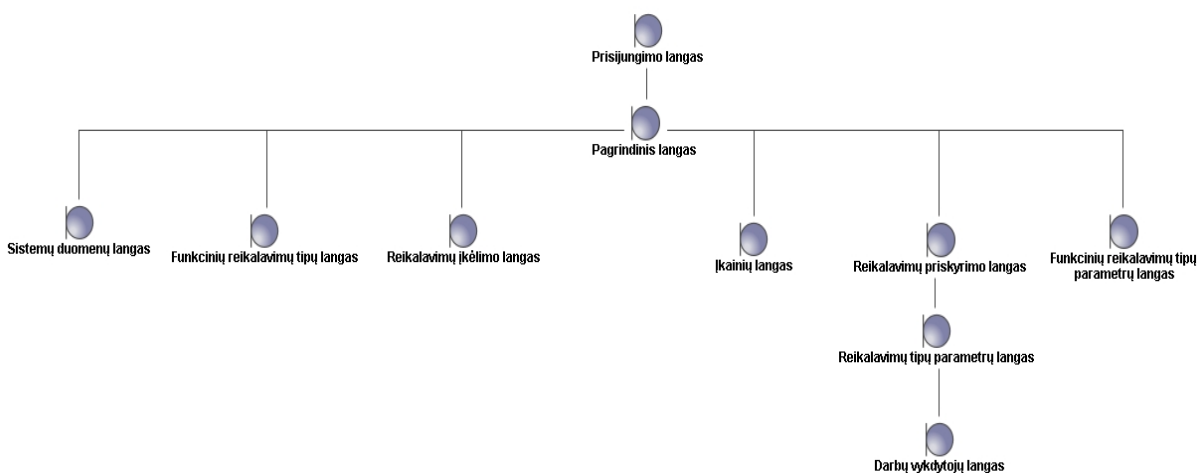
26 pav. Navigavimo planas administratoriui

Darbų vykdytojas gali atlikti veiksmus su reikalavimais, profilio keitimu. Reikalavimų lange darbų vykdytojas gali peržiūrėti priskirtus atlikti funkcinius reikalavimus (kada jie turi būti atlikti, koks jų įkainis). Taip pat profilio lange keisti savo profilio informaciją. Darbų vykdytojo navigavimo planas yra pateikiamas 27 pav.



27 pav. Navigavimo planas darbų vykdytojui

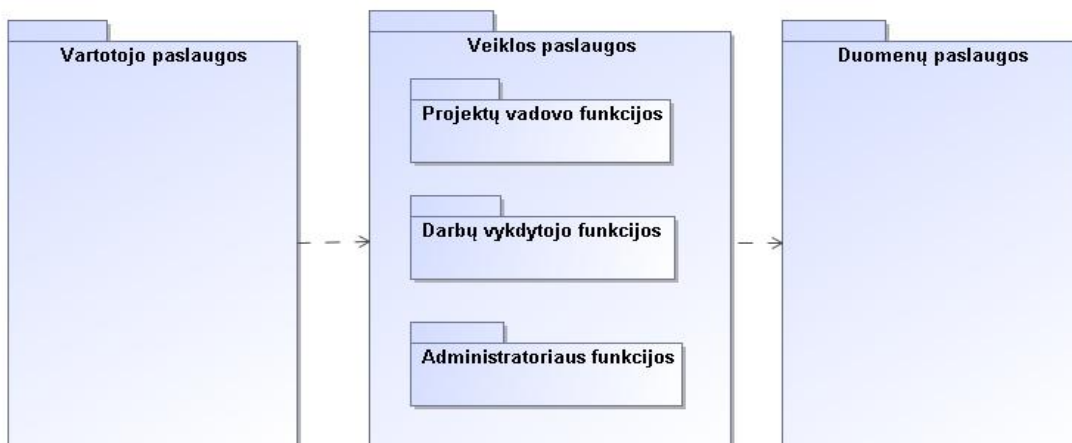
Projektų vadovas gali įvesti su sistema susijusius duomenis, čia bus registruojama teorinis, faktinis darbų apimties atlikimo laikas. Iš įkeltų reikalavimų bus atskiriami funkciniai reikalavimai, su kuriais bus atliekami tolimesni apskaičiavimai. Vykdytojai, kurie atliks reikalavimus yra sukuriami vykdytojų formoje. Funkcinių reikalavimų tipų formoje įvedami nauji, koreguojami, šalinami tipai, suvedama šių tipų parametrų informacija. Funkcinio reikalavimo tipų parametrų formoje suvedamos koeficientų reikšmės, pagal pasirinktą funkciniam reikalavimui tinkamą tipą. Įkainių lange galima įvesti naują įkainį, jį redaguoti, pašalinti. Funkcinio reikalavimo priskyrimo lange pasirenkamas vykdytojas, kuris realizuos reikalavimą, priskiriamas įkainis, terminas, iki kada reikalavimas turi būti atliktas. Projektų vadovo navigavimo planas pavaizduotas 28 pav.



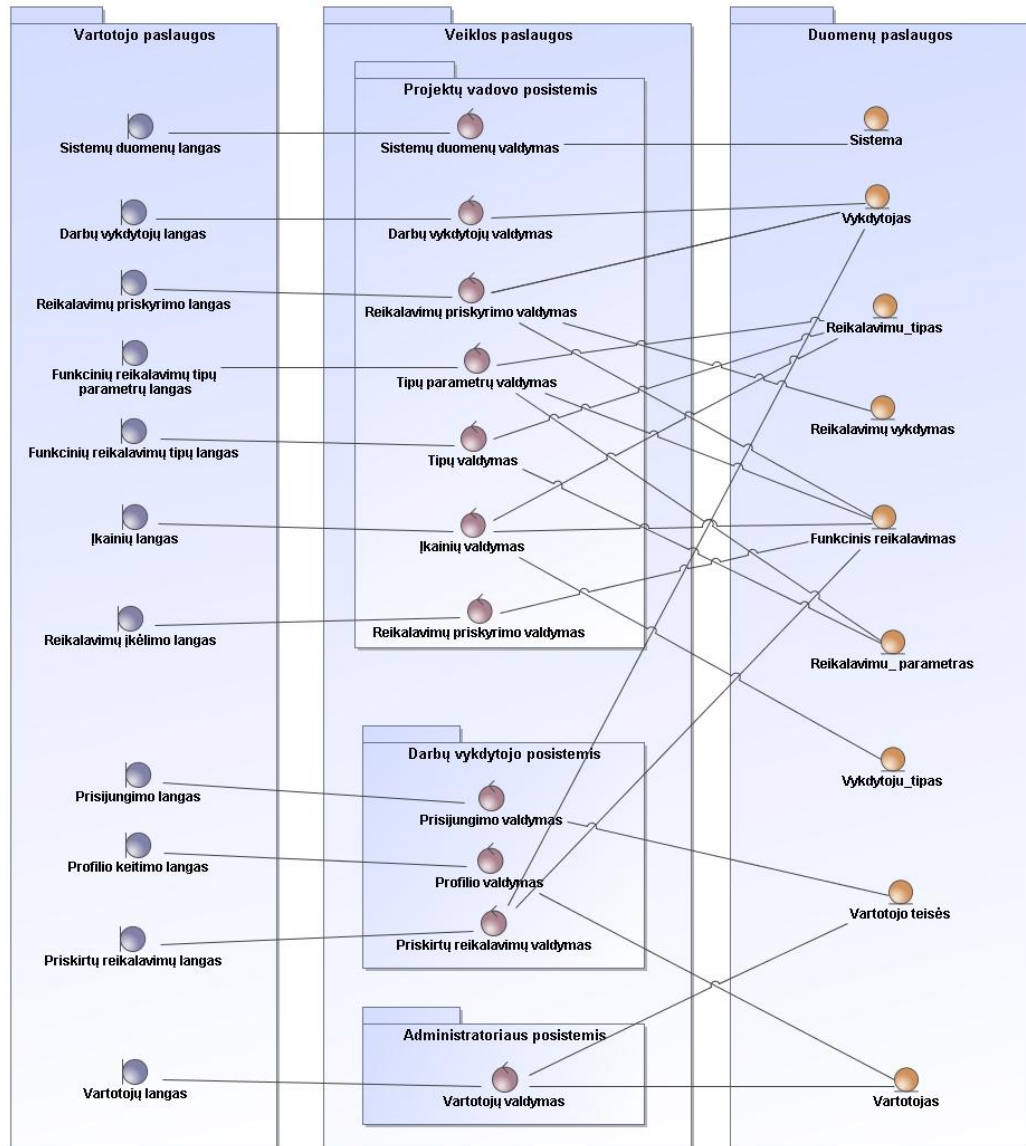
28 pav. Navigavimo planas projektų vadovui

4.1.5.2. Loginė sistemos architektūra

Naudojama trijų lygių architektūra atskiriant vartotojo, veiklos ir duomenų paslaugas (29 - 30 pav.). Vartotojo paslaugų pakete realizuojama sistemos vartotojo sąsaja, veiklos paslaugų pakete – veiklos logika. Duomenų paslaugos atspindi duomenis, kuriuos naudoja valdymo klasės.



29 pav. „FiSMA“ loginė architektūra

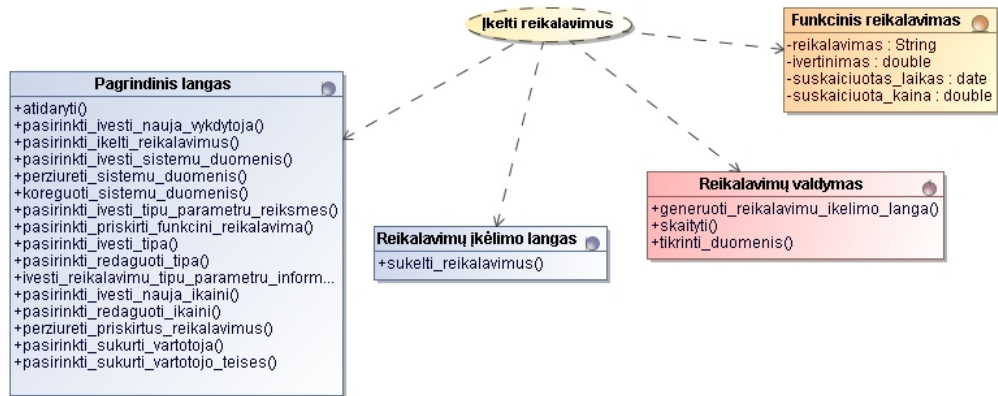


30 pav. Detali „FiSMA“ projekto loginė architektūra

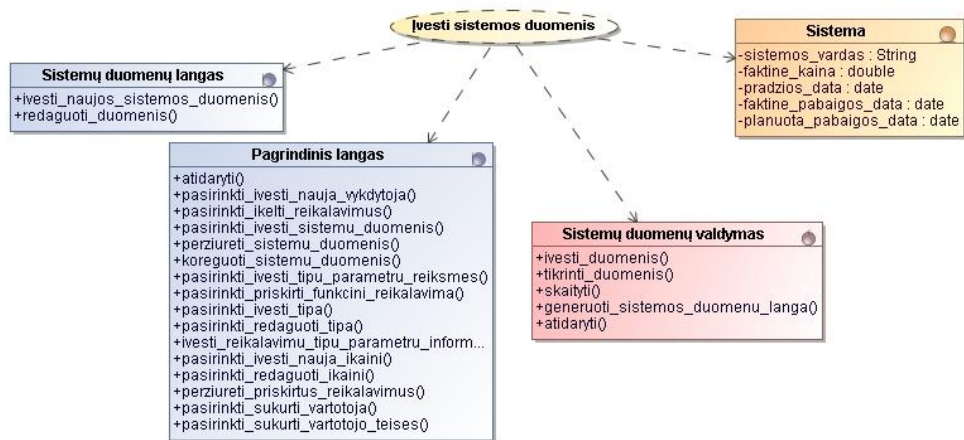
4.1.5.3. Panaudojimo atvejų realizacijos klasių diagramos

31 – 41 paveiksluose pateiktos panaudojimo atvejų realizacijos projekto klasėmis diagramos. Jos padeda išanalizuoti, kurie veiksmai bei atributai reikalingi norint realizuoti projektą.

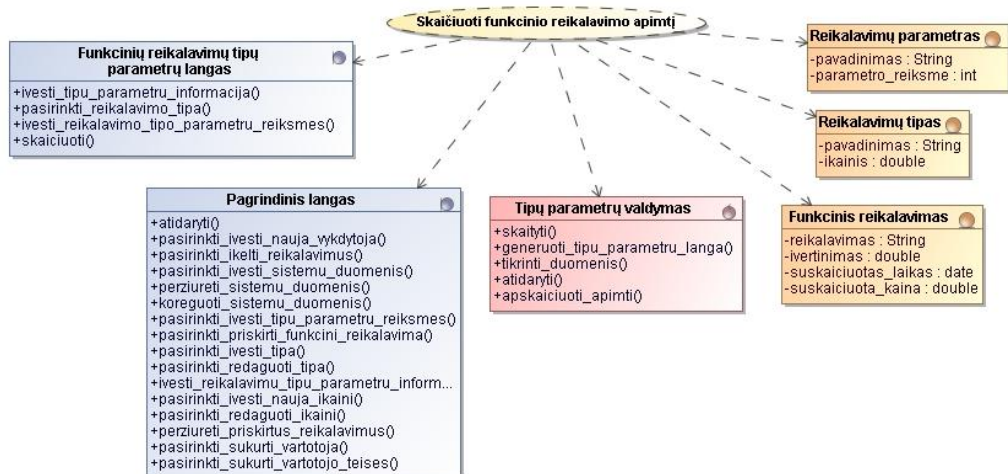
Toliau pateikiamos projektų vadovo posistemyje esančių panaudojimo atvejų „Įkelti reikalavimus“ (31 pav.), „Įvesti sistemos duomenis“ (32 pav.), „Skaičiuoti funkcinio reikalavimo apimtį“ (33 pav.), „Priskirti funkcinį reikalavimą vykdytojui“ (34 pav.), „Valdyti funkcinį reikalavimų tipų informaciją“ (35 pav.), „Valdyti įkainių informaciją“ (36 pav.), „Valdyti vykdytojų duomenis“ (37 pav.) realizacijos diagramos.



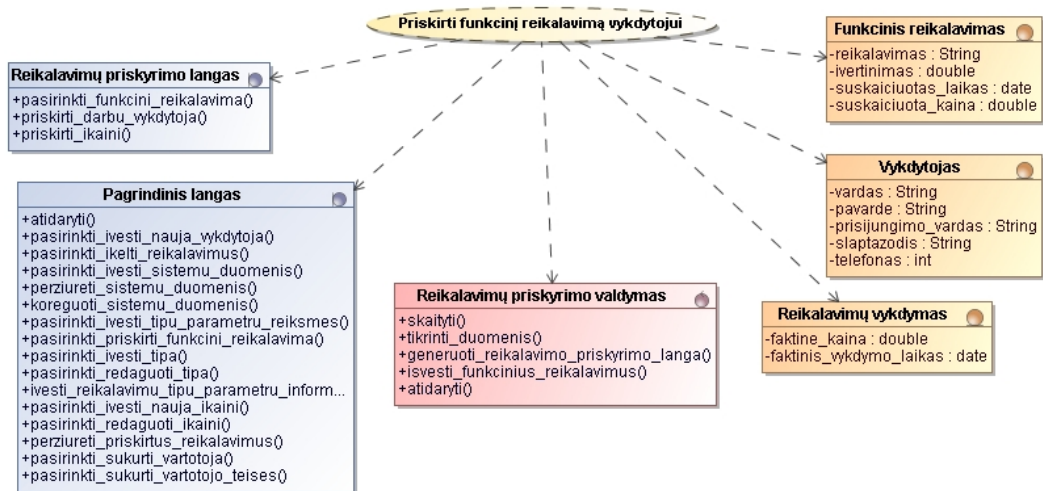
31 pav. Panaudojimo atvejo „Įkelti reikalavimus“ realizacijos diagrama



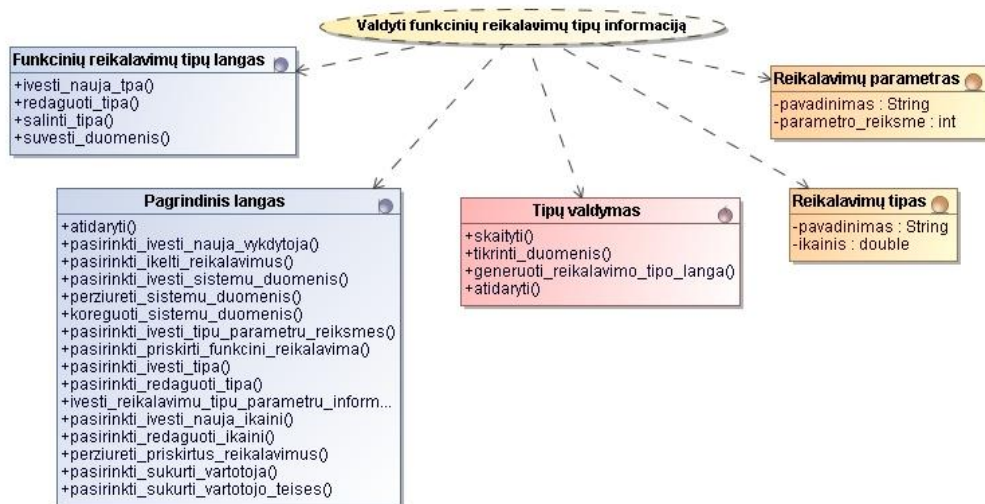
32 pav. Panaudojimo atvejo „Įvesti sistemos duomenis“ realizacijos diagrama



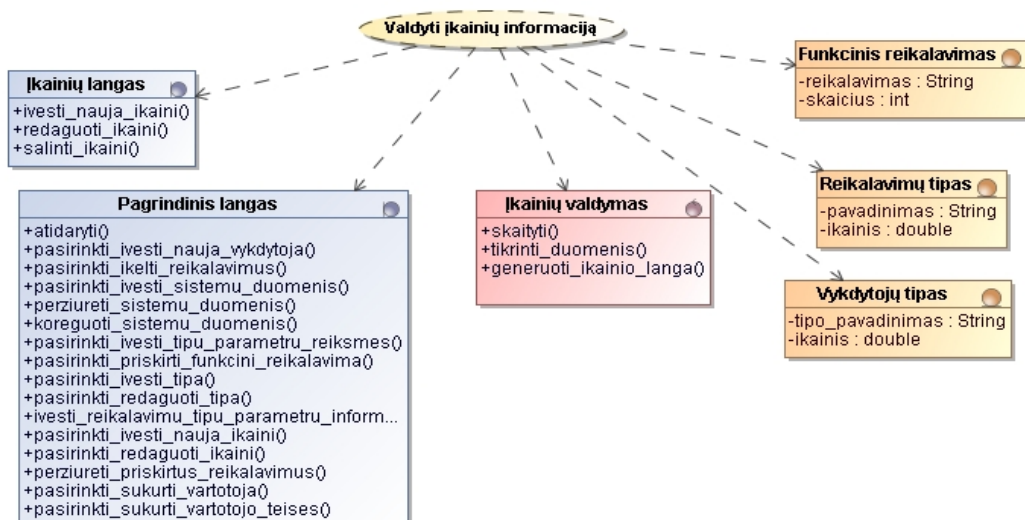
33 pav. Panaudojimo atvejo „Skaičiuoti funkcinio reikalavimo apimtį“ realizacijos diagrama



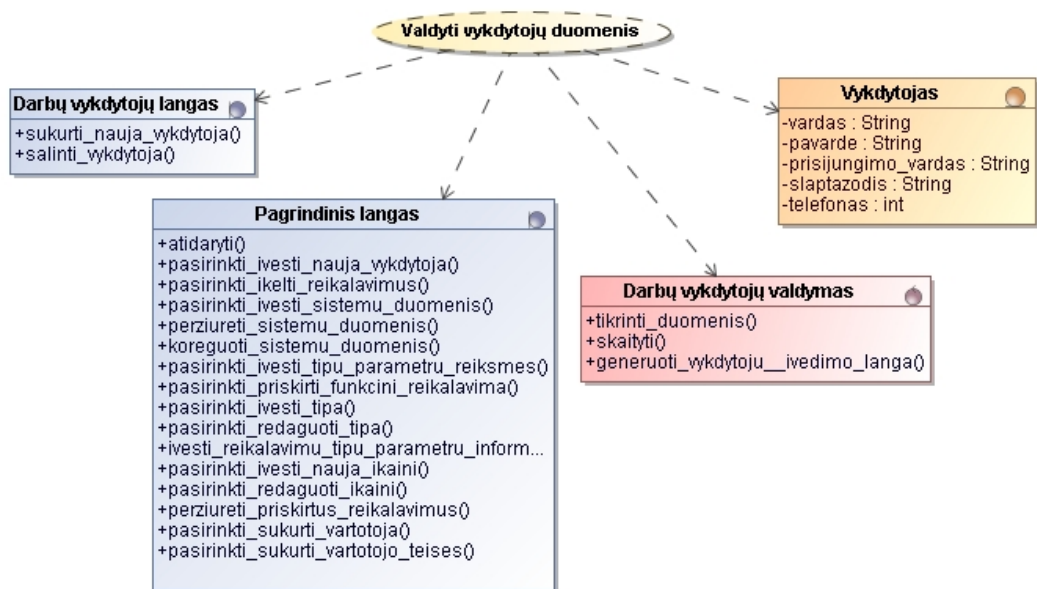
34 pav. Panaudojimo atvejo „Priskirti funkcinį reikalavimą vykdytoji“ realizacijos diagrama



35 pav. Panaudojimo atvejo „Valdyti funkcinių reikalavimų tipų informaciją“ realizacijos diagrama

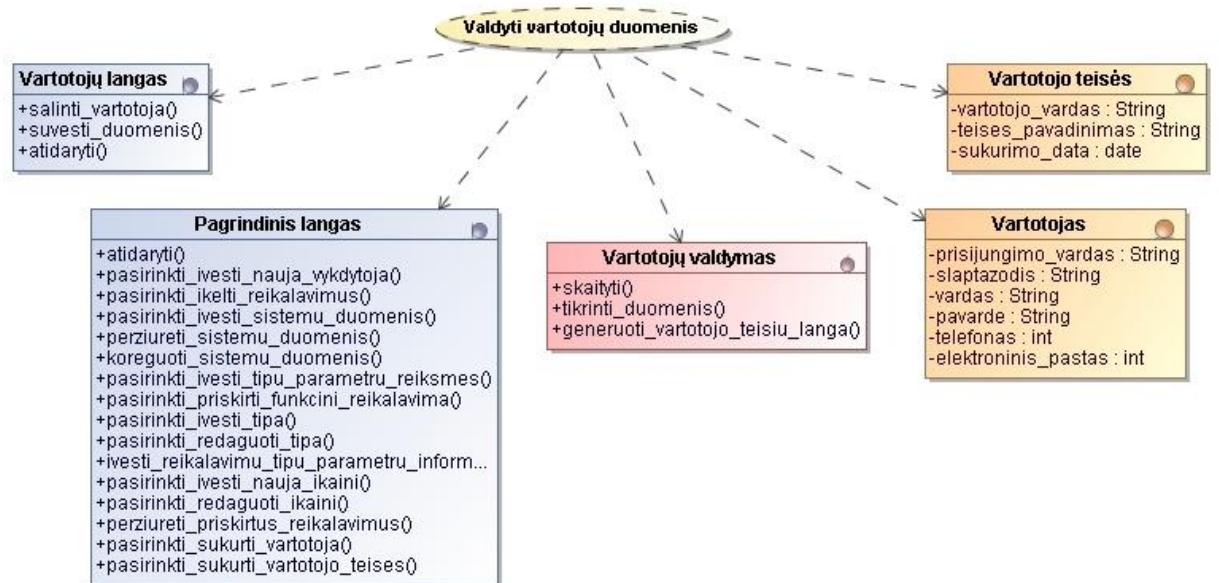


36 pav. Panaudojimo atvejo „Valdyti įkainių informaciją“ realizacijos diagrama



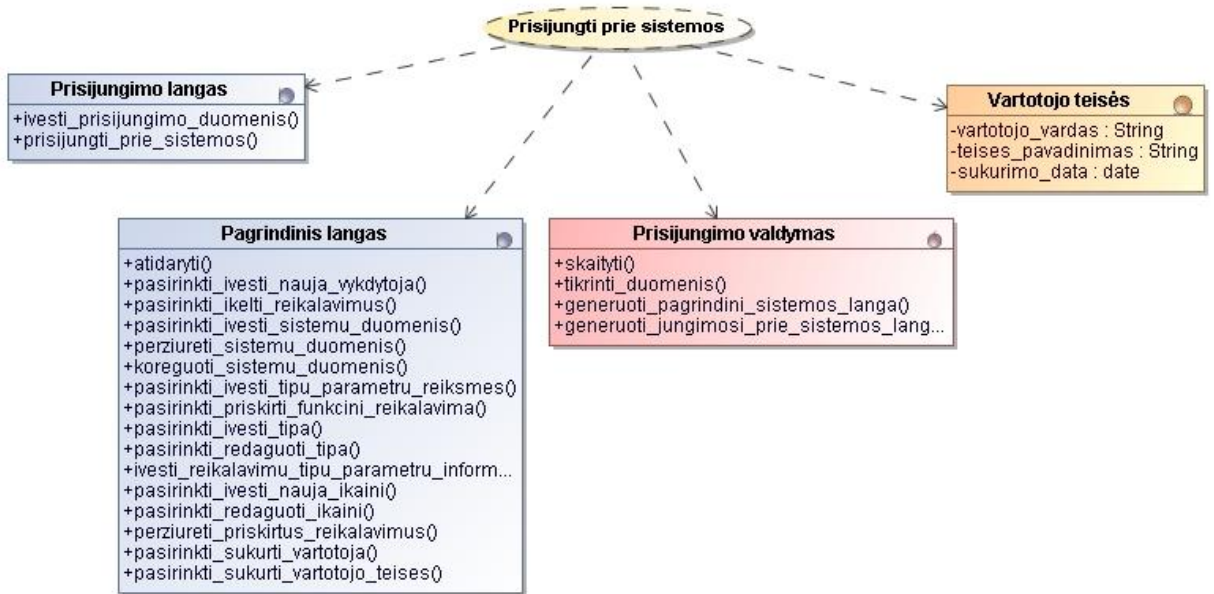
37 pav. Panaudojimo atvejo „Valdyti vykdytojų duomenis“ realizacijos diagrama

Toliau pateikiamos administratoriaus posistemyje esantis panaudojimo atvejis „Valdyti vartotojų duomenis“ (38 pav.)

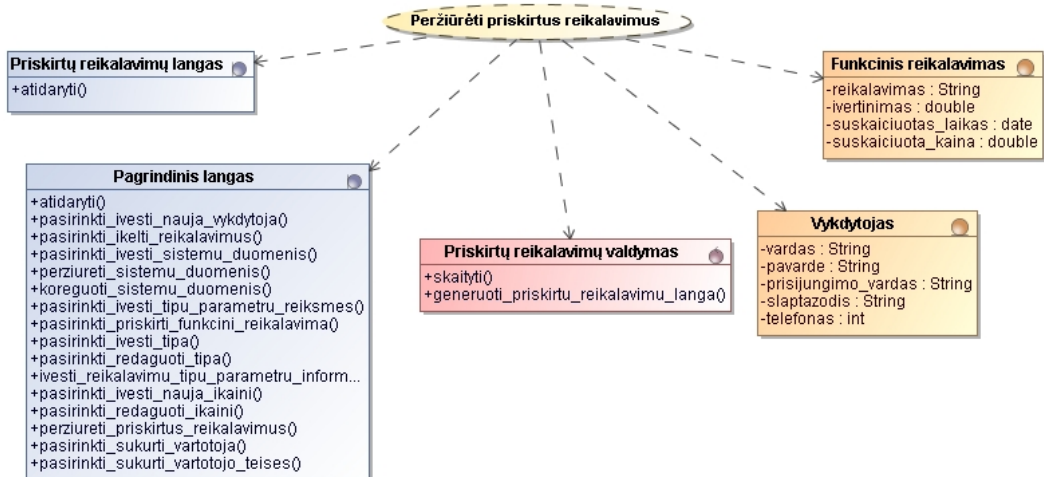


38 pav. Panaudojimo atvejo „Valdyti vartotojų duomenis“ realizacijos diagrama

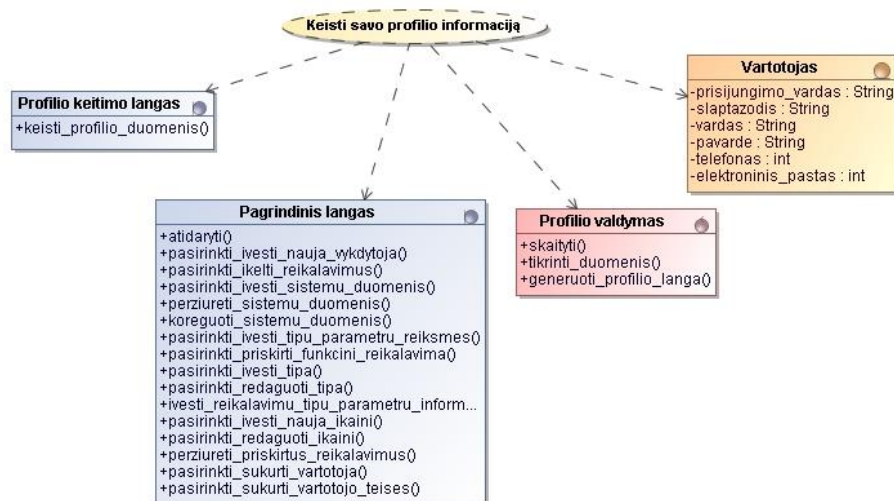
Toliau pateikiamos darbų vykdytojo posistemyje esančių panaudojimo atvejų „Prisijungti prie sistemos“ (39 pav.), „Peržiūrėti priskirtus reikalavimus“ (40 pav.), „Keisti savo profilio informaciją“ (41 pav.) realizacijos diagramos.



39 pav. Panaudojimo atvejo „Prisijungti prie sistemos“ realizacijos diagrama



40 pav. Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ realizacijos diagrama

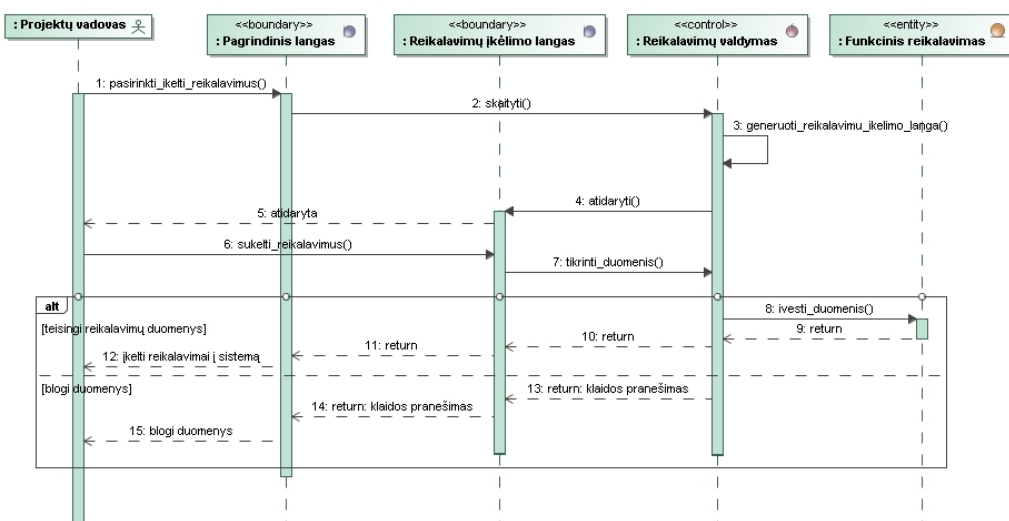


41 pav. Panaudojimo atvejo „Keisti savo profilio informaciją“ realizacijos diagrama

4.1.5.4. Sistemos panaudojimo atvejus realizuojančios sekų diagramos

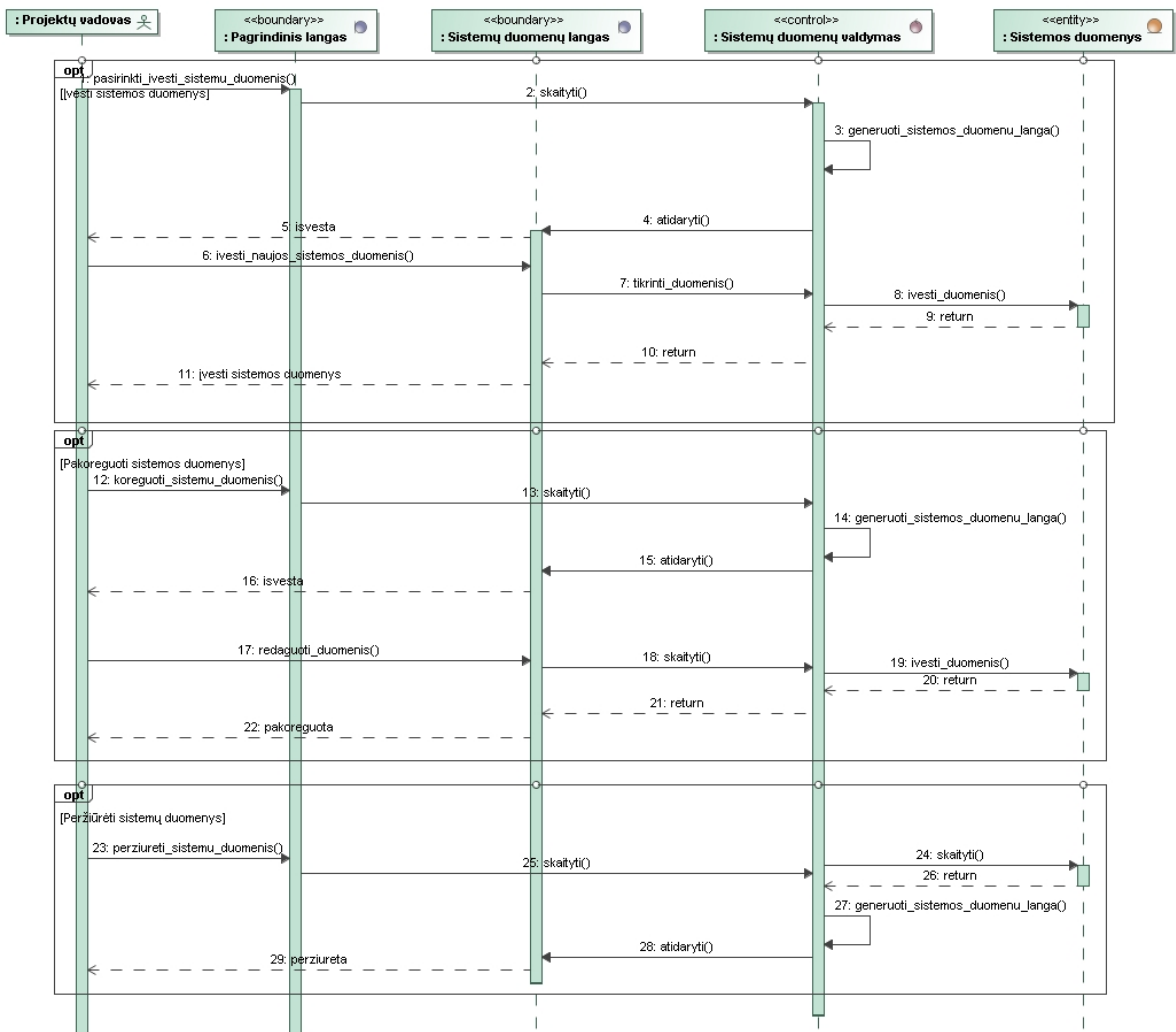
Panaudojimo atvejų sekų diagramos atvaizduoja kaip sistemos klasės dirba tarpusavyje kiekvieno panaudojimo atveju. Toliau pateikiamos sekų diagramos parodančios sistemos elgseną pasirenkant vienas ar kitas sistemos funkcijas.

Panaudojimo atvejo „Įkelti reikalavimus“ sekos diagramoje (42 pav.) matome, kaip vartotojas pasirenka įkelti reikalavimus į sistemą, generuojamas reikalavimų įkėlimo langas, vartotojas (šiuo atveju projektų vadovas) juos sukelia, sistema patikrina duomenis ir juos įveda.



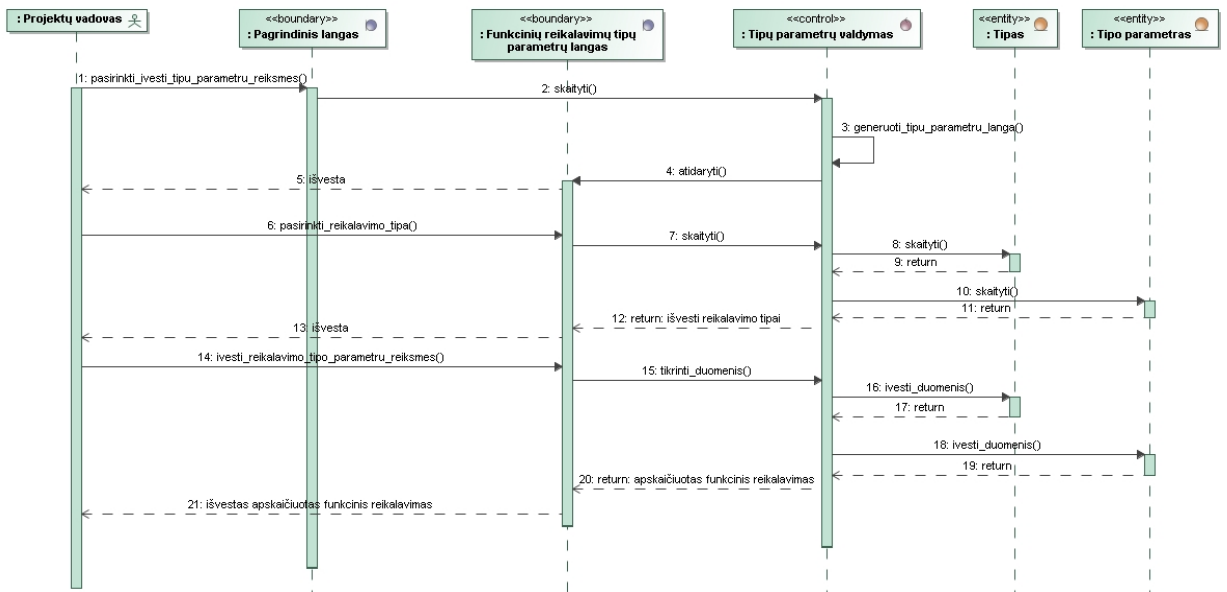
42 pav. Panaudojimo atvejo „Įkelti reikalavimus“ sekų diagrama

Panaudojimo atvejo „Įvesti sistemos duomenis“ sekos diagramoje (43 pav.) vartotojas gali įvesti naujai planuojamos realizuoti sistemos duomenis, juos pakoreguoti ar peržiūrėti. Prieš įvedant juos į sistemą jie yra patikrinami.



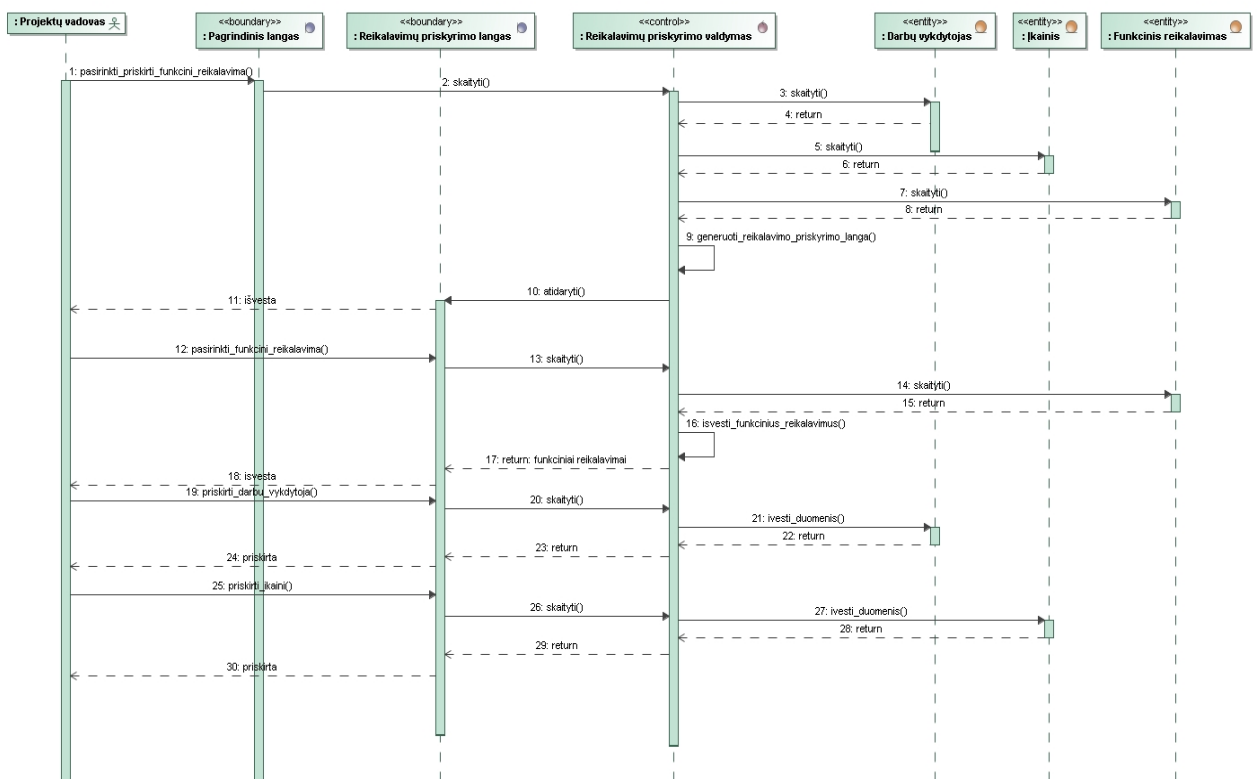
43 pav. Panaudojimo atvejo „Įvesti sistemos duomenis“ sekų diagrama

Panaudojimo atvejo „Įvesti funkcinio reikalavimo tipų parametrų reikšmes“ sekos diagramoje (44 pav.) pavaizduota, kaip generuojamas tipų parametrų langas ir įvedamos tipų parametrų reikšmės.



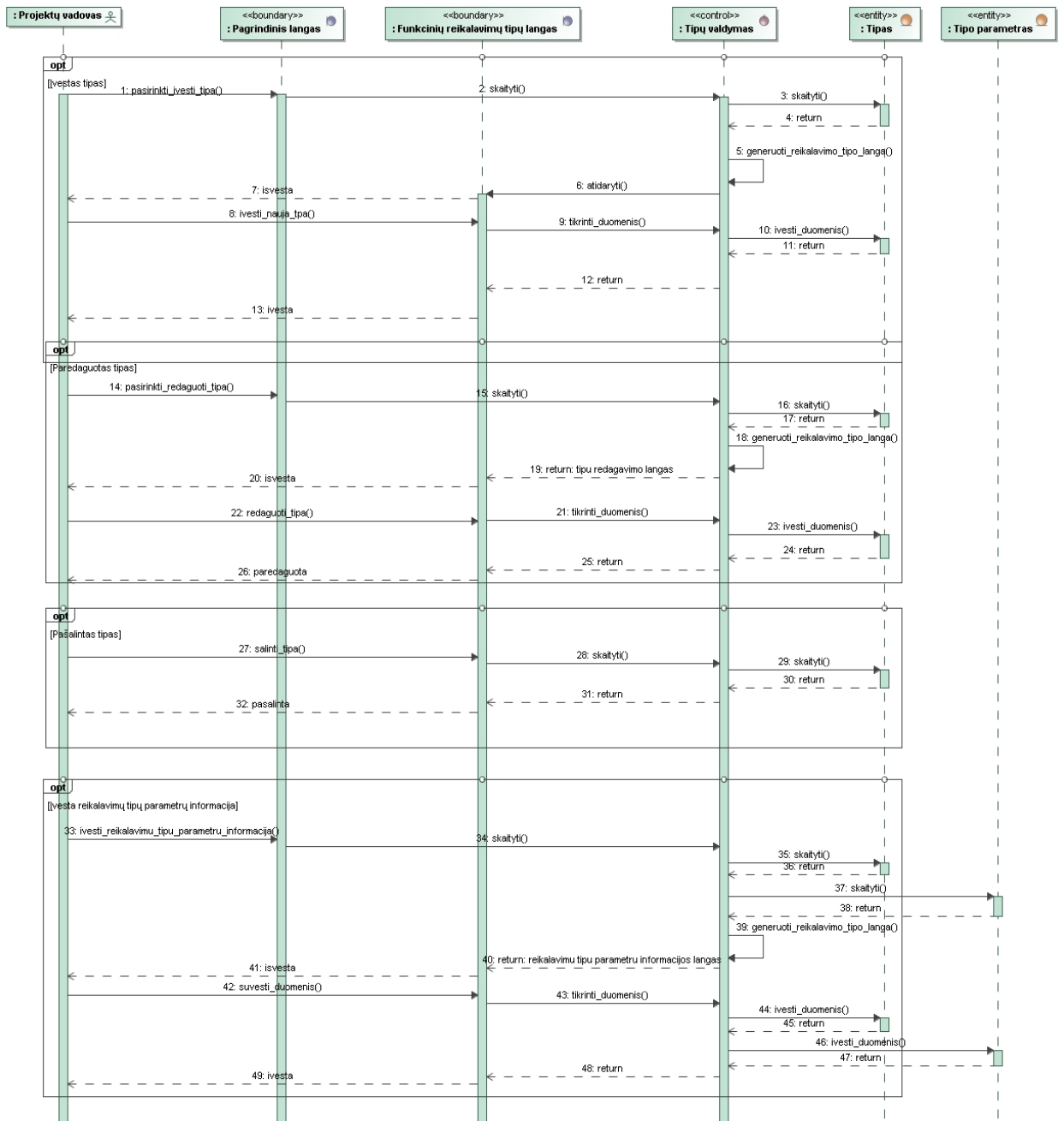
44 pav. Panaudojimo atvejo „Įvesti funkcinio reikalavimo tipų parametrų reikšmes“ sekų diagrama

Panaudojimo atvejo „Priskirti funkcinį reikalavimą vykdytojui“ sekos diagramoje (45 pav.) pavaizduota, kaip išvedami funkciniai reikalavimai juos projektų vadovas priskiria darbų vykdytojui, taip pat priskirdamas ir kainą.



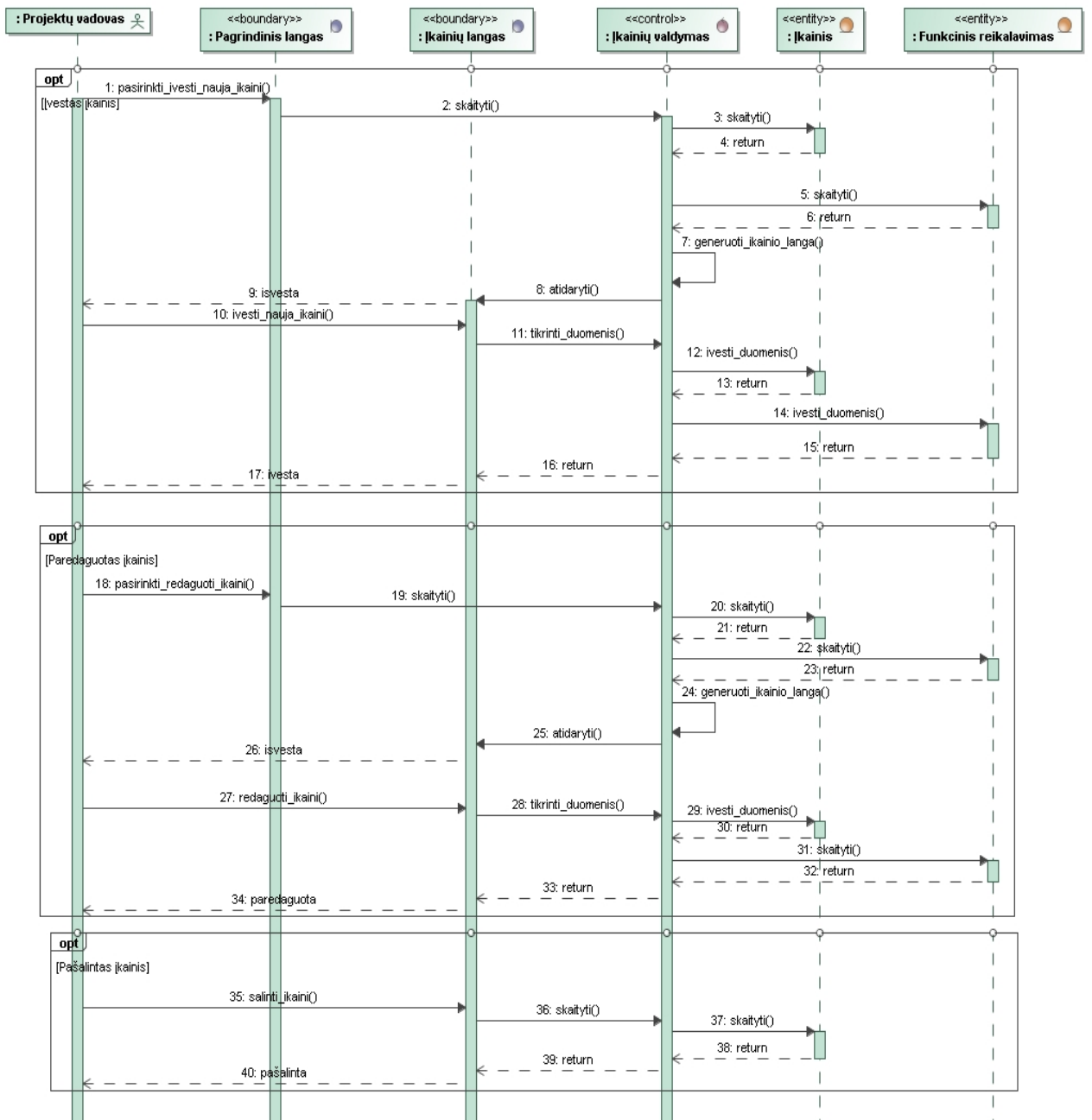
45 pav. Panaudojimo atvejo „Priskirti funkcinį reikalavimą vykdytojui“ sekų diagrama

Panaudojimo atvejo „Valdyti funkcinų reikalavimų tipų informaciją“ sekos diagramoje (46 pav.) pavaizduota galimybė įvesti naują reikalavimų tipą, jį redaguoti, pašalinti, įvesti reikalavimų tipų parametrų informaciją.



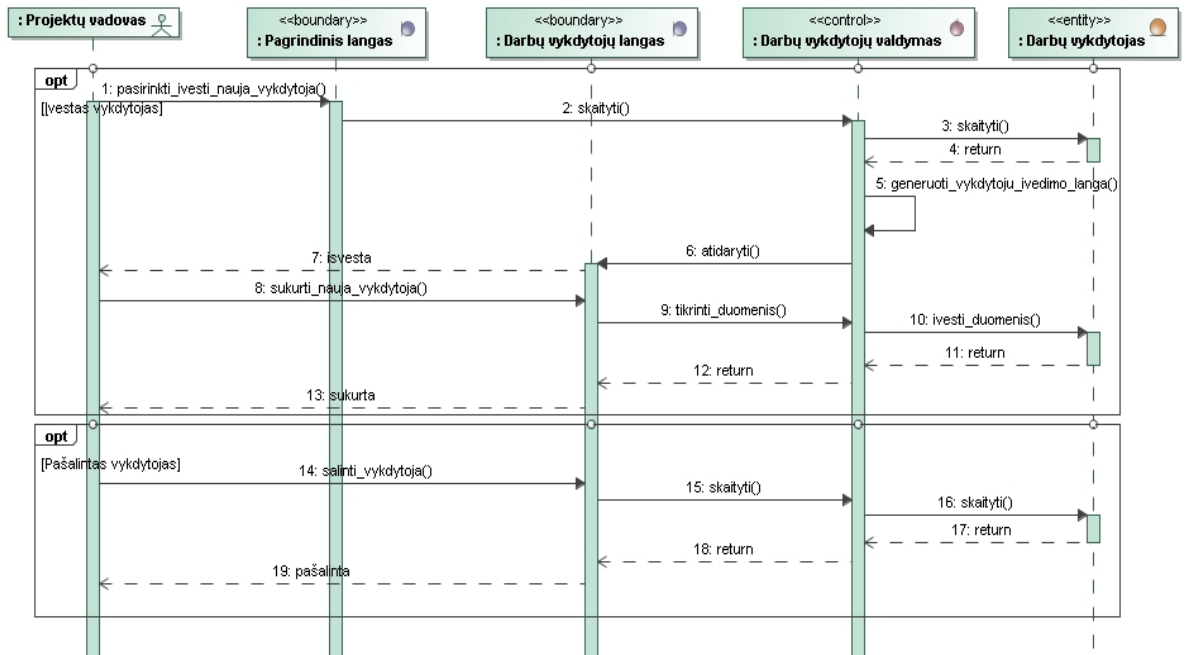
46 pav. Panaudojimo atvejo „Valdyti funkcinų reikalavimų tipų informaciją“ sekų diagrama

Panaudojimo atvejo „Valdyti įkainių informaciją“ sekos diagramoje (47 pav.) pavaizduota galimybė įvesti naują įkainį, jį redaguoti, pašalinti.



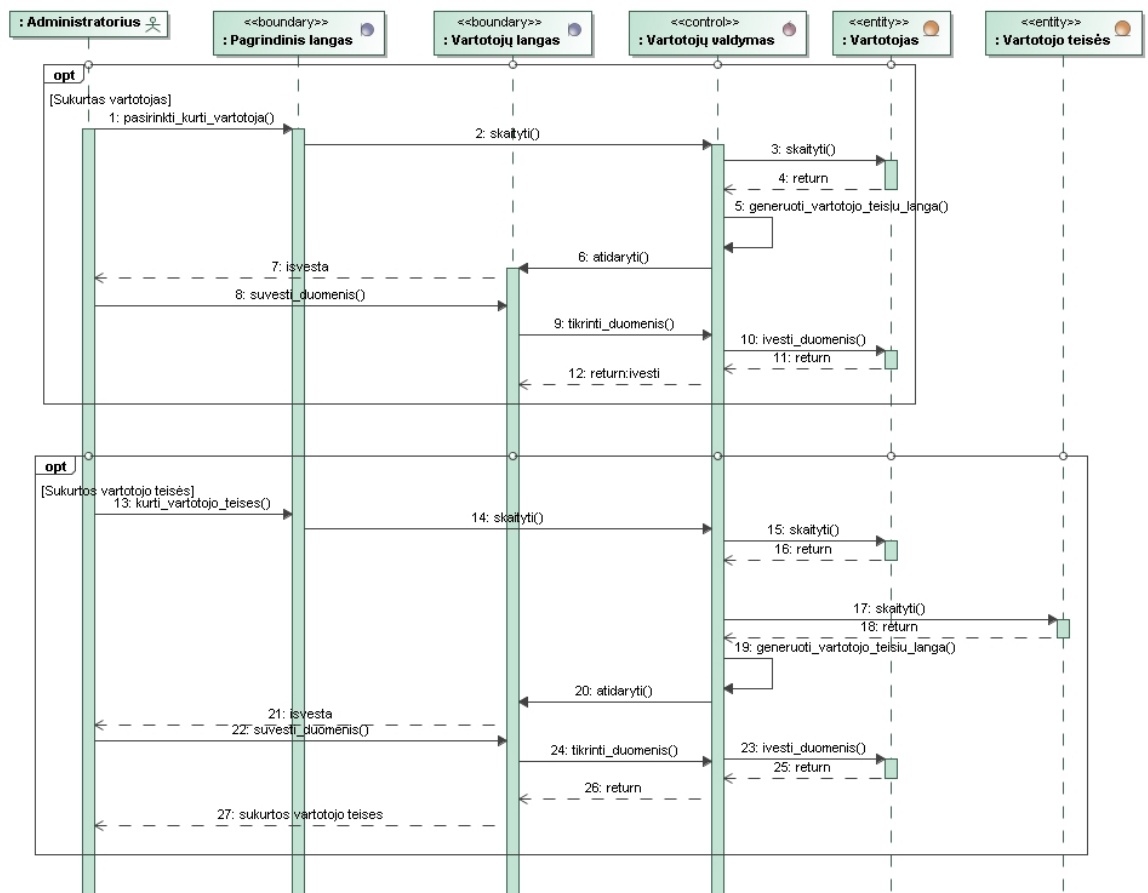
47 pav. Panaudojimo atvejo „Valdyti įkainių informaciją“ sekų diagrama

Panaudojimo atvejo „Valdyti vykdytojų duomenis“ sekos diagramoje (48 pav.) pavaizduota galimybė įvesti naują vykdytoją, jį pašalinti.



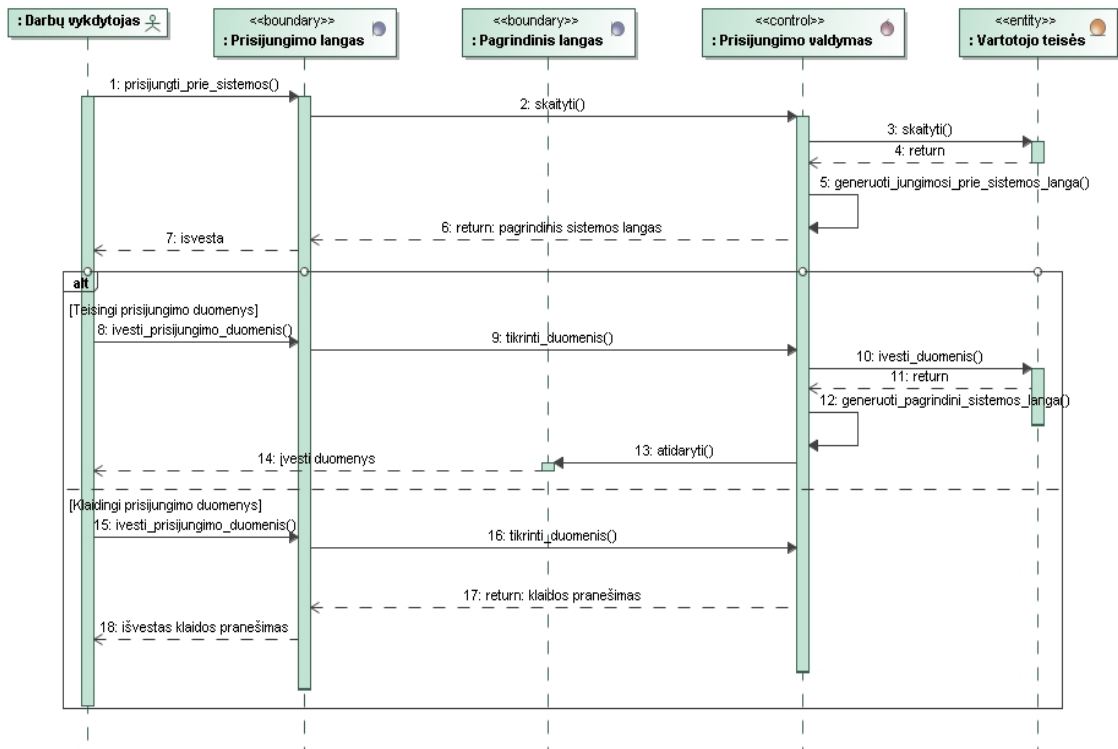
48 pav. Panaudojimo atvejo „Valdyti vykdytojų duomenis“ sekų diagrama

Panaudojimo atvejo „Valdyti vartotojų duomenis“ sekos diagramoje (49 pav.) pavaizduota galimybė sukurti naują vartotoją, jo teises.



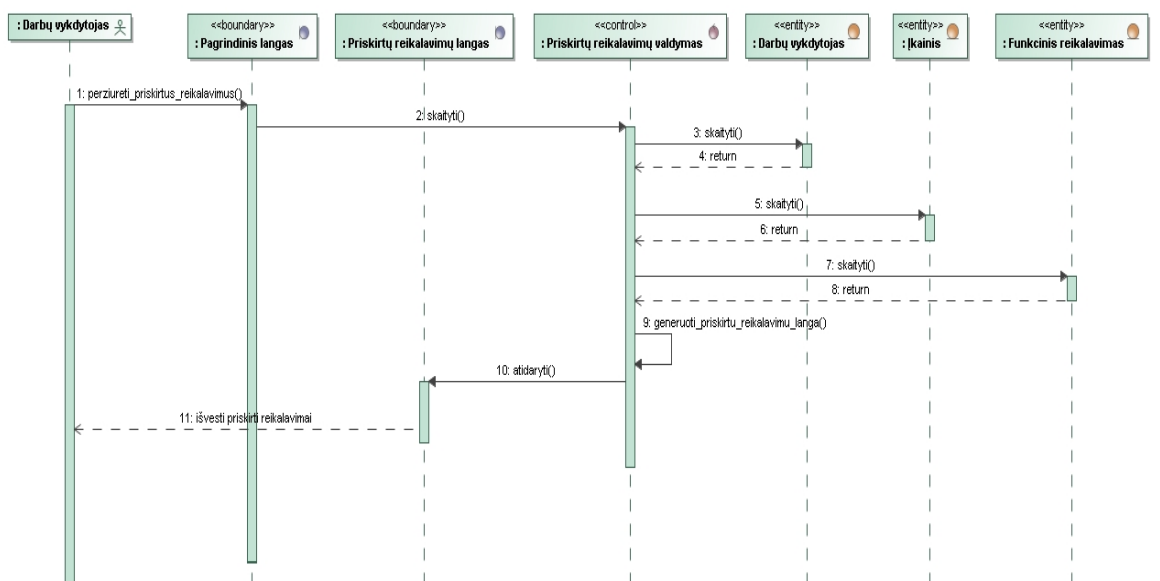
49 pav. Panaudojimo atvejo „Valdyti vartotojų duomenis“ sekų diagrama

Panaudojimo atvejo „Prisijungti prie sistemos“ sekos diagramoje (50 pav.) pavaizduota, kaip vartotojas jungiasi prie sistemos, pavaizduota sistemos reakcija į neteisingus duomenis.



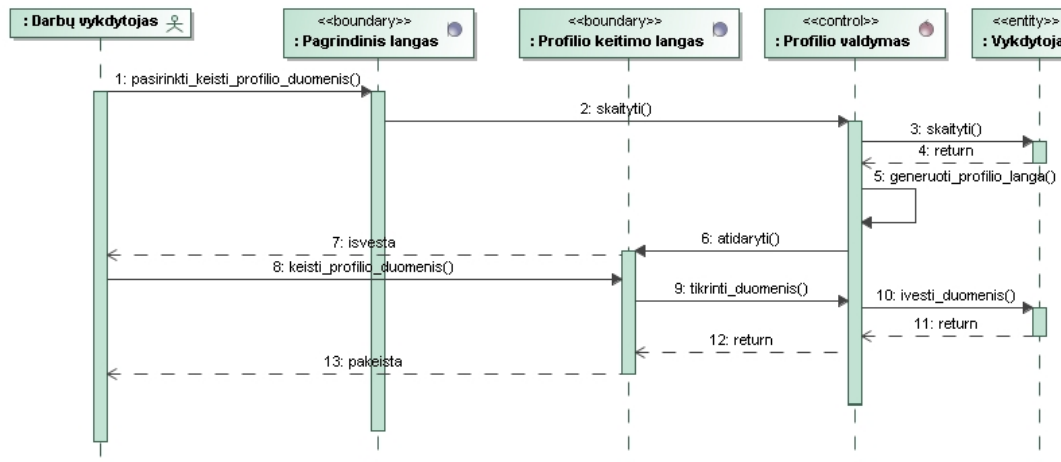
50 pav. Panaudojimo atvejo „Prisijungti prie sistemos“ sekų diagrama

Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ sekos diagramoje (51 pav.) pavaizduota, kaip darbų vykdytojas gali peržiūrėti jam priskirtus reikalavimus.



51 pav. Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ sekų diagrama

Panaudojimo atvejo „Keisti savo profilio informaciją“ sekos diagramoje (52 pav.) pavaizduota, kaip darbų vykdytojas gali keisti savo profilio duomenis.



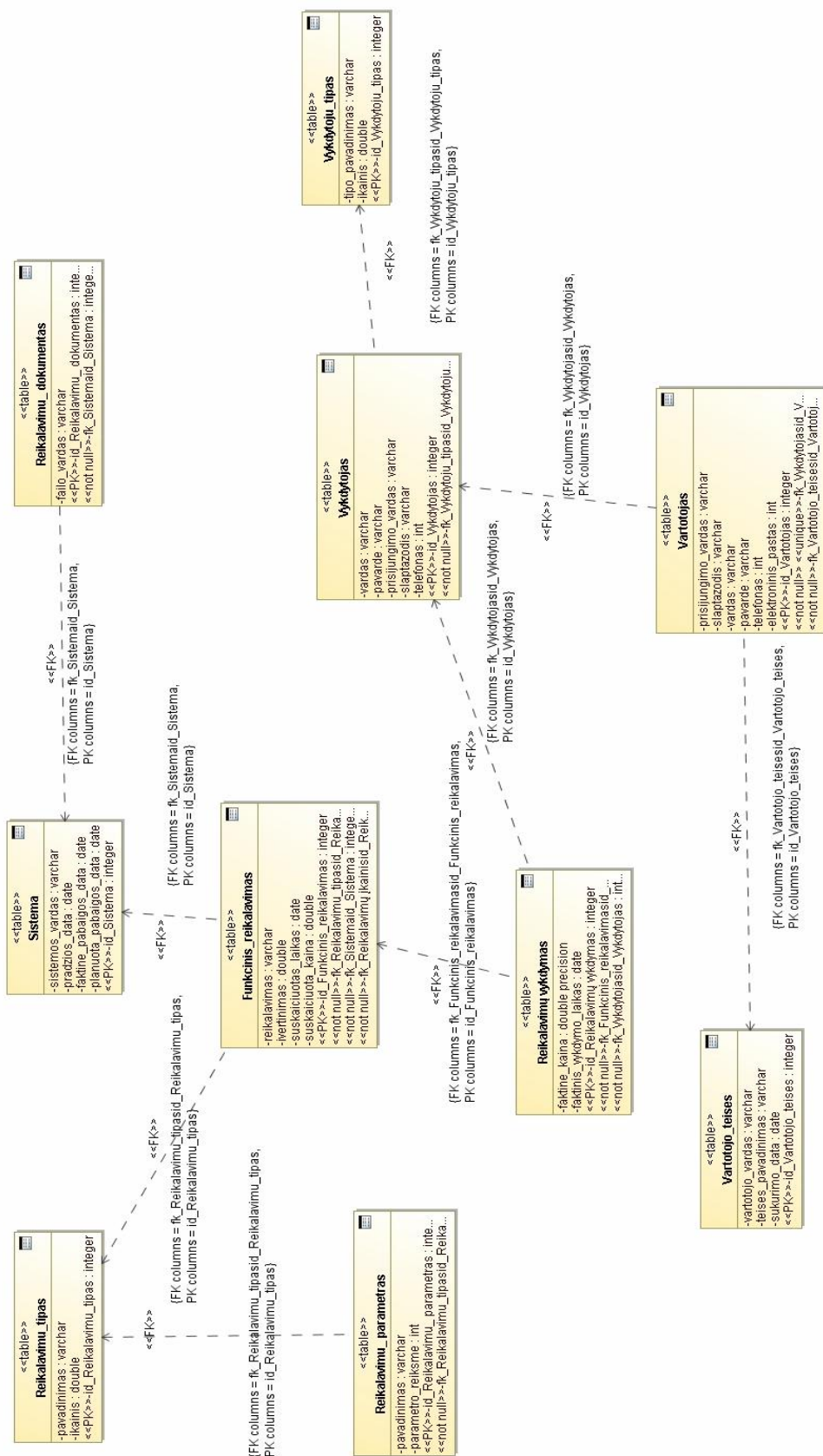
52 pav. Panaudojimo atvejo „Peržiūrėti priskirtus reikalavimus“ sekų diagrama

4.1.6. Sistemos klasių modelis

Sistemos klasių modelyje (53 pav.) matome sistemoje esančias duomenų esybes, valdiklius ir ribines klases.

4.1.7. Duomenų bazės schema

54 paveikslėlyje pateikta duomenų bazės diagrama.



54 pav. DB schema

4.2. Metodikos apibendrinimas

1. Pateiktas *FiSMA* metodikos aprašas. Aprašas padėjo įsigilinti į *FiSMA* klases ir jų tipus, skaičiavimo taisykles kiekvienam tipui, funkcinės apimties įvertinimo taikymo procesą.
2. Aprašytas detalizuotas taikymo procesas, t.y vartotojo paslaugos (navigavimo planai projektų vadovui, administratoriui, darbų vykdytojui) padėjo suprasti kokios sąsajos naujai kuriamoje sistemoje bus reikalingos.
3. Nustatyti naujai kuriamos sistemos reikalavimai pagal *FiSMA* metodiką: pateiktos panaudojimo atvejų realizacijos klasių diagramos, sistemos panaudojimo atvejus realizuojančios sekų diagramos, duomenų modelis, sudarytas sistemos architektūros modelis, suprojektuota duomenų bazės schema. Įsigilinimas į *FiSMA* metodiką padėjo suprasti šios metodikos taikymo procesą, klasių ir tipų struktūrą, kaip galima būtų panaudoti ir papildyti standartą realizuojant sistemą.

5. Sistemos realizacija pagal *FiSMA* metodiką

Šiame skyriuje pateikiamas sistemos realizacijos pagal *FiSMA* metodiką procesas, naudotojo vadovas, testavimo medžiaga.

5.1. Realizuota funkcinių reikalavimų įvertinimo ideologija

Norint tiksliau įvertinti *IS* projekto darbų apimtį realizuota informacinė sistema paremta *FiSMA* standarto ideologija. Minėtas standartas siūlo funkcinius reikalavimus priskirti atitinkamoms klasėms, kurios yra smulkinamos į tipus, šie tipai turi savo parametrus, kas įtakoja darbų apimties skaičiavimo tikslumą. *FiSMA* tikslas padėti suskaičiuoti konkretaus reikalavimo, bendros sistemos funkcinius taškus. Tačiau nėra nustatomi rėžiai, kokia funkcinio taško vertė laiko atžvilgiu. Tai yra tikslinga, kadangi sistemos gali būti skirtingo dydžio, sudėtingumo, paliekama laisvė vartotojui sekti projektų eigą, jų istoriją, kaupti statistinius duomenis, su tikslu tobulinti laiko ir funkcinių taškų tinkamumą.

Kadangi literatūroje kiti standartai ir metodikos, skirtos projekto darbų apimčiai įvertinti, turi skirtingus funkcinių taškų vertinimus laiko atžvilgiu arba jų nepateikia, norint atlikti vertinimą buvo nutarta standartą papildyti. Funkcinio reikalavimo tipai įkainoti valandiniu įkainiu, tai teorinis reikalavimo įkainis. Faktinis reikalavimo įkainis formuojamas įvedant reikalavimo tipų parametrų reikšmes. Teorinis laikas nurodomas prie tipo priskiriant kiek bus užtrukta laiko. Faktinis laikas – kuomet fiksuojamas realiai atliktas darbas, jo trukmė.

5.2. Realizuotas funkcinių reikalavimų įvertinimo procesas

Informacinėje sistemoje *FiSMA* sukuriama nauja sistema, kuriai bus atliekamas laiko ir kainos vertinimas. Įkeliami suformuoti pagal *Volere* šabloną funkciniai reikalavimai. Funkcinio reikalavimo pavyzdys pateikiamas 55 pav.:

Reikalavimas: 1	Reikalavimo tipas:	9	Įvykis/panaudojimo atvejis: 1
Aprašymas: Įmonių sąskaitos-faktūros peržiūrėjimas.			
Pagrindimas: Organizacijai, išorinėms institucijoms reikalinga dokumentų apskaita, kad už kokį renginį buvo sumokėta, sisteminti su įmone susijusius duomenis.			
Šaltinis: Analitikas, administratorius			
Tinkamumo kriterijus: Įvedant paieškos duomenis išvesti atliktų pirkimų sąskaitų-faktūras.			
Teorinis reikalavimui skiriamas laikas: 5h			
Užsakovo patenkinimas:	5	Užsakovo nepatenkinimas:	5
Priklausomybės:		Konfliktai:	
Papildoma medžiaga:			
Istorija: 2008.03.22			

55 pav. Funkcinis reikalavimas "Įmonių sąskaitos – faktūros peržiūrėjimas" pateiktas Volere šablonu, paruoštas kėlimui į „FiSMA“ sistemą

Į sistemą yra tikslas įkelti aprašymo laukelį, kuriame nurodomas funkcinio reikalavimo pavadinimas. Šiam tikslui buvo panaudota atviro kodo projekto bendruomenės (*Open Source Project Community - CodePlex*) biblioteka *DocX.dll*, kurios pagalba atidaromi MS Word 2007 failai [14]. Jo veikimo principas: eiti per paragrafus ir ieškoti teksto „Aprašymas:" ir paimiti funkcinio reikalavimo pavadinimą.

Tolesnėje realizacijos veiksmų sekoje pasirinktam funkciniam reikalavimui yra nustatoma atitinkama klasė. Išvedami klasei priklausantys tipai, suvedus šių tipų parametrus ir laiką, kiek prie kiekvieno planuojama užtrukti, skaičiuojami ir pateikiami funkciniai taškai. Kiekviena funkcinio reikalavimo klasė yra įvertinama valandiniu įkainiu, kuris yra dauginamas iš valandų skaičiaus, priskirto konkrečiam klasės tipui. Taip yra gaunamas apskaičiuotas laikas ir kaina pagal *FiSMA* klasių klasifikaciją, kuomet projekto vadovas įveda funkcinio reikalavimo klasių įkainį ir jų atlikimo laiką. Faktinis laikas ir kaina apskaičiuojama kuomet projektų vadovas priskiria konkretų funkcinį reikalavimą darbų vykdytojui, prieš tai priskyres vykdytojui tipą, kuris įkainojamas taip pat valandiniu įkainiu. Kuomet darbų vykdytojas atlieka pavestus darbus, fiksuojamas faktinis darbo laikas valandomis ir kaina, kiek funkcinis reikalavimas kainavo viso projekto kontekste.

5.3. Naudotojo vadovas

Prie sistemos prisijungti ir su ja dirbti gali tik registruoti sistemos naudotojai. Kadangi šioje sistemoje išskiriamos 3 vartotojų grupės: projektų vadovai, darbų vykdytojai ir darbų vykdytojai, todėl jie turi 3 skirtingas prisijungimo sąsajas.

Projektų vadovas prisijungęs prie sistemos sąsajos gali dirbti su naujai kuriamais projektais, įkeldamas į sistemą reikalavimus, priskirdamas reikalavimams klases, nurodydamas tipų reikšmes.

Funkcinio reikalavimo įvertinimo langas pateiktas 56 paveiksle.

The screenshot shows a window titled 'Vartotojo profilio informacijos keitimo galimybė'. It contains a tab 'Funkcinis reikalavimas' with a sub-tab 'Vykdytojai'. The main content area displays the following details:

- Reikalavimas: Vartotojo profilio informacijos keitimo galimybė.
- Klasė: Interaktyvios galinio vartotojo na...
- Ivertinimas (Ffp): 11.7571428571429
- Laikas (valandomis): 9
- Kaina: 305

Parametras	Reikšmė
Pradžios ikonos	
Duomenų elementų skaičius, laukai	2
Esybių skaitymų nuorodų skaičius	5
Užtrukas (valandas)	1
Prisijungimo ir atsijungimo langai	
Duomenų elementų skaičius, laukai	6
Esybių skaitymų nuorodų skaičius	4
Užtrukas (valandas)	2

56 pav. „FiSMA“ Konkretaus funkcinio reikalavimo detalių langas

Įvertintą funkcinį reikalavimą projektų vadovas gali priskirti darbų vykdytojui, parenkant darbų vykdytojo tipą. Vykdytojo tipo priskyrimo langas pateiktas 57 paveiksle:

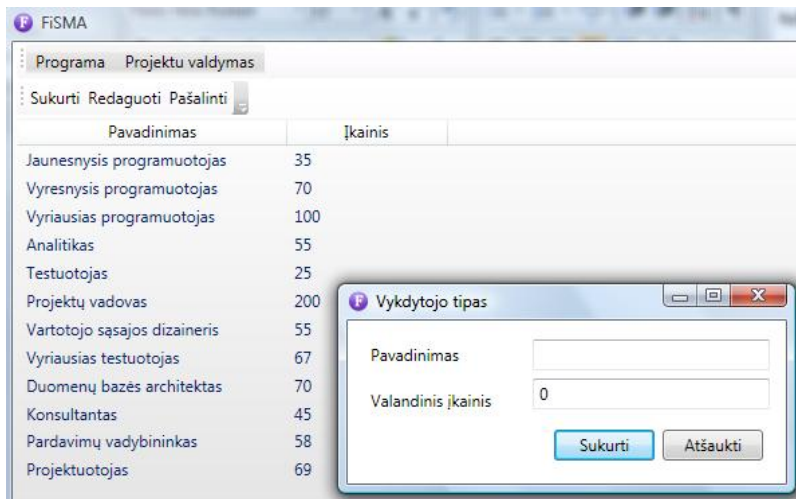
The screenshot shows the same window as above, but with a dialog box titled 'Priskirti vykdytoją' open. The dialog box contains the following fields:

- Tipas: Jaunesnysis programuotojas
- Vykdytojas: A dropdown menu with the following options: jonas, kazys, marija, martynas, petras, worker, zigmas.

57 pav. „FiSMA“ vykdytojo priskyrimo detalių langas

FiSMA metodika buvo papildyta vykdytojų tipų priskyrimu. Šie tipai yra reikalingi tam, kad apibrėžti valandinį įkainį priklausomai nuo to kokiam vartotojo tipui priskirtas darbu vykdytojas. Nuo šio faktoriaus priklauso funkcinio reikalavimo faktinė kaina.

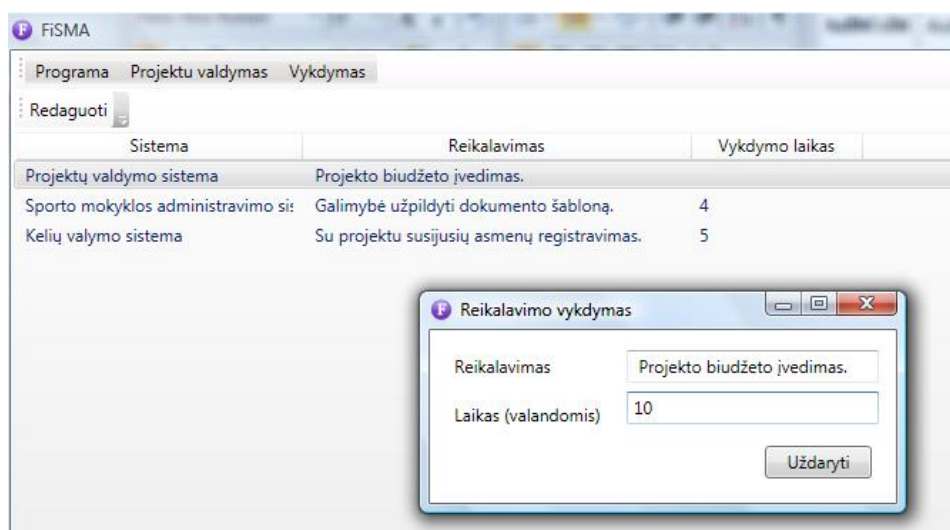
Vykdytojo tipo kūrimo langas pateikiamas 58 paveiksle.



58 pav. „FiSMA“ vykdytojo tipo kūrimo detalių langas

Taip pat sistema buvo papildyta reikalavimų tipų įkainiais. Šiems tipams taip galima parinkti tinkamą įkainį.

Kadangi projektų vadovas veda spėjamąjį laiką, prisijungęs darbu vykdytojas fiksuoja prie reikalavimo užgaištą faktinį laiką, jei iškyla nesklandumų su reikalavimu ir reikalingi reikalavimo pataisymai, reikalavimo laiką galima koreguoti. 59 paveiksle pateikiama faktinio laiko vedimo sąsaja.



59 pav. „FiSMA“ Naujos sistemos kūrimo detalių langas

Išsamesnis naudotojo vadovas yra pateikiamas **10.1. Priede. Naudotojo vadovas**

5.4. Testavimo medžiaga

Vartotojo sąsajos testavimas atliekamas pagal „juodos dėžės“ principą užbaigtam produktui.

Sistema buvo nuodugniai ištestuota pagal reikalavimų specifikaciją ir patikrinta. Radus neatitikimą tarp sistemos ir vartotojo poreikių buvo patikrinta reikalavimų specifikacija.

Testavimo resursai

Jokių papildomų techninių ar programinių resursų, kurie viršytų jau turimus resursus skirtus sistemos kūrimui, nereikia. Testavimo darbams buvo naudojamas asmeninis programinės įrangos kūrėjos kompiuteris.

Testavimo atvejai

Skyriuje pateikiami testavimo atvejai, kurie buvo naudojami atliekant programinės įrangos testavimą.

Vartotojo sąsajos (UI) detalesnis testavimas

Žemiau lentelėse pateikiami testavimo atvejai.

18 lentelė. Prisijungimo lango testavimo atvejai

Veiksmas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Įvedami teisingi prisijungimo duomenys	Vartotojas = dzulija Slaptažodis= zzzz1234!	Parodomas pagrindinis programos langas, pagal tai kokios rolės priskirtos naudotojui, tokie atitinkami meniu punktai pateikiami.	Prie sistemos prisijungiama
Įvedamas neteisingai prisijungimo vardas	Vartotojas = jolanta Slaptažodis= zzzz1234!	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Įvedamas neteisingai slaptažodis	Vartotojas = jolanta Slaptažodis= zzzz	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Įvedami neteisingai abu laukeliai	Vartotojas = mantas Slaptažodis= kkkk	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Į sistemą prisijungimo duomenys nevedami, spaudžiamas uždaryti mygtukas.	-	Programa užsidaro.	Baigiamas darbas.

Veiksmas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Pasirenkamas meniu punktas „Vykdymo tipai“	Paspaustas „Vykdymo tipų“ meniu	Atidaromas vartotojų tipų langas. Jame parodomas įvestų vykdytojų tipų sąrašas.	Atidaromas vykdytojų tipų langas
Pasirenkamas meniu punktas „Reikalavimų tipų įkainiai“	Paspaustas „Reikalavimų tipų įkainiai“ meniu	Atidaromas reikalavimų tipų langas. Jame parodomas reikalavimų tipų įkainių sąrašas.	Atidaromas reikalavimų tipų įkainių langas
Pasirenkamas meniu punktas „Sistemos“	Paspaustas „Sistemos“ meniu	Atidaromas sistemų langas. Jame parodomas sistemų sąrašas.	Atidaromas sistemų langas
Pasirenkamas meniu punktas „Naudotojai“	Paspaustas „Naudotojai“ meniu	Atidaromas naudotojų langas. Jame parodomas įvestų naudotojų sąrašas.	Atidaromas naudotojų langas
Pasirenkamas meniu punktas „Priskirti reikalavimai“	Paspaustas „Priskirti reikalavimai“ meniu	Atidaromas priskirtų reikalavimų langas.	Atidaromas priskirtų reikalavimų langas
Pasirenkamas meniu punktas „Išeiti“ arba langas uždaromas	Paspaustas „Išeiti“ meniu	Programa užsidaro.	Programa užsidaro.
Pasirenkamas meniu punktas „Atsijungti“	Paspaustas „Atsijungti“ meniu	Atsijungiama nuo sistemos, galima alternatyva prisijungimui prie kt. vartotojo prieigos.	Atsijungiama nuo sistemos.

Išsamesnė testavimo medžiaga yra pateikiama **10.2. Priede. Testavimo medžiaga**

5.5. Sukurtos sistemos apibendrinimas

1. Pateiktas sistemos funkcinių reikalavimų įvertinimo procesas. Buvo nutarta sutrumpinti apimties skaičiavimo procesą, realizuojant funkcinio reikalavimo įkėlimą pagal pateiktą šabloną.
2. Išsamiai aprašytas naudotojo vadovas, su tikslu projektuotojui lengviau įsisavinti sistemos veikimą.
2. Testavimo metu buvo stengtasi atrasti vietų, kurias galima tobulinti, kad standarto *FiSMA* skaičiavimo procesą pateikti suprantamesnį ir intuityvesnį. Testuojant buvo nuspręsta sistemos funkcionalumą praplėsti sukūrus ataskaitą, kurioje bus galima stebėti reikalavimų tipų vidurkius, t. y bus kaupiami reikalavimų tipų funkcinių taškų įvertinimo vidurkiai, teoriniai, faktiniai laikai ir teorinės,

faktinės kainos. Šioje ataskaitoje įvertindamas naujus funkcinius reikalams projektų vadovas galės stebėti kintančias su tipais susijusias vertes.

4. Testavimas buvo atliekamas nuolat, t. y. testuojami atskiri sistemos elementai, vėliau jų integracija tarpusavyje. Dauguma defektų buvo pašalinta programavimo fazės metu. Atliktas grafinės vartotojo sąsajos detalus testavimas, sistema buvo nuodugniai testuojama pagal reikalavimų specifikaciją. Tačiau testavimas dar nėra baigtinis, kadangi atsiradus naujai ataskaitai, bus reikalinga suvesti nemažai duomenų, kad būtų galima stebėti duomenų pasiskirstymą. Atliekant eksperimentą, gali iškilti poreikis sistemą koreguoti.

6. Eksperimentinis sistemos realizuotos pagal FiSMA metodiką tyrimas

6.1. Eksperimento tikslas

Eksperimentas atliekamas siekiant nustatyti:

1. pradinę funkcinio taško vertę, kurią būtų galima panaudoti skaičiuojant kitų projektų vykdymo laiką;
2. projekto elementų skaičiavimo laiką ir jo santykį su visa projekto trukme, t. y. patikrinti, ar šis laikas nėra pernelyg didelis.

Taip pat siekiama išbandyti sukurtą sistemą ir įsitikinti, ar ji gali padėti įvertinti informacinių sistemų projekto darbų apimtį pagal *FiSMA* standartą ir stebėti, kiek skaičiuojama projektų realizavimo trukmė skiriasi nuo faktinės, siekiant patikslinti *FiSMA* įvertinimus pagal sukauptų projektų duomenis.

Eksperimentas turi padėti įvertinti, ar šis įrankis gali palengvinti darbą projektų vadovui. Eksperimento metu su tam tikru projektų skaičiumi siekiama sužinoti, kokia šiems projektams susidarys vieno funkcinio taško vertė laiko atžvilgiu. Žinant bendrą projekto funkcinį taškų sumą ir projekto trukmę bus gauta 1 funkcinio taško reikšmė. Kadangi projekto darbų apimties įvertinimo metodikos yra svarbi projekto planavimo dalis reikia įvertinti ar laikas skirtas funkcinio reikalavimų suvedimui ir projekto elementų suskaičiavimui nėra per ilgas ir sudėtingas atsižvelgiant į projekto numatomą realizavimo trukmę.

6.2. Eksperimento apibrėžimas

Eksperimentas bus vykdomas dviem etapais, jo planas atsispindi 20 lentelėje.

20 lentelė. Eksperimento vykdymo etapai

Etapas	Aprašymas
1. Spėjamo projekto vykdymo laiko nustatymas	Į sistemą įvedami spėjami projektų vykdymo laiko įverčiai. Fiksuojamas projekto elementų skaičiavimo laikas.
2. Faktinio projekto vykdymo laiko nustatymas	Įvedamos faktinės projekto vykdymo laiko vertės. Lyginami spėjami ir faktiniai rezultatai.
3. Funkcinio taško vertės nustatymas pagal tipus.	Pagal funkcinio taško vertes ir laiką užgaištą prie jų apskaičiuota funkcinio taško vertė tipui.

Eksperimento objektas yra sistema, realizuojanti FiSMA metodiką.

Eksperimento subjektas yra projekto vadovas, šiuo atveju – magistro darbo autorė.

Nepriklausomi kintamieji yra 5 IS projektai: 3 bakalauro ir 2 magistrų darbai.

Priklausomi kintamieji yra įvairių tipų projekto elementų spėjami ir faktiniai vykdymo laikai (jie išvardyti anketoje, kuri pateikta 2 priede) bei projekto elementų skaičiavimo laikai.

Šį eksperimentą galima įvardyti kaip mini eksperimentą, kadangi tikslesnei analizei reikalingas kelių metų laikotarpis, kuomet turėtų būti registruojami vykdomi projektai, stebimi projektų spėjamų ir faktinių laikų nuokrypiai. Toks eksperimentas padėtų tiksliau įvertinti faktinę 1 funkcinio taško vertę. Dėl laiko stokos eksperimentas buvo atliekamas 2 mėnesius, todėl gauti tik preliminarūs rezultatai, tačiau iš jų galima spręsti, ar sistema bus pritaikoma.

6.3. Eksperimento vykdymas

Eksperimento metu dalyviams 2 kartus buvo pateikta identiška anketa.

Pirmo apklausos etapo metu siekiama sužinoti, kokia spėjama bendra visos sistemos realizavimo trukmė. Taip pat eksperimento dalyvių prašoma įvesti savo projektų elementų tipų skaičius, kiek numatoma prie jų užtrukti. Pateikti rezultatai yra suvedami į naujai realizuotą sistemą pagal *FiSMA* standartą, fiksuojamas projekto pavadinimas, pradžios data, įkeliami projekto funkciniai reikalavimai, pagal pateiktus anketos duomenis, kiekvienam reikalavimui parenkama klasė ir jos tipas. Apskaičiuojami reikalavimo tipo funkciniai taškai, priskiriamas vykdytojas.

Antros apklausos etapo metu eksperimento dalyvių pateiktas faktinis laikas yra suvedamas prie darbo vykdytojui priskirtų reikalavimų tipų. Susumuotas laikas išvedamas kaip faktinis projekto pabaigos laikas.

6.4. Eksperimento interpretavimas ir pateikimas

Žemiau esančioje lentelėje pateikiamos spėjamos programos realizavimo trukmės bei faktinės.

21 lentelė. Pateiktos spėjamos ir faktinės programos realizavimo trukmės

Studentas (magistrantas/bakalaurantas)	Sistemos pavadinimas (magistro/bakalauro darbo tema)	Spėjama programos realizavimo trukmė	Spėjama programos realizavimo trukmė pagal <i>FiSMA</i> tipus	Faktinė visos programos realizavimo trukmė
Arvydas Leikus (magistrantas)	Specialių statinių projektavimo įmonės projektų valdymo sistema	720 h	425 h	536 h
Džulija Leščinskaitė (magistras)	IS projekto darbų apimties įvertinimo sistema pagal <i>FiSMA</i> standartą	150 h	109 h	134 h

Eglė Mickevičiūtė (bakalauras)	Integruoto į „Facebook“ MMO žaidimo „X-Hero“ administravimo zonos dalis	300 h	229 h	281,5 h
Gintarė Bernotaitytė (bakalauras)	Kauno „Atžalyno“ vidurinės mokyklos virtualaus mokymosi ir moksleivių žinių testavimo informacinė sistema	99,66 h	100,16 h	105 h
Birutė Kaminskaitė (bakalauras)	„Darom“ vartotojų teisių valdymo sistema	150 h	66 h	140 h

21 lentelės 3 stulpelyje yra pateikiama eksperimento apklaustųjų spėjama visos programos realizavimo trukmė, pateiktos valandos - preliminarus spėjimas, remiantis prieš tai realizuotais projektais ar vertinant panašius projektus.

4 stulpelio spėjamas programos realizavimo laikas pagal *FiSMA* tipus skiriasi ženkliausiai nuo spėjamo visos programos realizavimo trukmės magistranto Arvydo projekte, greičiausiai todėl, kad projektas pradėtas dar bakalauro studijose ir buvo plėstas projektų valdymo procesas, kas įtakojo papildomus laiko ir išteklių įvertinimus. Kitaip tariant kuo projekte daugiau pokyčių, tuo labiau reikalingas tikslesnis projekto elementų įvertinimas.

Lyginant spėjamas programos realizavimo trukmes su faktinėmis, galima matyti, jog apklaustųjų projektų faktinis laikas buvo didesnis už spėjamąjį (pagal *FiSMA* tipus), nuokrypiausiai pateikti 22 lentelėje.

22 lentelė. Pateikti projektų spėjamo ir faktinio laiko nuokrypiausiai

Sistemos pavadinimas (magistro/bakalauro darbo tema)	Nuokrypiausiai
Arvydo Leikaus (Specialių statinių projektavimo įmonės projektų valdymo sistema)	111 h
Džulijos Leščinskaitės (IS projekto darbų apimties įvertinimo sistema pagal <i>FiSMA</i> standartą)	25 h
Eglės Mickevičiūtės (Integruoto į „Facebook“ MMO žaidimo „X-Hero“ administravimo zonos dalis)	52,5 h
Gintarės Bernotaitytės („Atžalyno“ vidurinės mokyklos virtualaus mokymosi ir moksleivių žinių testavimo informacinė sistema)	4,84 h
Birutės Kaminskaitės („Darom“ vartotojų teisių valdymo sistema)	74 h

Pateikti rezultatai rodo, kad egzistuoja neatitikimai, vėluojama su užsibrėžtais terminais ir reikalinga metodika, kuri galėtų padėti tiksliau įvertinti.

Norint tiksliau įvertinti teorinį laiką (pagal *FiSMA* standartą) reikalingas supratimas kokia vertė bus funkcinio taško laiko skalėje.

Visų funkcinų reikalavimų funkcinų taškų iteracijos yra sumuojamos ir gaunamas bendras projekto įvertinimas, 23 lentelėje pateikti bendri projektų funkciniai taškai.

23 lentelė. Pagal FiSMA klasių tipus apskaičiuoti bendri sistemų FFP taškai

Studentas (magistrantas/bakalaurantas)	Sistemos pavadinimas (magistro/bakalauro darbo tema)	Bendri sistemos FFP taškai
Arvydas Leikus (magistrantas)	Specialių statinių projektavimo įmonės projektų valdymo sistema	343,14 taškų
Džulija Leščinskaitė (magistras)	IS projekto darbų apimties įvertinimo sistema pagal FiSMA standartą	303,05 taškų
Eglė Mickevičiūtė (bakalauras)	Integruoto į „Facebook“ MMO žaidimo „X-Hero“ administravimo zonos dalis	220,44 taškų
Gintarė Bernotaitytė (bakalauras)	Kauno „Atžalyno“ vidurinės mokyklos virtualaus mokymosi ir moksleivių žinių testavimo informacinė sistema	115,95 taškų
Birutė Kaminskaitė (bakalauras)	„Darom“ vartotojų teisių valdymo sistema	86,56 taškų

Funkciniai taškai yra skaičiuoti pagal *FiSMA* standarto suformuotas klasių tipams tinkamas formules, kurios pateiktos darbo praktinės dalies 59 - 62 psl. Vedant projektų elementus į sistemą pagal *FiSMA* standartą yra automatiškai paskaičiuojami funkciniai taškai. Iš 23 lentelės galima matyti, jog magistrantų projektai įvertinti didesniu kiekiu funkcinų taškų, kadangi projektų sudėtingumas yra ženklesnis.

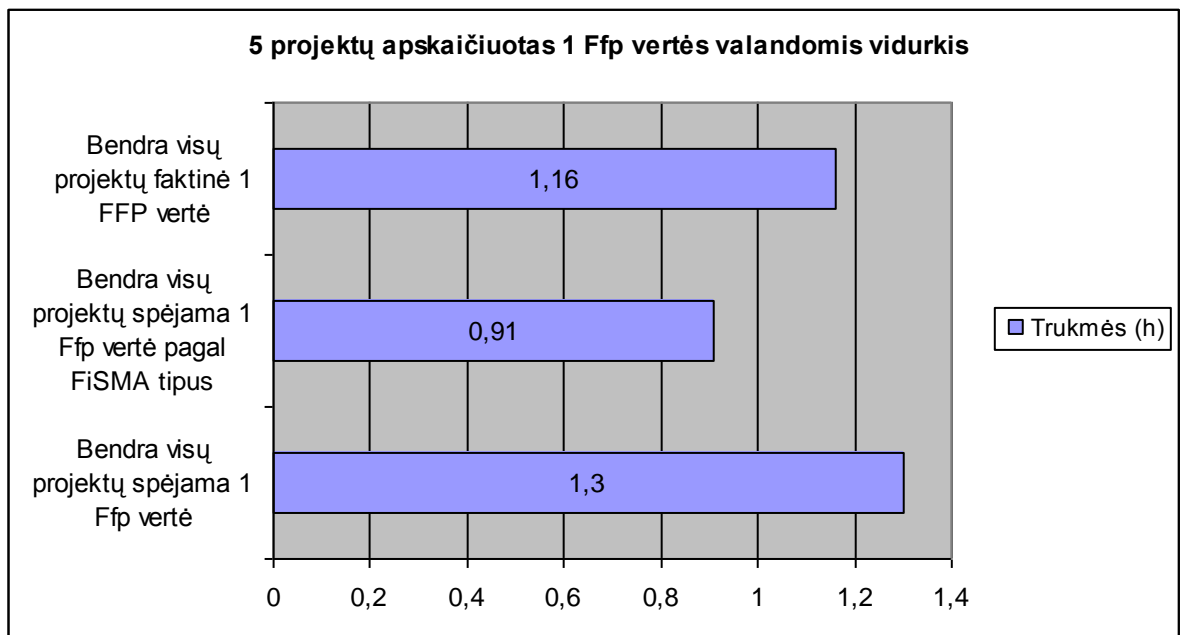
Iš turimų duomenų 21 - 23 lentelėse buvo apskaičiuotos funkcinų taškų vertės, kokios jos buvo susidariusios preliminariai įvertinus sistemą, suskaičiavus projektų elementus pagal *FiSMA* tipus ir pabaigus projektą, įvedant faktinius laikus.

24 lentelė. 1 funkcinio taško vertės

Sistemos pavadinimas (magistro/bakalauro darbo tema)	Spėjama programos 1 FFP vertė	Spėjama programos (pagal FiSMA tipus) 1 FFP vertė	Faktinė 1 FFP vertė
Specialių statinių projektavimo įmonės projektų valdymo sistema	2,09 h	1,23 h	1,56 h

IS projekto darbų apimties įvertinimo sistema pagal FiSMA standartą	0,49 h	0,35 h	0,44 h
Integruoto į „Facebook“ MMO žaidimo „X-Hero“ administravimo zonos dalis	1,36 h	1,03 h	1,27 h
Kauno „Atžalyno“ vidurinės mokyklos virtualaus mokymosi ir moksleivių žinių testavimo informacinė sistema	0,85 h	0,86 h	0,9 h
„Darom“ vartotojų teisių valdymo sistema	1,73 h	0,76 h	1,61 h

60 paveiksle pateiktos 5 projektų 1 funkcinio taško vertės vidurkiai, pagal šias reikšmes galima matyti, jog pagal numatytus projektų laikus, kurie buvo skaičiuoti pagal *FiSMA* elementus buvo nukrypta 0,25 h nuo faktinio laiko.

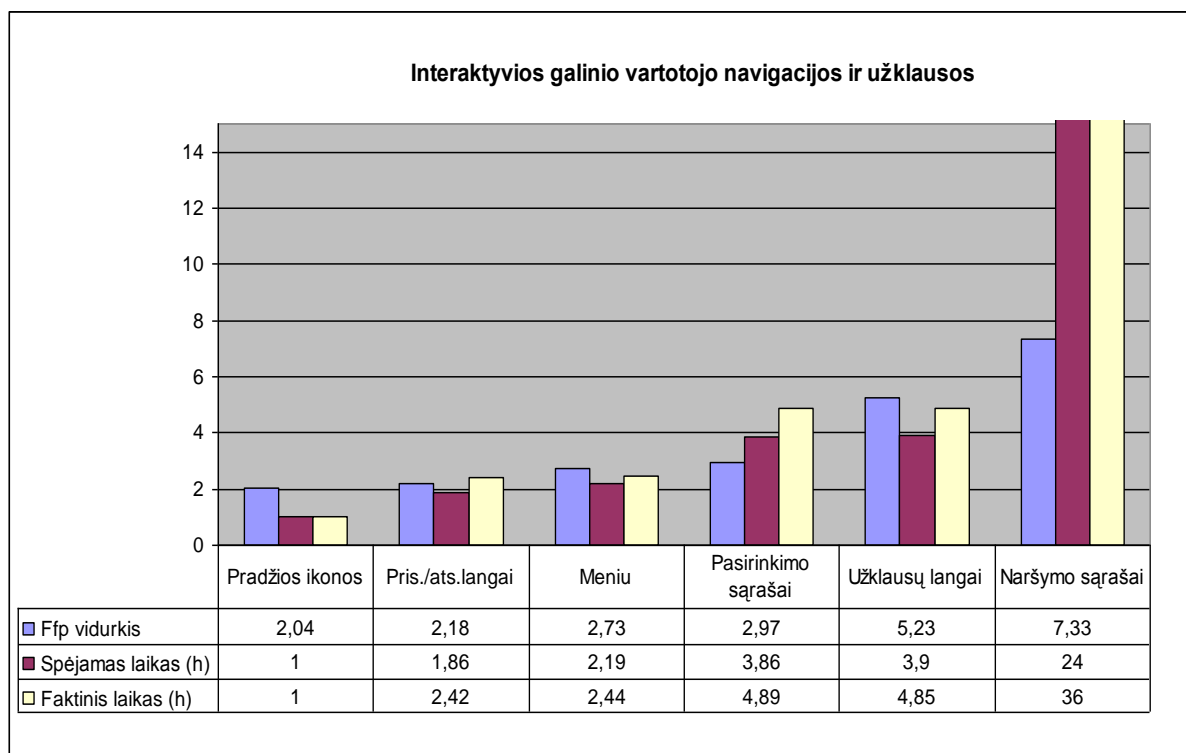


60 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

Vertės buvo paskaičiuotos turint bendrą funkcinį taškų skaičių ir priklausomai nuo trukmių gaunama atitinkama vertė. Dalinant trukmę iš suskaičiuotų funkcinį taškų skaičiaus, buvo gauta 1 funkcinio taško reikšmė. Taigi, matome, kad 5 projektų bendra 1 funkcinio taško vertė yra lygi 1,16 valandos. Kadangi eksperimentas truko 2 mėnesius ir buvo įvesti 5 projektai gautas valandų skaičius yra orientacinė funkcinio taško vertė.

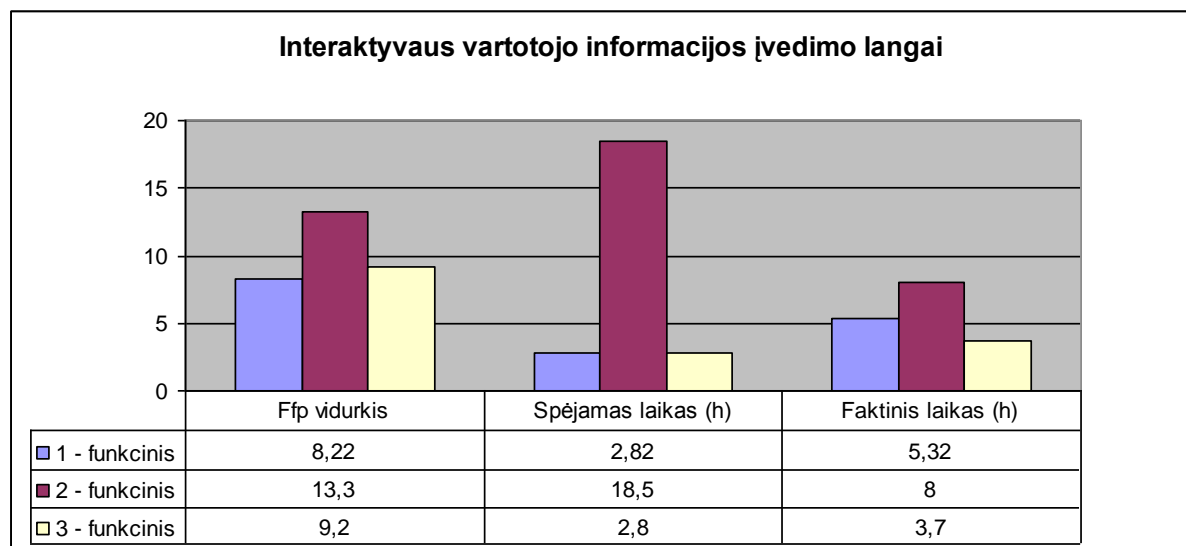
FiSMA standartas remiasi klasių ir jas apimančių tipų hierarchija. Kadangi tipai yra skirtingi, prie jų buvo užgaišta skirtingai laiko, todėl pateikiami apskaičiavimai, kokia buvo gaunama funkcinio taško vertė pagal tipus.

61 - 65 paveiksluose pateikiami šioms klasėms priklausančių tipų apskaičiuoti funkciniai taškų vidurkiai bei atitinkami spėjami ir faktiniai laikai.



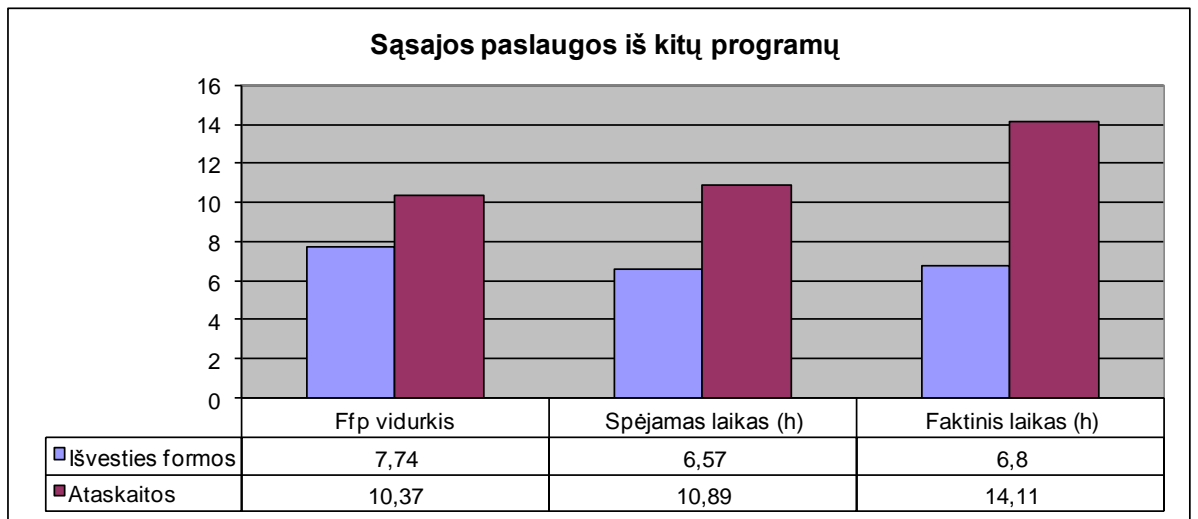
61 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

61 paveiksle matome, jog tipo pradžios ikonos spėjamas ir faktinis laikai sutampa, didžiausia paklaida tarp spėjamo ir faktinio laikų – naršymo sąrašų tipas (12 h skirtumas).



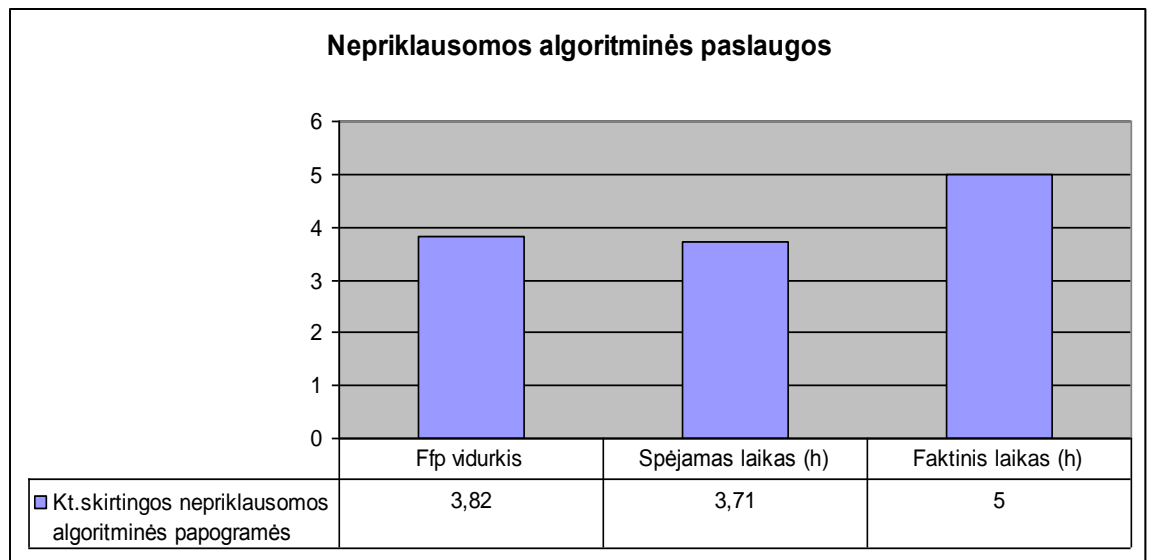
62 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

62 paveiksle 2 funkcinio tipo (pvz.: vartotojo langas su dviem funkcijom įvesti ir redaguoti) faktinio laiko paklaida buvo mažesnė lyginant su spėjamu laiku.



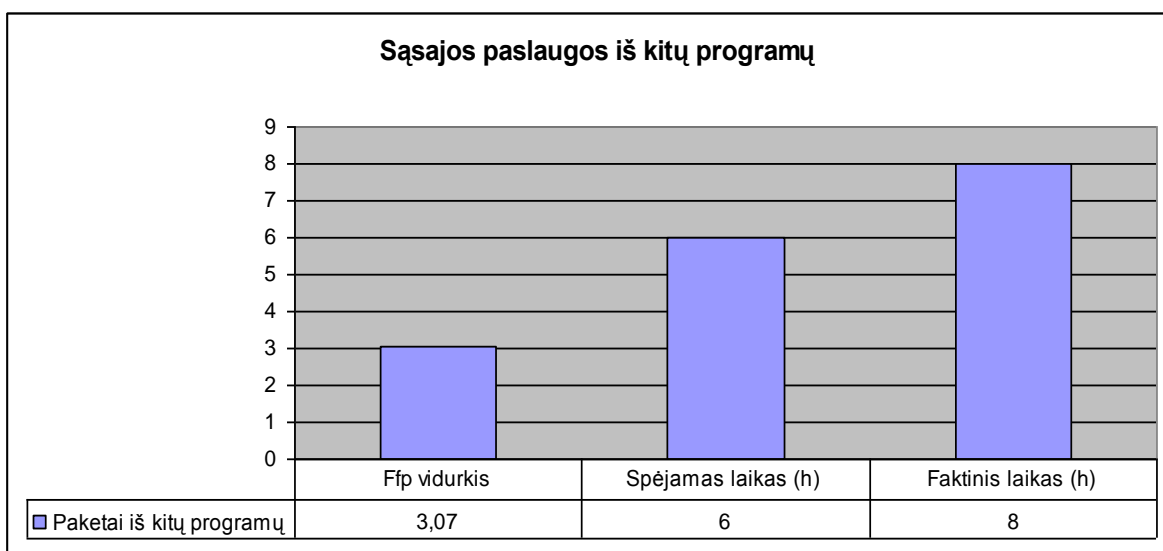
63 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

63 paveiksle išvesties formų spėjamo laiko paklaida neženkliai skyrėsi nuo faktinio laiko (0,23 min), ataskaitų laiko skirtumas – 3,22 h



64 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

64 paveiksle kitų skirtingų nepriklausomų algoritminių paprogramių spėjamo ir faktinio laikų paklaidos skirtumas – 1,29 h.



65 pav. 5 projektų apskaičiuotas 1 Ffp vertės valandomis vidurkis

65 paveiksle paketų iš kitų programų tipo spėjamo ir faktinio laikų paklaidos skirtumas – 2 h.

Iš 61 - 65 paveiksluose pateiktų spėjamų ir faktinių laikų galima matyti, jog lyginant spėjamus ir faktinius laikus pagal tipus egzistuoja neatitikimų paklaidos. Tai yra tik vienam tipui (1 – funkciniam langui) faktinis laikas buvo mažesnis už spėjamą. Tipui pavadinimu pradžios ikonos spėjamas ir faktinis laikas sutapo. Likusiems tipams (tokiems kaip naršymo sąrašai) susidarė gana didelė neatitikimo paklaida, net iki 12 h skirtumo, kitų tipų neatitikimo laikas svyravo nuo 0,2 iki 10 h. Tai rodo, naujai realizuotojo sistemoje pagal *FiSMA* standartą statistinių duomenų kaupimas apie spėjamų ir faktinių laikų, bei funkcinį taškų vertes gali būti orientacinis rodiklis įvertinant projektų laiką.

Taip pat vienas iš svarbių kriterijų naudojant vienokią ar kitokią įvertinimo metodiką yra paskaičiuoti kokią procentinę dalį užima projekto elementų apskaičiavimas, vedimas į sistemą, kad nuspręsti ar tai nėra per sudėtingas, per ilgai trunkantis procesas.

Eksperimento dalyvių buvo prašoma įvesti laiką kiek buvo užtrukta skaičiuojant projektų elementus ir kreipimasi į esybes.

25 lentelė. Trukmės vertinant projekto elementus

Studentas (magistrantas/bakalaurantas)	Sistemos pavadinimas (magistro/bakalaurato darbo tema)	Trukmė skaičiuojant projektų elementus	Vertinimo užtrukimo procentinė dalis
Arvydas Leikus (magistrantas)	Specialių statinių projektavimo įmonės projektų valdymo sistema	2,5 h	0,47
Džulija Leščinskaitė (magistras)	IS projekto darbų apimties įvertinimo	1,5 h	1,12

	sistema pagal FiSMA standartą		
Eglė Mickevičiūtė (bakalauras)	Integruoto į „Facebook“ MMO žaidimo „X-Hero“ administravimo zonos dalis	3 h	1,07
Gintarė Bernotaitytė (bakalauras)	Kauno „Atžalyno“ vidurinės mokyklos virtualaus mokymosi ir moksleivių žinių testavimo informacinė sistema	2,5 h	2,38
Birutė Kaminskaitė (bakalauras)	„Darom“ vartotojų teisių valdymo sistema	2,5 h	1,79

Vertinant projektų elementus pagal užtrukimo trukmę, visi projektuotojai nesunkiai suprato *FiSMA* metodiką, o jų sugaištas laikas neviršijo 3 procentų (svyruoja nuo 0,5 – 2,4 procentų) nuo visos projekto realizavimo trukmės. (projektų faktiniai laikai yra patiekiami 21 lentelėje).

6.5. Eksperimento išvados

1. Pateikiama eksperimento apklaustųjų spėjama visos programos realizavimo trukmė ir spėjama programos realizavimo trukmė pagal *FiSMA* tipus. Spėjamas programos realizavimo laikas pagal *FiSMA* tipus skiriasi ženkliausiai nuo spėjamos visos programos realizavimo trukmės magistranto Arvydo projekte, greičiausiai todėl, kad projektas pradėtas dar bakalauro studijose ir buvo plėstas projektų valdymo procesas, kas įtakojo papildomus laiko ir išteklių įvertinimus.
2. Eksperimento apklaustųjų projektų faktinis laikas buvo didesnis už spėjamąjį: Arvydo projektas nukrypo 111 h nuo spėjamojo laiko, Birutės projektas – 74 h, Eglės projektas – 52,5 h, Džulijos projektas – 25 h ir tik Gintarės Bernotaitytės projekto spėjamas laikas nuo faktinio laiko skyrėsi tik 4,84 h. Gauti rezultatai rodo, kad egzistuoja neatitikimai, vėluojama su užsibrėžtais terminais ir reikalinga metodika, kuri galėtų padėti tiksliau įvertinti.
3. 5 projektų bendra 1 funkcinio taško vertė yra lygi 1,16 valandos. Kadangi eksperimentas truko 2 mėnesius ir buvo įvesti tik 5 projektai gautas valandų skaičius yra orientacinė funkcinio taško vertė.
4. *FiSMA* standartas remiasi klasių ir jas apimančių tipų hierarchija. Kadangi tipai yra skirtingi, prie jų buvo užgaišta skirtingai laiko, pateikti 5 projektų tipų

apskaičiuoti funkcinių taškų vidurkiai bei atitinkami spėjami ir faktiniai laikai. Didžiausia paklaida gauta tipui - naršymo sąrašai, net iki 12 h skirtumo, likusių tipų neatitikimo laikas svyravo nuo 0,2 iki 10 h, ir tik vienam tipui (1 – funkciniam langui) faktinis laikas buvo mažesnis už spėjamą, t.p tik vienam tipui (pradžios ikonos) spėjamas ir faktinis laikas sutapo. Tai rodo, kad statistinių duomenų kaupimas apie spėjamų ir faktinių laikų, bei funkcinių taškų vertes gali būti orientacinis rodiklis įvertinant projektų laiką.

5. Vertinant projektų elementus pagal užtrukimo trukmę, visi projektuotojai nesunkiai suprato *FiSMA* metodiką, o jų sugaištas laikas neviršijo 3 procentų (svyruoja nuo 0,5 – 2,4 procentų) nuo visos projekto realizavimo trukmės.

6. Kadangi realizuotoje sistemoje numatomas tipų funkcinių verčių, faktinio, spėjamo laiko saugojimas, vertinant daugiau projektų bus gaunama tikslesnė 1 faktinio taško vertė. Todėl daroma prielaida, jog ilgesniame laikotarpyje, naujai realizuota sistema turėtų padėti tiksliau vertinti *IS* projektų darbų apimtį.

7. Išvados

1. Literatūros šaltinių ir esamų projektų darbų apimties skaičiavimo metodų analizė rodo, kad praktikoje trūksta tinkamos metodikos darbų apimčiai skaičiuoti.
2. Buvo pasirinkta ištirti naujo *FiSMA* standarto tinkamumą, kadangi ši metodika atrodė labiausiai tinkama, tačiau jai taikyti gaištamasis laikas galėjo būti pernelyg didelis.
3. Kuriant sistemą projektų apimčiai skaičiuoti, *FiSMA* metodika buvo papildyta galimybe kaupti faktinius projektų duomenis ir patikslinti funkcinių taškų įvertinimus, taip pat realizuotas funkcinių reikalavimų įkėlimas į sistemą, tuo palengvinama reikalavimų analizė, sutrumpinamas jos laikas.
4. Atliekant eksperimentą su 5 projektais gauta 1 funkcinio taško vertė yra lygi 1,16 valandos, gautas valandų skaičius yra orientacinė funkcinio taško vertė. Visi projektuotojai nesunkiai suprato *FiSMA* metodiką, o jų sugaištas laikas neviršijo 3 procentų nuo visos projekto realizavimo trukmės.
5. Siekiant įrodyti, kad sistema realizuota pagal *FiSMA* metodiką pasiteisins, reikėtų išbandyti projektų vertinimą realioje aplinkoje ir ilgesnį laiką, kadangi taip bus sukaupta daugiau statistinių duomenų, todėl bus gaunama vis tikslesnė faktinė 1 funkcinio taško vertė. Atliktas eksperimentas parodė, kad metodika yra naudinga.

8. Santrumpų ir terminų žodynas

Angliška santrumpa	Angliškas terminas	Lietuviškas terminas
FSM	<i>Function Size Measurement</i>	Funkcinės apimties matavimas
COCOMO	<i>CO</i> nstructive <i>CO</i> st <i>MO</i> del	Konstruktyvus kainos modelis
SLIM	<i>Software Life-cycle Model</i>	Programinės įrangos gyvavimo raidos ciklo valdymo modelis
FP	<i>Function Point</i>	Funkcinis taškas
FPA	F unction P oint A nalysis	Funkcinio taško analizė
IFPUG	<i>International Function Point Users Group</i>	Tarptautinė funkcinio taškų vartotojų grupė
MPC	<i>Metrics Practices Committee</i>	Metrikų praktikos komitetas
UKSMA	<i>UK Software Metrics Association</i>	Didžiosios Britanijos programinės įrangos metrikų asociacijos
COSMIC	<i>Common Software Measurement International Consortium</i>	Bendras programinės įrangos tarptautinio įvertinimo konsorciumas
FISMA	<i>Finnish Software Measurement Association</i>	Suomijos įvertinimo asociacija
LOC	<i>Length Of Code</i>	Kodo eilučių skaičius
VAF	<i>Value Adjustment Factor</i>	Vertės koregavimo faktorius
BFCs	<i>Base Functional Components</i>	Pagrindiniai funkciniai komponentai
FURs	<i>Functional User Requirements</i>	Funkciniai vartotojų reikalavimai
DETs	<i>Data Element Types</i>	Duomenų elementų tipai
LTs	<i>Logical Transactions</i>	Loginės transakcijos
RETs	<i>Record Element Types</i>	Įrašų elementų tipai

9. Literatūra

- [1] Tamošaitis R., Projektų vadybos metodiniai nurodymai, Vilnius, Technika, 2004, [žiūrėta 2008-01-20]. Prieiga internete: http://www.vgtu.lt/leidiniai/elektroniniai/projektu_vad_met_nurodymai.pdf
- [2] Lapienė A., IT projekto apimties įvertinimas, 2008-06-30, [žiūrėta 2008-01-21]. Prieiga internete: <http://www.softconsulting.lt/next.php?nr=181&news=33&page=31>
- [3] Lašinskas K., Geros ir blogos projektų valdymo praktikos, 2008-04-23, [žiūrėta 2008-01-21]. Prieiga internete: <http://www.google.lt/search?hl=lt&q=IS+projekto+darbu+apimties&meta=&aq=f&oq=IS+projekto+darbu+apimtie>
- [4] Banker R.D., Kemerer C.F., Algorithmic cost models, [žiūrėta 2008-01-23]. Prieiga internete: <http://www.ecfc.u-net.com/cost/models.htm>
- [5] Sommerville I. „Software engineering“, Eighth edition, 2007
- [6] Wallace S., Estimating. Why, What, How?, 1999-2007, [žiūrėta 2008-01-24]. Prieiga internete: <http://www.epmbook.com/estimating.htm#approaches>
- [7] Putnam, Lawrence H., Ware M., „Metrics – based scheduling, ITT corporation [žiūrėta 2008-01-24]. Prieiga internete: <https://www.goldpractices.com/practices/mbs/index.php>
- [8] Karakaž U., Sultanoğlu S., Cost Estimation Models, 1998, [žiūrėta 2008-01-23]. Prieiga internete: <http://yunus.hacettepe.edu.tr/~sencer/cocomo.html>
- [9] Symons C., COSMIC and Mark II Function Point interview, Computer Aid, Inc. (CAI), 2006, [žiūrėta 2008-01-24]. Prieiga internete: <http://www.compaid.com/caiinternet/ezine/charlessymonsinterview.pdf>
- [10] SLIM (Putnam’s Software Life-cycle Model), Quantitative Software Management, [žiūrėta 2008-01-25]. Prieiga internete: <http://softwareestimation.googlepages.com/-slim>
- [11] Gencil C., An architectural dimensions based software functional size measurement method, Middle East technical university, 2005, [žiūrėta 2008-01-24]. Prieiga internete: <http://etd.lib.metu.edu.tr/upload/12606732/index.pdf>
- [12] FiSMA 1.1. A functional size measurement method with continuous scale: basic principles and practical examples, [žiūrėta 2008-01-24]. Prieiga internete: <http://www.fisma.fi/wp-content/uploads/2008/02/fisma-11-introduction-with-examples.pdf>
- [13] Forselius P., A CAI State of the Practice Interview, 2007, [žiūrėta 2008-01-24]. Prieiga internete: <http://www.compaid.com/caiinternet/ezine/pekkaforseliusinterview.pdf>
- [14] What is DocX?2006-2010,[žiūrėta 2009-11-20]. Microsoft, Prieiga internete: <http://docx.codeplex.com/>

10. Priedai

10.1. Priedas. Naudotojo vadovas

Prisijungimas prie IS „FiSMA“

Šia sistema gali naudotis tik registruoti naudotojai, kuriems yra suteiktas prisijungimo vardas bei slaptažodis. Prisijungimo vardas bei slaptažodis yra suteikiamas sistemos administratoriaus. Prie sistemos gali prisijungti 3 vartotojų tipai: administratorius, projektų vadovas ir darbų vykdytojas.

Prie sistemos prisijungti galima:

- paleidus „FiSMA“ sistemą,
- jei sistema yra paleista, tačiau prie jos neprisijungęs joks naudotojas, pasirinkite pagrindinio meniu komandą **Programa → Prisijungti**.

Abiem atvejais atidaromas prisijungimo langas, kuriame nurodykite:

1. Lauke **Naudotojas** įvedamas prisijungimo vardas.
2. Lauke **Slaptažodis** įvedamas slaptažodis.

Pastaba: Jeigu neteisingai suvedėte slaptažodį kelis kartus iš eilės, prisijungimas yra automatiškai blokuojamas. Sistemos administratorius gali atblokuoti naudotoją meniu *Administravimas* punkte *Naudotojai*.

IS „FiSMA“ langų struktūra

IS langai yra kelių tipų:

1. Peržiūros (žr. 66 pav.) – kuriuose matomi įvestų objektų (sistemų, naudotojų, vykdytojų tipų, reikalavimų tipų įkainių) rezultatai, kiek aukščiau pateiktos informacijos matoma peržiūros lango standartinė juosta, kurioje matomos visos galimos komandos įrašams.

Pavadinimas	Pradžios data	Planuojama pabaiga	Faktinė pabaiga
Nauja sistema	10/13/2009	10/16/2009	10/22/2009
Bilietu rezervavimo sistema	12/28/2009	01/22/2010	01/19/2010
Mokytojo dienyno pildymo sistema	12/14/2009	12/15/2009	12/14/2009
Projektų valdymo sistema	09/16/2009	09/18/2009	09/18/2009
Sporto mokyklos administravimo sistema	05/12/2009	05/13/2009	05/12/2009
Teisinių bylų sistema	01/04/2010	01/05/2010	01/05/2010
Giraitės mokyklos renovacijos projektas	01/27/2010	01/27/2010	01/29/2010
Namų ūkio sistema	01/18/2010		
Vadybos apmokymo projektas			
Namų ūkio šildymo sistema	01/12/2010	01/12/2010	01/13/2010
Kelių valymo sistema	12/28/2009	12/28/2009	12/29/2009
Testavimo sistema			
Pastatų atnaujinimo darbų sistema	01/19/2010	01/20/2010	01/22/2010

66 pav. „FiSMA“ sistemų peržiūros langas

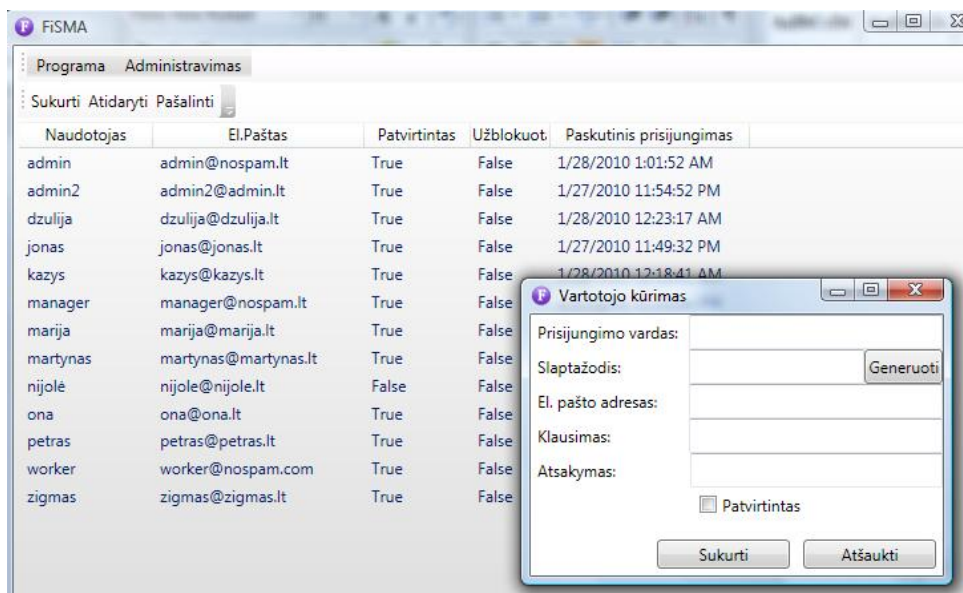
2. detalių (žr. 67 pav.) – kuriuose matoma visa informacija apie pasirinktą įrašą.

67 pav. „FiSMA“ konkrečios sistemos detalių langas

Detalių lange esanti informacija priklauso nuo pasirinkto lapo. Informacijos zonoje gali būti vienas ar keli lapai (žr. 67 pavyzdyje matomus lapus **Sistema** ir **Funkciniai reikalavimai**). Virš lapų yra detalių lango įrankių juosta, kurioje matomos visos tam įrašui galimos komandos.

Naudotojų kūrimas

Prie sistemos prisijungti ir su ja dirbti gali tik registruoti sistemos naudotojai. Įdiegus sistemą sukuriamas pirmasis naudotojas, kuris turi administratoriaus teises. Kitus naudotojus turi sukurti sistemos administratorius. Naudotojo kūrimo langas pateikiamas 68 pav.:



68 pav. „FiSMA“ vartotojo kūrimo detalių langas

Norint sukurti naują sistemos naudotoją:

1. Prisijungiamo įvedant prisijungimo vardą ir slaptažodį.
2. Atveriamas meniu **Administravimas** punktas **Naudotojai**.

Pastaba: Jei nėra matomas meniu **Administravimo** punktas **Naudotojai**, gali būti, kad nesuteikta privilegija valdyti naudotojus. Reikia kreiptis į sistemos administratorių.

3. Atsidariusiame naudotojų peržiūros lange spaudžiamas mygtukas **Sukurti**.
4. Atsidariusiame naudotojo detalių lange nurodomas naudotojo prisijungimo vardas, slaptažodis, el. pašto adresas, klausimas/atsakymas pamiršus slaptažodį, žymė skirta statusui Patvirtintas.
5. Nurodoma naudotojo rolė sistemoje: paspaudžiant mygtuką **Priskirti rolę**, atsidariusiame lange pažymima reikiama rolė ir spaudžiamas mygtukas Patvirtinti pasirinktus įrašus. Pridėjus visas reikiamas roles uždaromas rolių pasirinkimo langas.

Pastaba: Naudotojui būtina priskirti bent vieną rolę. Jei naudotojui nurodomos kelios rolės, naudotojas įgyja visas teises, kurios numatytos kiekvienai rolei atskirai.

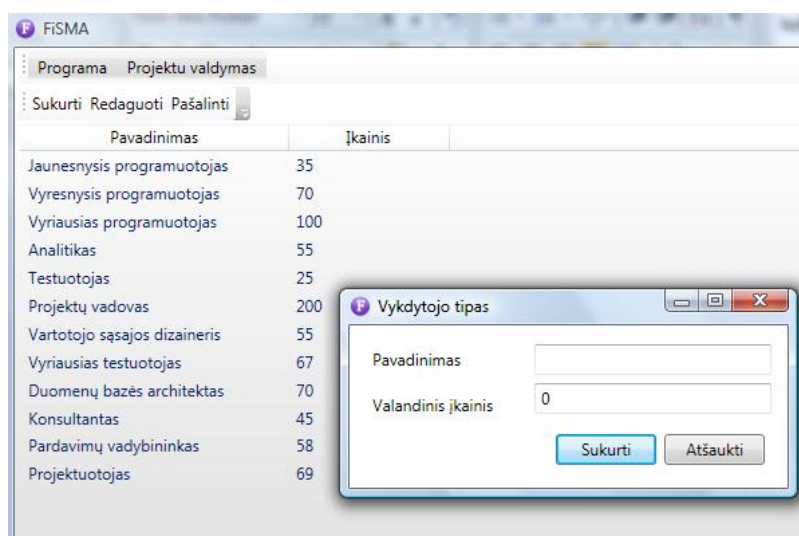
6. Statuso pasirinkimo lange galima matyti kada slaptažodis buvo keistas, kada naudotojas paskutinį kartą buvo prisijungęs, galima vartotoją užblokuoti, patvirtinti.
7. Patvirtinama įvesta informacija apie naudotoją.

Sukurtas naudotojas iš karto gali jungtis prie sistemos, su nurodytu prisijungimo vardu ir slaptažodžiu.

Vykdytojų tipų kūrimas

Vykdytojų tipai yra kuriami meniu *Projektų valdymas* punkte *Vykdytojų tipai*. Šie tipai yra reikalingi tam, kad apibrėžti valandinį įkainį priklausomai nuo to kokiam vartotojo tipui priskirtas darbų vykdytojas. Nuo šio faktoriaus priklauso funkcinio reikalavimo faktinė kaina.

Vykdytojo tipo kūrimo langas pateikiamas 69 pav.:



69 pav. „FiSMA“ vykdytojo tipo kūrimo detalių langas

Norint sukurti naują vykdytojo tipą:

1. Atsidariusiame vykdytojų tipų peržiūros lange spaudžiamas mygtukas **Sukurti**.
2. Įvedamas vykdytojo tipo pavadinimas ir valandinis įkainis, spaudžiamas mygtukas **Sukurti**.

Reikalavimų tipų įkainių redagavimas

Reikalavimų tipų įkainiai yra redaguojami meniu *Projektų valdymas* punkte *Reikalavimų tipų įkainiai*. Kadangi buvo remtasi “FiSMA” standartu sukurtos 7 klasės, kurios turi 27 tipus. Naujos klasės, tipai nėra kuriami. Iš vartotojo sąsajos galima pasikoreguoti tipo pavadinimą ir priskirti jam valandinį įkainį.

Reikalavimų tipų įkainių redagavimo langas pateikiamas 70 pav.:

Klase	Tipas	Įkainis
Interaktyvios galinio vartotojo navigacijos ir užklauso	Pradžios ikonos	15
Interaktyvios galinio vartotojo navigacijos ir užklauso	Prisijungimo ir atsijungimo langai	25
Interaktyvios galinio vartotojo navigacijos ir užklauso	Meniu	45
Interaktyvios galinio vartotojo navigacijos ir užklauso	Pasirinkimo sąrašai	35
Interaktyvios galinio vartotojo navigacijos ir užklauso	Užklauso langai	100
Interaktyvios galinio vartotojo navigacijos ir užklauso	Naršymo sąrašai	
Interaktyvios galinio vartotojo navigacijos ir užklauso	Ekranai ataskaitoms	
Interaktyvios galinio vartotojo įvestys	1-funkcinė	
Interaktyvios galinio vartotojo įvestys	2-funkcinė	
Interaktyvios galinio vartotojo įvestys	3-funkcinė	
Neinteraktyvios galinio vartotojo išvestys	Įšvesties formos	
Neinteraktyvios galinio vartotojo išvestys	Ataskaitos	
Neinteraktyvios galinio vartotojo išvestys	Elektroniniai laišukai ar tekstines žinutes	52
Neinteraktyvios galinio vartotojo išvestys	Monitoriaus ekrano išvestys	150
Sąsajos paslaugos kitoms programoms	Žinutės kt. programoms	156
Sąsajos paslaugos kitoms programoms	Signalų įrenginiai kt. programoms	160
Sąsajos paslaugos kitoms programoms	Paketų įrašai kt. programoms	145
Sąsajos paslaugos iš kitų programų	Žinutės iš kt. programų	165
Sąsajos paslaugos iš kitų programų	Signalai iš kt. įrenginių	123
Sąsajos paslaugos iš kitų programų	Paketai iš kt. programų	100
Duomenų saugyklų paslaugos	Objektai ar veiklos klasės	89
Duomenų saugyklų paslaugos	Kiti loginiai įrenginiai	95
Nepriklausomos algoritminės paslaugos	Nepriklausomos saugumo paprogramės	220
Nepriklausomos algoritminės paslaugos	Nepriklausomos simuliacijos paprogramės	300
Nepriklausomos algoritminės paslaugos	Nepriklausomos formataavimo paprogramės	400
Nepriklausomos algoritminės paslaugos	Nepriklausomos duomenų bazės valymo paprogramės	220
Nepriklausomos algoritminės paslaugos	Kitos skirtingos nepriklausomos algoritminės paprogramės	310

Reikalavimo tipo įkainis

Pavadinimas:

Valandinis įkainis:

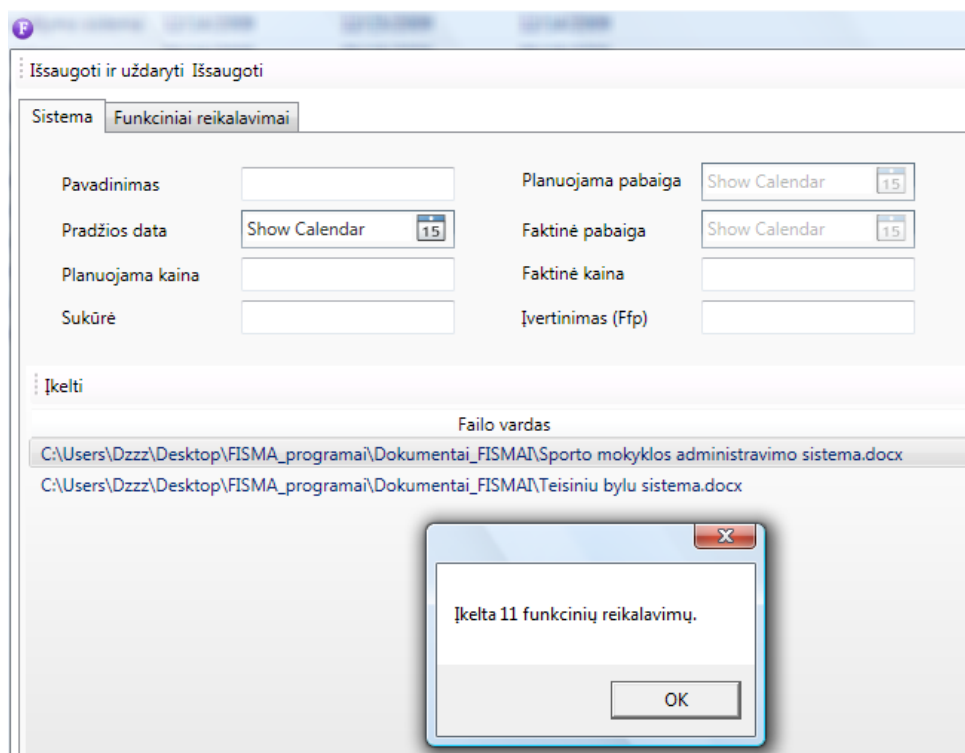
70 pav. „FiSMA“ Reikalavimų tipų įkainių redagavimo detalių langas

Norint poredaguoti reikalavimų tipų įkainį:

1. Atsidariusiame reikalavimų tipų peržiūros lange spaudžiamas mygtukas **Redaguoti**.
2. Pakoreguojamas reikalavimo tipo įkainio pavadinimas ir valandinis įkainis, spaudžiamas mygtukas **Uždaryti**.

Sistemų kūrimas

Sistemos yra kuriamos meniu **Projektų valdymas** punkte **Sistemos**. Kuriamos naujos sistemos, kurioms tikslinga apskaičiuoti darbų apimtį ir kainą.



71 pav. „FiSMA“ Naujos sistemos kūrimo detalių langas

Norint sukurti naują sistemą:

3. Atsidariusiame sistemų peržiūros lange spaudžiamas mygtukas **Sukurti**.

4. Atveriamas sistemos detalių langas. Į sistemos lapą įvedamas sistemos pavadinimas, pradžios data.

Pastaba: Laukeliai Planuojama kaina, Sukūrė, Planuojama pabaiga, Faktinė pabaiga, Faktinė kaina, Įvertinimas (Ffp) yra užpildomi sistemos, vartotojui atliekant funkcinio reikalavimo įvertinimo procesą.

5. Reikalavimai yra įkeliami spaudžiant mygtuką **Įkelti**. Išvedamas pranešimas kiek reikalavimų buvo įkelta (žr. 71 pav.).

6. Su funkciniais reikalavimais yra dirbama Funkcinių reikalavimų lape. Funkcinių reikalavimų peržiūros lange matomos apskaičiuotos reikšmės, priklausomai nuo to kokios parametru reikšmės buvo įvestos (žr. 72 pav.).

Reikalavimas	Įvertinimas	Laikas	Kaina	Faktinis Laikas	Faktinė Kaina
Atliktos rezervacijos peržiūrėjimas.	47.9	15	2475	50	7500
Įmonių sąskaitos-faktūros peržiūrėjimas.	1.4	0	0	20	3000
Parduotų bilietų kasos ataskaitos peržiūrėjimas.	8.971428571	4	60		
Ataskaitos, sąskaitos-faktūros spausdinimas.	17.5	1	90		
Vartotojo prisiregistravimas prie sistemos.	1.4	2	440		
Vartotojo galimybė pirkti bilietą internete.	31	5	450		
Vartotojo profilio informacijos keitimo galimybė.					

Pastaba: Raudonai pažymėta faktinė kaina reiškia, jog darbų vykdytojas suvedė faktiškai užgaištas valandas prie reikalavimo.

7. Konkretus reikalavimas yra atidaromas mygtuku **Atidaryti**. Lape Funkcinis reikalavimas (žr. 73 pav.) parenkama reikalavimui „Vartotojo profilio informacijos keitimo galimybė“ tinkama klasė „Interaktyvios galinio vartotojo navigacijos ir užklauso“, ją parinkus išvedami klasės parametrai, tokie kaip „Pradžios ikonos“, „Prisijungimo ir atsijungimo langai“, „Meniu“, „Parinkimo sąrašai“ ir t.t. Suvedus reikalavimui reikiamas parametrų reikšmes apskaičiuojami laikas, įvertinimas ir kaina.

Vartotojo profilio informacijos keitimo galimybė.

Uždaryti

Funkcinis reikalavimas: Vykdytojai

Reikalavimas: Vartotojo profilio informacijos keitimo galimybė.

Klasė: Interaktyvios galinio vartotojo na | Įvertinimas (Ffp): 11.7571428571429

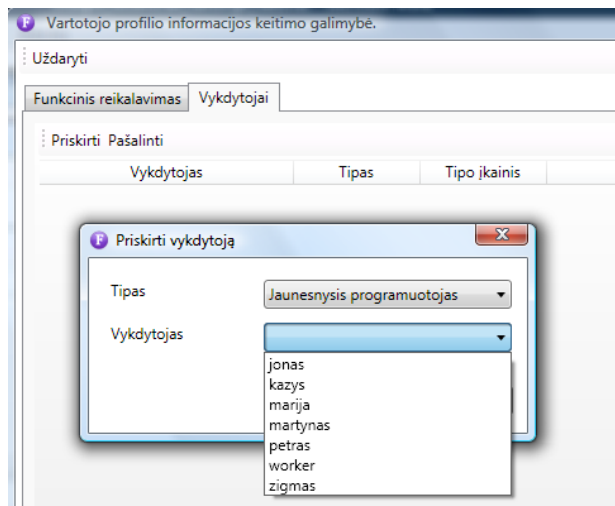
Laikas (valandomis): 9 | Kaina: 305

Parametras	Reikšmė
Pradžios ikonos	
Duomenų elementų skaičius, laukai	2
Esybių skaitymų nuorodų skaičius	5
Užtruks (valandas)	1
Prisijungimo ir atsijungimo langai	
Duomenų elementų skaičius, laukai	6
Esybių skaitymų nuorodų skaičius	4
Užtruks (valandas)	2
Meniu	
Duomenų elementų skaičius, laukai	9
Esybių skaitymų nuorodų skaičius	2
Užtruks (valandas)	3
Pasirinkimo sąrašai	
Duomenų elementų skaičius, laukai	10
Esybių skaitymų nuorodų skaičius	2
Užtruks (valandas)	3
Užklauso langai	
Duomenų elementų skaičius, laukai	0
Esybių skaitymų nuorodų skaičius	0

73 pav. „FiSMA“ Konkretaus funkcinio reikalavimo detalių langas

Pastaba: Funkciniai reikalavimai gali nereikėti tam tikrų klasės tipų, jų vietoje paliekami 0.

8. Pasirinkus lapą Vykdytojas (žr. 74 pav.) mygtuku **Priskirti** yra atidaromas vykdytojo priskyrimo langas. Priskyrus tipą ir vykdytoją mygtuku **Uždaryti** priskiriamas reikalavimui darbų vykdytojas.

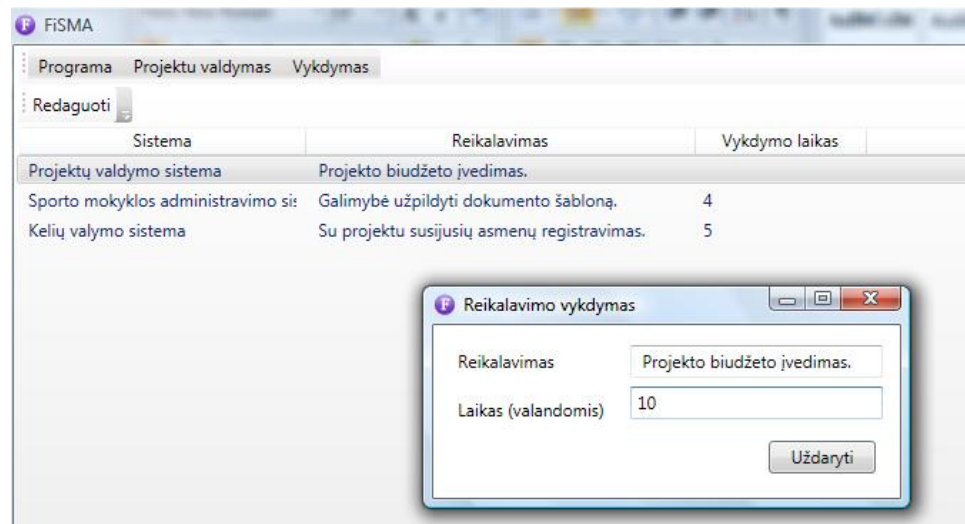


74 pav. „FiSMA“ vykdytojo priskyrimo detalių langas

Pastaba: Funkcinis reikalavimas gali būti priskirtas keliems darbų vykdytojams.

Faktinio laiko įvedimas

Funkciniam reikalavimui yra priskiriamas darbų vykdytojas. Prisijungęs vartotojas prie savo prieigos gali atidaryti meniu **Vykdymas** punktą **Priskirti reikalavimai** ir peržiūrėti reikalavimus, kurie jam skirti.



75 pav. „FiSMA“ Naujos sistemos kūrimo detalių langas

Norint poredaguoti, susivesti naujai faktiškai užgaištą laiką (valandomis):

1. Atsidariusiame priskirtų reikalavimų peržiūros lange spaudžiamas mygtukas **Redaguoti**.

2. Pakoreguojamas arba įvedamas naujai konkretaus priskirto reikalavimo valandinis įkainis, spaudžiamas mygtukas **Uždaryti**.

Pastaba: Darbų vykdytojo suvestas faktinis laikas yra fiksuojamas prie sistemos, kuriai jis priskirtas, pagal tai kokiam vykdytojo tipui priskirtas vartotojas taip pat apskaičiuojama faktinė kaina.

10.2. Priedas. Testavimo medžiaga

Pateikiama bendra testavimo apžvalga, testavimo tikslai, testavimo strategija su testavimo atvejais.

Testavimo tikslai ir objektai

Kuriant informacinę sistemą siekiama sukurti produktą, turintį kuo mažiau klaidų ir defektų. Testavimo tikslas atskleisti kaip galima daugiau programinės įrangos klaidų, su tikslu jas ištaisyti, beabejonės išlieka visada kiekis defektų, naujų sistemos savybių, kurios nebus pastebėtos, kadangi nėra žinoma kaip vartotojui bus patogiau prieiti prie vienokių ar kitokių sistemos funkcijų.

Taip pat svarbu patikrinti ar kurtojo informacinė sistema yra tinkama skirtam tikslui, t. y. ar ji atitinka vartotojo reikalavimus ir specifikaciją, ar vartotojui bus patogiu, nesudėtinga ja naudotis. Testavimo metu stengiamasi atrasti vietų, kurias galima tobulinti, kad standarto „FiSMA“ skaičiavimo procesą pateikti suprantamesnį ir intuityvesnį.

Skyriuje pateikiami testavimo atvejai leidžiantys užtikrinti kiek įmanoma mažesnę informacinės sistemos projekto darbų apimties nustatymo metodikos klaidų kiekį.

Testavimo apimtis

Informacinės sistemos projekto darbų apimties nustatymo metodika pagal standartą „FiSMA“ buvo ištestuota pagal reikalavimų specifikaciją ir patikrinta ar sistema atitinka vartotojo poreikius.

Pagrindiniai apribojimai

Pagrindiniai apribojimai, kurie galėjo įtakoti testavimo plano nevykdymą:

- Laiko stoka.
- Patirties trūkumas dalykinėje srityje.
- Kadangi lygiagrečiai atliekama keletas darbų, tai įtakoja testavimo neišbaigtumą.

Testuojama programinė įranga

Testuojama informacinė sistema, kuri leidžia įvertinti darbų apimtį, prieš pateikiant funkcinį reikalavimą, įvertinant planuojamą kainą ir laiką ir apskaičiuojant faktinę kainą ir laiką atlikus projekto darbus.

Testuojant sąsajas buvo ištestuoti šie langai (sąsajos):

Prisijungimo langas. Lange prašoma įvesti sistemos vartotojo prisijungimo duomenis: vartotojo vardą ir slaptažodį. Paspaudus prisijungimo mygtuką, tikrinami įvesti duomenys. Įvedus neteisingus duomenis apie tai sistemos vartotojas informuojamas atskiru pranešimu. Paspaudus uždarymo mygtuką langas užsidaro ir programa baigia darbą. Įvedus teisingus duomenis prisijungimo langas uždaromas, aktyvuojamas pagrindinis langas, priklausomai nuo to kokiai vartotojų grupei besijungiantysis buvo priskirtas.

Pagrindinis langas. Lange rodomas pagrindinis meniu. Pasirinkus meniu punktą pagal priskirtą funkciją atidaromas atitinkamas kitas langas. Galimi langai aprašyti žemiau.

Vykdytojų tipų langas. Lange yra vaizduojami vykdytojų tipų pavadinimai ir jų valandiniai įkainiai. Pasirinkus konkretų vykdytojo tipą ir paspaudus redagavimo mygtuką yra išskviečiamas atskiras vykdytojo tipo langas su užpildytais laukais. Toks pats langas išskviečiamas ir pasirinkus naujo vykdytojo tipo sukūrimo mygtuką, tik su tuščiais laukais. Pasirinkus konkretų vykdytojo tipą ir paspaudus mygtuką šalinti, įrašas pašalinamas

Reikalavimų tipų įkainių langas. Pateikiamos reikalavimų klasės ir šioms klasėms priskirti tipai. Kadangi pagal „FiSMA“ standartą klasės yra apibrėžtos, vartotojo sąsajoje jos yra neredaguojamos. Galima tik pakoreguoti klasėms priskirtų tipų pavadinimus. Paspaudus redaguoti mygtuką yra išskviečiamas konkretaus tipo langas, kuriame vartotojas gali įvesti naujai, redaguoti tam tikros klasės, jai priskirto tipo pavadinimą ir šiam įkainiui atlikti nustatytą valandinį įkainį.

Sistemų langas. Šiame lange yra pateikiamos įvestos planuojamos kurti/kuriamos sistemos, galima matyti, sistemos pavadinimą, pradžios datą, planuojamą pabaigą, faktinę pabaigą. Pradžios data yra pasirenkama tik įvedus sistemą. Planuojama pabaigos data yra apskaičiuojama, atsižvelgiant į funkciniam reikalavimams priskirtų klasių susumuotus valandinius atlikimo laikus. Faktinė sistemos pabaigos data yra įvedama, kuomet vartotojai atlieka jiems priskirtus reikalavimus, susivedę prie reikalavimo užgaištą laiką. Mygtuku sukurti išskviečiamas naujos sistemos sukūrimo langas, kuriame langas skiriamas į du lapus: sistema ir funkciniai reikalavimai.

Sistemos lape įvedamas sistemos pavadinimas, pasirenkama sistemos realizacijos pradžios data, mygtuku įkelti įkeliami į sistemą pasirenkami iš vartotojo kompiuteryje esančio katalogo MS Word dokumentai, kurių funkciniai reikalavimai pateikti pagal Volere šablona.

Funkcinių reikalavimų lape galima matyti įkeltus reikalavimus. Šiame lape matoma kiek Ffp taškų įvertinti reikalavimai, kiek laiko numatoma užtrukti ir kiek preliminariai kainuos reikalavimo atlikimas, taip pat faktinė kaina ir laikas.

Funkcinio reikalavimo langas. Paspaudus atidaryti mygtuką yra iškviečiamas konkretaus funkcinio reikalavimo langas, kuriame vartotojas gali pasikoreguoti įkelto reikalavimo pavadinimą, priskirti šiam reikalavimui tinkamą klasę, pagal klasės tipus iškviečiami šiai klasei priklausantys tipai ir jų parametrai. Vartotojui įvedus tipo parametru reikšmes įvertinamas Ffp, įvedus šiam tipui laiką, kiek bus užgaišta, sumuojamos valandos, priklausomai nuo to, kiek reikalavimas įvertintas apskaičiuojamas funkcinio reikalavimo preliminari kaina. Vykdytojų priskyrimas vykdomas vykdytojų lape, mygtuku priskirti yra iškviečiamas vykdytojų priskyrimo langas.

Vykdytojų priskyrimo langas. Šis langas yra iškviečiamas paspaudus priskirti mygtuką paskaičiavus konkretaus funkcinio reikalavimo atlikimo laiką ir kainą. Iš tipo laukelio yra pasirenkamas vykdytojo tipas, iš vykdytojo laukelio sistemos registruotas darbų vykdytojas. Pasirinkus reikšmes vykdytojas yra priskiriamas funkciniam reikalavimui. Jei vykdytojas buvo pasirinktas klaidingai, ar nutarta jį pakeisti kitu vykdytoju, šalinama mygtuku prie reikalavimo priskirtą vykdytoją.

Priskirtų reikalavimų langas. Paspaudus meniu punktą vykdymas yra iškviečiamas vykdytojui priskirtų reikalavimų langas, kuriame patiekama informacija apie sistemą, šiai sistemai reikalingą atlikti funkcinį reikalavimą, paspaudus redaguoti veiklą iškviečiamas konkretaus reikalavimo vykdymo langas, kuriame vykdytojas nurodo valandas, kiek konkrečiai užtruko atlikdamas reikalavimą.

Sistemos naudotojų langas. Lange vaizduojami užregistruoti naudotojai. Pasirinkus naudotoją ir paspaudus redagavimo mygtuką yra iškviečiamas atskiras naudotojo langas su užpildytais laukais, kur galima keist naudotojo duomenis. Toks pats langas iškviečiamas ir pasirinkus naujo naudotojo sukūrimo mygtuką, tik su tuščiais laukais. Pasirinkus šalinimo mygtuką šalinamas pasirinktas naudotojas.

Sistemos naudotojo langas. Langas skirtas naudotojo duomenims įvesti, redaguoti arba patvirtinti naudotojo šalinimą. Jame yra vartotojo duomenims reikalingi laukai, bei patvirtinimo ir atšaukimo mygtukai. Lango ir jo mygtukų funkcijos priklauso nuo to koks metodas jį iškvietė.

Testavimo strategija

Vartotojo sąsajos testavimas atliekamas pagal „juodos dėžės“ principą užbaigtam produktui.

Sistema buvo nuodugniai ištestuota pagal reikalavimų specifikaciją ir patikrinta ar atitinka vartotojo poreikius. Radus neatitikimą tarp sistemos ir vartotojo poreikių, buvo patikrinta reikalavimų specifikacijoje.

Testavimo resursai

Jokių papildomų techninių ar programinių resursų, kurie viršytų jau turimus resursus skirtus sistemos kūrimui, nereikia. Testavimo darbams buvo naudojamas asmeninis programinės įrangos kūrėjos kompiuteris.

Testavimo atvejai

Skyriuje pateikiami testavimo atvejai, kurie buvo naudojami atliekant programinės įrangos testavimą.

Vartotojo sąsajos (UI) detalesnis testavimas

Žemiau lentelėse pateikiami testavimo atvejai.

26 lentelė. Prisijungimo lango testavimo atvejai

Veiksmas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Įvedami teisingi prisijungimo duomenys	Vartotojas = dzulija Slaptažodis= zzzz1234!	Parodomas pagrindinis programos langas, pagal tai kokios rolės priskirtos naudotojui, tokie atitinkami meniu punktai pateikiami.	Prie sistemos prisijungiama
Įvedamas neteisingai prisijungimo vardas	Vartotojas = jolanta Slaptažodis= zzzz1234!	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Įvedamas neteisingai slaptažodis	Vartotojas = jolanta Slaptažodis= zzzz	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Įvedami neteisingai abu laukeliai	Vartotojas = mantas Slaptažodis= kkkk	Vartotojas informuojamas, kad įvesti neteisingi duomenys.	Prie sistemos neprisijungiama.
Į sistemą prisijungimo duomenys nevedami, spaudžiamas uždaryti mygtukas.	-	Programa uždaro.	Baigiamas darbas.

27 lentelė. Pagrindinio lango testavimo atvejai

Veiksmas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Pasirenkamas meniu punktas „Vykdymo tipai“	Paspaustas „Vykdymo tipų“ meniu	Atidaromas vartotojų tipų langas. Jame parodomas įvestų vykdytojų tipų	Atidaromas vykdytojų tipų langas

		sąrašas.	
Pasirenkamas meniu punktas „Reikalavimų tipų įkainiai“	Paspaustas „Reikalavimų tipų įkainiai“ meniu	Atidaromas reikalavimų tipų langas. Jame parodomas reikalavimų tipų įkainių sąrašas.	Atidaromas reikalavimų tipų įkainių langas
Pasirenkamas meniu punktas „Sistemos“	Paspaustas „Sistemos“ meniu	Atidaromas sistemų langas. Jame parodomas sistemų sąrašas.	Atidaromas sistemų langas
Pasirenkamas meniu punktas „Naudotojai“	Paspaustas „Naudotojai“ meniu	Atidaromas naudotojų langas. Jame parodomas įvestų naudotojų sąrašas.	Atidaromas naudotojų langas
Pasirenkamas meniu punktas „Priskirti reikalavimai“	Paspaustas „Priskirti reikalavimai“ meniu	Atidaromas priskirtų reikalavimų langas.	Atidaromas priskirtų reikalavimų langas
Pasirenkamas meniu punktas „Išeiti“ arba langas uždaromas	Paspaustas „Išeiti“ meniu	Programa uždaro.	Programa uždaro.
Pasirenkamas meniu punktas „Atsijungti“	Paspaustas „Atsijungti“ meniu	Atsijungiama nuo sistemos, galima alternatyva prisijungimui prie kt. vartotojo prieigos.	Atsijungiama nuo sistemos.

28 lentelė. Vykdytojų tipų lango testavimo atvejai

Veiksmas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Sukurti“.	Paspaustas mygtukas „Sukurtas“	Parodomas vykdytojo tipo sukūrimo langas su tuščiais laukais.	Atidaromas vykdytojo tipo langas.
Paspaudžiamas mygtukas „Redaguoti“.	Paspaustas mygtukas „Redaguoti“	Parodomas vykdytojo tipo redagavimo langas su vykdytojo tipo duomenimis užpildytais laukais.	Atidaromas vykdytojo langas su užpildytais laukais.
Paspaudžiamas mygtukas „Pašalinti“.	Pažymimas ir šalinamas vykdytojo tipas.	Pašalinamas įrašas su užpildytais laukais.	Vykdytojo tipas pašalinamas.
Neįvedami duomenys į laukelius, spaudžiama „Uždaryti“.	Paspaustas mygtukas „Uždaryti“	Duomenys neįvedami.	Duomenys neįvedami.
Suvedami teisingi duomenys į laukelius, spaudžiama „Uždaryti“.	Pavadinimas= Analitikas Įkainis = 70	Langas uždaromas. Vartotojo tipo duomenys įvedami į duomenų bazę.	Duomenys išsaugomi duomenų bazėje.

29 lentelė. Reikalavimų tipų įkainių lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Redaguoti“.	Paspaustas mygtukas „Redaguoti“	Parodomas reikalavimo tipo įkainio redagavimo langas su užpildytais	Atidaromas reikalavimo tipo langas su užpildytais

		laukais.	laukais.
Redaguojami reikalavimo tipo įkainio lango laukeliai ir spaudžiama „Uždaryti“.	Pavadinimas =Menu, Valandinis įkainis = 15	Reikalavimo tipo įkainio langas uždaromas. Paredaguoti duomenys atnaujinami duomenų bazėje.	Paredaguoti duomenys užsaugomi duomenų bazėje.

30 lentelė. Sistemų lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Sukurti“.	Paspaustas mygtukas „Sukurtas“	Parodomas naujos sistemos sukūrimo langas su tuščiais laukais.	Atidaromas naujas sistemos langas
Paspaudžiamas mygtukas „Atidaryti“.	Paspaustas mygtukas „Atidaryti“	Parodomas sistemos redagavimo langas su sistemos duomenimis užpildytais laukais.	Atidaromas sistemos langas su užpildytais laukais.
Paspaudžiamas mygtukas „Šalinti“.	Pažymima ir šalinama sistema.	Pašalinama sistema su užpildytais laukais.	Pašalinta sistema.
Neįvedami jokie sistemos laukeliai, spaudžiama „Išsaugoti“.	Paspaustas mygtukas „Uždaryti“	Prašoma įvesti sistemos pavadinimą, sistema neišsaugoma.	Sistema neišsaugoma
Užpildomi visi sistemos laukeliai, spaudžiama „Išsaugoti“.	Pavadinimas = Bilietų rezervavimo sistema, Pradžios data = 2010-01-15	Sistemos duomenys išsaugomi duomenų bazėje.	Duomenys užsaugomi.
Paspaudžiamas mygtukas „Įkelti“	Paspaustas mygtukas „Įkelti“	Filtruojami MS Word 2007 dokumentai.	Filtruojami MS Word 2007 dokumentai.
Pasirenkamas tinkamas dokumentas su funkciniais reikalavimais.	Pasirinktas tinkamas dokumentas	Pasirinkto dokumento funkciniai reikalavimai įkeliami į sistemą.	Įkelta į sistemą 10 reikalavimų.
Pasirenkamas dokumentas be funkinių reikalavimų.	Pasirinktas dokumentas	Prie sistemos nepriskiriamas nei vienas funkcinis reikalavimas.	Įkelta į sistemą 0 reikalavimų.

31 lentelė. Funkcinio reikalavimo lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Atidaryti“.	Paspaustas mygtukas „Atidaryti“	Parodomas funkcinio reikalavimo langas.	Atidaromas funkcinio reikalavimo langas.
Pasirenkama klasė.	Klasė = Interaktyvios galinio vartotojo įvestys	Išvedami klasei priklausantys tipų parametrai.	Išvedami pasirinktos klasės tipų parametrai.
Nėra pasirenkama klasė	-	Neišvedama klasė, su jos tipų parametrais.	Neišvedami klasės tipų parametrai.

Suvedamos parametrų reikšmės.	Funktionalumo daugiklis = 5, Duomenų elementų laukų skaičius = 15, Esybių tik skaitymo nuorodų skaičius = 5, Esybių rašymo nuorodų skaičius = 2, Užtruks valandas = 6	Apskaičiuojamas įvertinimas (Ffp), laikas, kaina, duomenys išsaugomi duomenų bazėje.	Apskaičiuojamas Ffp, laikas, kaina, duomenys išsaugomi.
Nesuvedamos parametrų reikšmės.	-	Neapskaičiuojamas įvertinimas (Ffp), laikas, kaina.	Neapskaičiuojamas Ffp, laikas, kaina
Paspaudžiamas mygtukas „Priskirti“.	Paspaustas mygtukas „Priskirti“	Parodomas vykdytojo tipo priskyrimo langas.	Atidaromas vykdytojo tipo priskyrimo langas.
Pasirenkami vykdytojo duomenys, spaudžiamas „Priskirti“ mygtukas.	Tipas = jaunesnysis programuotojas, Vykdytojas = Kazys	Vykdytojas priskiriamas prie sistemos. Duomenys išsaugomi duomenų bazėje.	Vykdytojas priskirtas prie sistemos.
Paspaudžiamas mygtukas „Pašalinti“.	Pažymimas ir šalinamas vykdytojas.	Pašalinamas vykdytojas.	Vykdytojas pašalintas.

32 lentelė. Vykdytojų priskyrimo lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Redaguoti“.	Paspaustas mygtukas „Redaguoti“	Parodomas funkcinio reikalavimo vykdymo langas.	Atidaromas funkcinio reikalavimo vykdymo langas.
Įvedamas laikas, spaudžiama „Uždaryti“.	Laikas(valandomis) = 5	Langas uždaromas. Duomenys išsaugomi duomenų bazėje.	Duomenys išsaugomi.
Atliekama reikalavimo peržiūra, laikas neįvedamas.	Peržiūrėtas reikalavimas.	Langas uždaromas. Duomenys neišsaugomi duomenų bazėje.	Duomenys neišsaugomi.

33 lentelė. Naudotojų lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Paspaudžiamas mygtukas „Sukurti“.	Paspaustas mygtukas „Sukurti“	Parodomas naujo naudotojo sukūrimo langas su tuščiais laukais.	Atidaromas naudotojo langas.
Paspaudžiamas mygtukas „Atidaryti“.	Paspaustas mygtukas „Atidaryti“	Parodomas naudotojo duomenų redagavimo langas su pasirinkto naudotojo duomenimis	Atidaromas naudotojo langas su duomenimis.

		užpildytais laukais.	
Paspaudžiamas mygtukas „Pašalinti“.	Pažymimas ir šalinamas naudotojas.	Pašalinamas vartotojas.	Naudotojas pašalintas.

34 lentelė. Naudotojo lango testavimo atvejai

Testas	Įvesti duomenys	Laukiamas rezultatas	Gauti rezultatai
Neįvedami duomenys į naudotojo laukus ir spaudžiamas mygtukas „Gerai“.	-	Parodomas validacijos pranešimas.	Naudotojas neišsaugomas.
Įvedami teisingi duomenys į naudotojo laukus ir spaudžiamas mygtukas „Gerai“.	Prisijungimo vardas = jonas, slaptažodis = zzzz1234!, el.pašto adresas= jonas@jonas.lt, Klausimas = Tėvo vardas?, Atsakymas = Petras	Langas uždaromas. Naudotojo duomenys atnaujinami duomenų bazėje.	Išsaugomas naudotojas.
Paspaudžiamas mygtukas „Atšaukti“ arba uždaromas langas.	Paspaustas „Atšaukti“ mygtukas	Langas uždaromas. Duomenys duomenų bazėje neatnaujinami.	Naudotojas neišsaugomas.

10.3. Priedas. Apklausos anketa

Prašome padėti atlikti tyrimą tema „Informacinių sistemų projekto darbų apimties nustatymo metodika“, Šiuo eksperimentu siekiama išsiaiškinti, ar „FiSMA“ metodika yra tinkama skaičiuoti funkcinių reikalavimų apimtį. Pagal gautus duomenis bus bandoma daryti išvadą, ar „FiSMA“ realiai galėtų pagerinti funkcinių reikalavimų realizavimo laiko skaičiavimą.

Prašome užpildyti lenteles savo kuriamos programos duomenimis. Ten, kur daugtaškiai, įterpkite tiek naujų eilučių, kiek reikia pagal Jūsų kuriamų elementų tipų skaičių. **Trukmę rašykite valandomis**, jei reikia tiksliau, pavyzdžiui, valandos ir 20 minučių, rašykite 1,33.

Pildymo pavyzdys. Tarkime, turite 4 užklausų langus, iš kurių du turi tą patį duomenų laukų ir skaitomų esybių skaičių. Tuomet sukuriate dar dvi eilutes ir įrašote atitinkamus duomenis:

UŽKLAUSŲ LANGAI (Data inquires)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
1 užklausos lango duomenų laukų skaičius (n)	2	1	2	
1 užklausos lango skaitomų esybių skaičius (n)	5			
2 užklausos lango duomenų laukų skaičius (n)	3	1	2,5	
2 užklausos lango skaitomų esybių skaičius (n)	4			
3 užklausos lango duomenų laukų skaičius (n)	5	2	3,6	
3 užklausos lango skaitomų esybių skaičius (n)	6			

Tai reiškia, kad turite vieną užklausą su 2 duomenų laukais ir 5 skaitomomis esybėmis; vieną užklausą su 3 duomenų laukais ir 4 skaitomomis esybėmis ir 2 užklausas su 5 duomenų laukais ir 6 skaitomomis esybėmis ir galvojate, kad pirmai užklausiai reikės 2 val., antrai – 2,5 val. ir kiekvienai trečio tipo užklausiai po 3,6 val. Jeigu Jūsų sistema jau realizuota, užpildykite faktines reikšmes. Jei žinote, kiek laiko įsivaizdavote, kad reikės, ir kiek reikėjo iš tiesų, užpildykite abi reikšmes. Gali dalis reikšmių būti numatytų ir faktinių, dalis – tik numatytų. Paskutinėje lentelėje įrašykite bendrą sistemos kūrimo laiką (numatytą arba faktinį)

Šios reikšmės gali būti labai apytikslės. Jei turite daug panašių langų ar užklausų, rašykite bendrą jų skaičių ir apytikslį elementų skaičių. Praleiskite elementus, kurių pas Jus nėra.

1. Interaktyvaus vartotojo navigavimo ir užklausų funkciniai komponentai (visos interaktyvios funkcijos be redagavimo galimybių):

PRADŽIOS IKONOS (dažniausiai bus viena, o interneto sistema neturi nė vienos)	n	Analogiškų ikonų skaičius	Numatoma vienos ikonos realizavimo trukmė	Faktinė vienos ikonos realizavimo trukmė
Pradžios ikonos skaitymo nuorodų skaičius (n)				
...				

PRISIJUNGIMO, ATSIJUNGIMO LANGAI (dažniausiai bus po vieną)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
---	---	---------------------------	---	--

1 prisijungimo lango duomenų laukų skaičius (n)				
1 prisijungimo lango skaitomų esybių skaičius (n)				
2 prisijungimo lango duomenų laukų skaičius (n)				
2 prisijungimo lango skaitomų esybių skaičius (n)				
...				
1 atsijungimo lango duomenų laukų skaičius (n)				
1 atsijungimo lango skaitomų esybių skaičius (n)				
2 atsijungimo lango duomenų laukų skaičius (n)				
2 atsijungimo lango skaitomų esybių skaičius (n)				
...				

MENIU	n	Analogiškų meniu skaičius	Numatoma realizavimo trukmė	Faktinė reikšmė
1 meniu skaitomų esybių skaičius (n)				
2 meniu skaitomų esybių skaičius (n)				
3 meniu skaitomų esybių skaičius (n)				

PASIRINKIMO SĄRAŠAI (drop-down lists)	n	Analogiškų sąrašų skaičius	Numatoma vieno sąrašo realizavimo trukmė	Faktinė vieno sąrašo realizavimo trukmė
1 pasirinkimo sąrašo duomenų laukų skaičius (n)				
1 pasirinkimo sąrašo skaitomų esybių skaičius (n)				
2 pasirinkimo sąrašo duomenų laukų skaičius (n)				
2 pasirinkimo sąrašo skaitomų esybių skaičius (n)				
...				

UŽKLAUSŲ LANGAI (Data inquires)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
1 užklauso lango duomenų laukų skaičius (n)				
1 užklauso lango skaitomų esybių skaičius (n)				
2 užklauso lango duomenų laukų skaičius (n)				
2 užklauso lango skaitomų esybių skaičius (n)				
...				

NARŠYMO SĄRAŠAI (browsing lists)	n	Analogiškų sąrašų skaičius	Numatoma vieno sąrašo realizavimo trukmė	Faktinė vieno sąrašo realizavimo trukmė
1 naršymo sąrašo duomenų laukų skaičius (n)				
1 naršymo sąrašo skaitomų esybių skaičius (n)				
2 naršymo sąrašo duomenų laukų skaičius (n)				
2 naršymo sąrašo skaitomų esybių skaičius (n)				
...				

2. Interaktyvaus vartotojo informacijos įvedimo langai

VARTOTOJO LANGAI SU VIENA FUNKCIJA¹ (arba įvesti, arba redaguoti, arba trinti)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
--	----------	----------------------------------	--	---

¹ Reikiamą reikšmę pabraukti, pavyzdžiui, jei lange galima tik įvesti duomenis, pabraukti įvesti

VARTOTOJO LANGAI SU VIENA FUNKCIJA¹ (arba įvesti, arba redaguoti, arba trinti)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
1 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) duomenų laukų skaičius (n)				
1 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) rašomų esybių skaičius (n)				
1 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) skaitomų esybių skaičius (n)				
2 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) duomenų laukų skaičius (n)				
2 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) rašomų esybių skaičius (n)				
2 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) skaitomų esybių skaičius (n)				
3 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) duomenų laukų skaičius (n)				
3 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) rašomų esybių skaičius (n)				
3 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) skaitomų esybių skaičius (n)				
4 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) duomenų laukų skaičius (n)				
4 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) rašomų esybių skaičius (n)				
4 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) skaitomų esybių skaičius (n)				
5 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) duomenų laukų skaičius (n)				
5 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) rašomų esybių skaičius (n)				
5 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, trinti) skaitomų esybių skaičius (n)				
6 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) duomenų laukų skaičius (n)				
6 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) rašomų esybių skaičius (n)				
6 vartotojo lango su viena funkcija (<u>įvesti</u> , redaguoti, <u>trinti</u>) skaitomų esybių skaičius (n)				

VARTOTOJO LANGAI SU 2 FUNKCIJOM (<u>įvesti ir redaguoti</u>; <u>įvesti ir trinti</u>; redaguoti ir trinti)²	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
1 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) duomenų laukų skaičius (n)				
1 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) rašomų esybių skaičius (n)				
1 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) tik skaitomų esybių skaičius (n)				
2 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) duomenų laukų skaičius (n)				
2 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) rašomų esybių skaičius (n)				
2 vartotojo lango su dviem funkcijom (<u>įvesti ir redaguoti</u> ; <u>įvesti ir trinti</u> ; redaguoti ir trinti) tik skaitomų esybių skaičius				

² Reikiamą reikšmę pabraukti, pavyzdžiui, jei lange galima tik įvesti ir redaguoti duomenis, pabraukti įvesti ir redaguoti

(n)				
...				

TRIJŲ FUNKCIJŲ VARTOTOJO ĮVEDIMO LANGAI (įvesti, redaguoti, trinti)	n	Analogiškų langų skaičius	Numatoma vieno lango realizavimo trukmė	Faktinė vieno lango realizavimo trukmė
1 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) duomenų laukų skaičius (n)				
1 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) rašomų esybių skaičius (n)				
1 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) tik skaitomų esybių skaičius (n)				
2 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) duomenų laukų skaičius (n)				
2 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) rašomų esybių skaičius (n)				
2 vartotojo lango su trim funkcijom (įvesti, redaguoti, trinti) tik skaitomų esybių skaičius (n)				
...				

3. Neinteraktyvaus vartotojo informacijos išvedimo funkciniai komponentai:

IŠVEDIMO FORMOS	n	Analogiškų formų skaičius	Numatoma vienos formos realizavimo trukmė	Faktinė vienos formos realizavimo trukmė
1 išvedimo formos duomenų laukų skaičius (n)				
1 išvedimo formos skaitomų esybių skaičius (n)				
2 išvedimo formos duomenų laukų skaičius (n)				
2 išvedimo formos skaitomų esybių skaičius (n)				
...				

ATASKAITOS	n	Analogiškų ataskaitų skaičius	Numatoma vienos ataskaitos realizavimo trukmė	Faktinė vienos ataskaitos realizavimo trukmė
1 ataskaitos duomenų laukų skaičius (n)				
1 ataskaitos skaitomų esybių kaičius (n)				
2 ataskaitos duomenų laukų skaičius (n)				
2 ataskaitos skaitomų esybių kaičius (n)				
...				

ELEKTRONINIAI LAIŠKAI, TEKSTINĖS ŽINUTĖS	n	Analogiškų laiškų skaičius	Numatoma vieno laiško realizavimo trukmė	Faktinė vieno laiško realizavimo trukmė
1 elektroninio laiško, žinutės duomenų laukų skaičius (n)				
1 elektroninio laiško, žinutės skaitomų esybių skaičius (n)				

2 elektroninio laiško, žinutės duomenų laukų skaičius (n)				
2 elektroninio laiško, žinutės skaitomų esybių skaičius (n)				
...				

4. Duomenų saugojimo paslaugos:

OBJEKTAI, VEIKLOS KLASĖS AR KITO TIPO LOGINIAI ĮRAŠAI	n	Analogiškų objektų skaičius	Numatoma vieno objekto realizavimo trukmė	Faktinė vieno objekto realizavimo trukmė
1 objekto duomenų laukų skaičius (n)				
2 objekto duomenų laukų skaičius (n)				
...				

5. Algoritmai

Algoritmai (saugumo tikrinimas, šifravimas, dešifravimas, skaičiavimas, formatų konvertavimas ir t. t.)	n	Analogiškų algoritmų skaičius	Numatoma vieno algoritmo realizavimo trukmė	Faktinė vieno algoritmo realizavimo trukmė
1 algoritmo duomenų elementų skaičius (n)				
1 algoritmo operacijų skaičius (n)				
2 algoritmo duomenų elementų skaičius (n)				
2 algoritmo operacijų skaičius (n)				
...				

6. Interfeisai į kitas programas

SIUNČIAMOS ŽINUTĖS	n	Analogiškų žinučių skaičius	Numatoma vienos žinutės realizavimo trukmė	Faktinė vienos žinutės realizavimo trukmė
1 žinutės skaitomų esybių skaičius (n)				
2 žinutės skaitomų esybių skaičius (n)				
...				

DUOMENŲ EKSPORTAVIMAS	n	Analogiškų rinkinių skaičius	Numatoma vieno rinkinio realizavimo trukmė	Faktinė vieno rinkinio realizavimo trukmė
1 duomenų rinkinio eksportuojamų elementų skaičius (n)				
2 duomenų rinkinio eksportuojamų elementų skaičius (n)				
...				

7. Interfeisai iš kitų programų

GAUNAMOS ŽINUTĖS	n	Analogiškų žinučių skaičius	Numatoma vienos žinutės realizavimo	Faktinė vienos žinutės realizavimo

			trukmė	trukmė
1 žinutės rašomų esybių skaičius (n)				
1 žinutės skaitomų esybių skaičius (n)				
2 žinutės rašomų esybių skaičius (n)				
2 žinutės skaitomų esybių skaičius (n)				
...				

DUOMENŲ IMPORTAVIMAS	n	Analogiškų rinkinių skaičius	Numatoma vieno rinkinio realizavimo trukmė	Faktinė vieno rinkinio realizavimo trukmė
1 failo rašomų esybių skaičius (n)				
1 failo skaitomų esybių skaičius (n)				
2 failo rašomų esybių skaičius (n)				
2 failo skaitomų esybių skaičius (n)				
...				

8. Visa programa

Studento vardas, pavardė	Sistemos pavadinimas (bakalauro darbo tema)	Numatoma visos programos realizavimo trukmė	Faktinė visos programos realizavimo trukmė

Dar kartą labai ačiū už sugaištą laiką

10.4. Priedas. Skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų

„IFPUG 4.11“

35 lentelė. Pagrindiniai skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų

Funkc. srities pritaikumas	Matavimo vienetas	Matavimo pozicija	BFC	BFC tipai		BFC tipų dedamoji	Kiekvienos dedamosios funkcionalumas	Pagr. apskaičiavimo išvedimas	Funkc. sudėtingumo svoris		Susijusios įvertinimo vertės
									Mažas	Didelis	
Verslo sistemų programinė įranga	IFPUG FP	Galinis vartotojas	EP	TF	EI	Įvesties žinutė	Elementarus procesas, kuris apdoroja duomenų ar kontrolės informaciją, kuri ateina iš sistemos išorės.	Skaičiuoti EIs skaičių	Mažas	3	
									Vidutinis	4	
									Didelis	6	
					EO	Išvesties žinutė	Elementarus procesas, kuris siunčia duomenų ar kontrolės informaciją iš sistemos. Pirminis tikslas pristatyti informaciją vartotojui per proceso logiką (matematinė formulė, skaičiavimas ar įgyti duomenys)	Skaičiuoti EOs skaičių	Mažas	4	
									Vidutinis	5	
									Didelis	7	
					EQ	Įvedimo/ Išvedimo pora	Elementarus procesas, kuris siunčia duomenų ar kontrolės informaciją iš sistemos. Išorinės užklauskos pristato informaciją vartotojui per paieškos duomenis ar kontrolės informaciją.	Skaičiuoti EQs skaičių	Mažas	3	
									Vidutinis	4	
									Didelis	6	
Verslo sistemų programinė įranga	IFPUG FP	Galinis vartotojas	EP	DF	ILF	Sistemos palaikomi duomenys	Grupė logiškai susietų duomenų ir kontrolės informacijos palaikomos sistemos. Pirminis tikslas valdyti duomenis, palaikomus vieno ar kelių elementarių procesų.	Skaičiuoti ILFs skaičių	Mažas	7	
									Vidutinis	10	
									Didelis	15	
					EIF	Kitų sistemų palaikomi duomenys	Grupė logiškai susietų duomenų ir kontrolės informacijos, bet palaikomos kitos sistemos. Pirminis tikslas valdyti duomenis, palaikomus vieno ar kelių elementarių procesų.	Skaičiuoti EIFs skaičių	Mažas	5	
									Vidutinis	7	
									Didelis	10	

„Mk II FPA 1.3.1“

36 lentelė. Pagrindiniai skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų (tęsinys)

Funkc. srities pritaikumas	Matavimo vienetas	Matavimo pozicija	BFC	BFC tipai	BFC tipų dedamoji	Kiekvienos dedamosios funkcionalumas	Pagr. apskaičiavimo išvedimas	Susijusios įvertinimo vertės
Stiprūs duomenys (Data – strong)	Mk II FP	Galinis vartotojas	LT	LT	Įvesties žinutė	Susideda iš duomenų kaupimo ir validavimo įeinančių duomenų, taip pat aprašant išorinius įvykius ar užklausų parametrus.	Skaičiuoti DETs skaičių įvesties žinutėje	0,58
					Išvesties žinutė	Susideda iš formataavimo ir informacijos pristatymo išorei.	Skaičiuoti DETs skaičių išvesties žinutėje	0,26
					Proceso dalis	Susideda iš saugyklos ir paieškos, aprašant kategorijų statusą išorei.	Skaičiuoti kreipinius į duomenų kategorijų tipus	1,66

„COSMIC FFP 2.2“

37 lentelė. Pagrindiniai skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų (tesinys)

Funkc. srities pritaikomumas	Matavimo vienetas	Matavimo pozicija	BFC	BFC tipai	BFC tipų dedamoji	Kiekvienos dedamosios funkcionalumas	Pagr. apskaičiavimo išvedimas	Susijusios įvertinimo vertės
Stiprūs duomenys (Data – strong),	Cfsu(COSMIC Functional Size Unit)	Galinis vartotojas ir projektuotojas	Duomenų judėjimo tipas	Iėjimas	Įvesties žinutė	Duomenų judėjimo tipas, kuris perkelia duomenų grupes iš vartotojo į būtinus funkcinius procesus. Perkeliama duomenys neatnaujinami.	Skaičiuoti įėjimus	1
Stipri Kontrolė (Control - strong)				Išėjimas	Išvesties žinutė	Duomenų judėjimo tipas, kuris perkelia duomenų grupes iš funkcinių procesų į būtiną vartotojui. Perkeliama duomenys neatnaujinami. Perkeliama duomenys neskaitomi.	Skaičiuoti išėjimus	1
Mišrus				Skaitymas	Išvesties žinutė iš saugyklos	Duomenų judėjimo tipas, kuris perkelia duomenų grupes iš saugyklos, pasiekiant būtinus funkcinius procesus.	Skaičiuoti skaitymus	1
				Rašymas	Įvesties žinutė iš saugyklos	Duomenų judėjimo tipas, kuris perkelia duomenų grupes iš saugyklos, esančias viduje funkcinių procesų į saugyklą.	Skaičiuoti rašymus	1

„*FiSMA 1.1*“ (kadangi šis metodas turi 7 BFC klases ir 28 BFC tipus, kaip pavyzdys, skirtumų stebėjimui tarp prieš tai aprašytų metodų, patiekamas vienos klasės aprašas)

38 lentelė. Pagrindiniai skirtumai tarp IFPUG FPA, Mk II FPA, COSMIC FFP, FiSMA FSM metodų (tęsinys)

Funkc. srities pritaikumas	Matavimo vienetas	Matavimo pozicija	BFC	BFC tipai	BFC tipų dedamoji	Kiekvienos dedamosios funkcionalumas	Pagr. apskaičiavimo išvedimas	Susijusios įvertinimo vertės
Bet kokia sritis	FiSMA	Galinis vartotojas	Interaktyvios galinio vartotojo navigacijos ir užklausų paslaugos	Funkcinės rodyklės (Ikonos)	(q1)	Tai paslaugos, kurios užtikrina specifinį, vizualų būdą vartotojui, nurodant specifinių paslaugų atlikimą.	Skaičiuoti q1 – q7 (duomenų elementai n) ir skaitymo nuorodos (r) Procesas: $0,2+n/7+r/2$	1
				Prisijungimo ir atsijungimo funkcijos	(q2)	Dažnai neatnaujinami duomenys, kontroliuoja vartotojo prisijungimą ir teisėtą naudojimą.		1
				Funkcijų sąrašas (Menu...)	(q3)	Paslaugos pateikti rinkinį, iš anksto apibrėžtų alternatyvų, įgalinti vartotoją nurodyti konkrečias paslaugas.		1
				Pasirinkimų sąrašas (List Boxes..)	(q4)	Galiniam vartotojui parodo priimtinas parametrų vertes.		1
				Duomenų užklausos	(q5)	Galiniam vartotojui parodo konkretų turinį duomenų saugyklos		1
				Generavimo indikatoriai (Generavimo dialogai)	(q6)	Padeda vartotojui paruošti duomenis ir/arba kontroliuoti informaciją vėlesnėms paslaugoms. Dažnai jie yra sujungti su kito tipo funkcinėmis paslaugomis, tokiomis kaip ataskaita.		1
				Naršymo sąrašai	(q7)	Parodo sąrašą panašių duomenų elementų, paprastai svarbiausios detalės padėti filtruoti kategorijas tolimesnėms operacijoms.		1