

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Raimondas Savulis

**Projektų dokumentavimo sistema  
„MagicDraw“ CASE įrankio pagrindu**

Magistro darbas

Darbo vadovė

prof. dr. L. Nemuraitė

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Raimondas Savulis

**Projektų dokumentavimo sistema  
„MagicDraw“ CASE įrankio pagrindu**

Magistro darbas

Recenzentė

doc. dr. R. Misevičienė

2011-05-26

Vadovė

prof. dr. L. Nemuraitė

2011-05-25

Atliko

IFM-9/4 gr. stud.  
Raimondas Savulis

2011-05-25

Kaunas, 2011

## Turinys

1.	Įvadas .....	7
2.	Dokumentų rengimo procesų ir dokumentų tipų analizė .....	11
2.1.	Dokumentacijos rengimo proceso analizė .....	11
2.2.	KIS kūrimo ir dokumentavimo veiklų susiejimas .....	13
2.3.	Projektavimo procesų ir su jais susijusių dokumentų analizė .....	17
2.3.1.	Kaskadinis (Software Development Life Cycle - SDLC) .....	17
2.3.2.	Persidengimų (Overlap) ciklas .....	18
2.3.3.	Evoliucinis ciklas .....	18
2.3.4.	Laipsniškas kūrimas .....	19
2.3.5.	Apribojimais grindžiamas ( <i>Bounding Box</i> ) kūrimas .....	20
2.3.6.	Formalus sistemų kūrimas .....	21
2.3.7.	Unifikuotas kūrimo procesas (RUP) .....	22
2.3.8.	Lankstieji ( <i>Agile</i> ) kūrimo procesai .....	24
2.3.9.	<i>Volere</i> šablonas .....	24
2.3.10.	ISO/IEC 12207 standarte apibrėžti procesai ir dokumentų tipai .....	24
2.3.11.	Kūrimo procesų dokumentavimo metodų palyginimas .....	34
2.4.	Galimų įgyvendinimo priemonių variantų analizė .....	35
2.5.	Dokumentavimo priemonių kūrimo formuluotė .....	36
2.6.	Nefunkciniai reikalavimai ir apribojimai .....	37
2.6.1.	Reikalavimai standartams .....	37
2.6.2.	Reikalavimai veikimui .....	37
2.6.3.	Reikalavimai sąveikai su kitomis sistemomis .....	37
2.6.4.	Reikalavimai panaudojamumui .....	37
2.7.	Dokumentavimo proceso ir dokumentacijos kokybės kriterijai .....	38
2.8.	Analizės išvados .....	39
3.	Projekto dokumentų kūrimo metodika .....	40
3.1.	Dokumentavimo procesas .....	41
3.2.	Projektų dokumentavimo metodikos taikymas .....	43
3.3.	Dokumento šablono koregavimas .....	46
3.4.	Projekto programavimo ir testavimo darbų apimčių skaičiavimas .....	47
3.5.	Projekto elementų ryšių pilnumo nustatymas .....	49
4.	Pavyzdinės apklausų informacinės sistemos projekto aprašas .....	51
4.1.	Pavyzdinės apklausų informacinės sistemos funkciniai reikalavimai .....	51
4.2.	Pavyzdinės apklausų informacinės sistemos reikalavimų specifikavimas, naudojant <i>Volere</i> šabloną .....	52
4.3.	Pavyzdinės apklausų informacinės sistemos projektas .....	58
5.	Pavyzdinės sistemos ataskaitų šablonų realizavimas <i>MagicDraw UML</i> įrankiu .....	63
5.1.	<i>MagicDraw UML</i> ataskaitų vedlio ( <i>Report Wizard</i> ) taikymas šablonams kurti .....	63
5.2.	Dokumento šablono importavimas .....	64
5.3.	Dokumento generavimas .....	67
5.4.	Dokumento šablono kintamieji .....	69
5.5.	Projektų dokumentavimo sistemos šablonas .....	73
5.5.1.	Šablonas su fiksuota projekto struktūra .....	73
5.5.2.	Šablonas projektams su kintančia struktūra .....	76
5.6.	Projekto dokumentavimo ir testavimo darbų apimties skaičiavimo realizacija .....	78

5.7. Projekto elementų ryšių pilnumo nustatymo realizacija.....	83
6. Eksperimentinis Projektų dokumentavimo sistemos tyrimas .....	85
6.1. Eksperimento apibrėžimas .....	85
6.2. Eksperimento planas .....	85
6.3. Eksperimento vykdymas .....	85
6.4. Eksperimento interpretavimas ir pateikimas.....	86
6.5. Eksperimento išvados .....	91
7. Išvados .....	92
8. Literatūra.....	93
9. Priedai .....	94
9.1. Priedas. Apklauso anketa .....	94
9.2. Priedas. Mokslinėje konferencijoje „Mag&Dok IT2011“ pristatytas straipsnis .....	95
9.3. Priedas. Pavyzdinės apklausų informacinės sistemos sekų ir veiklos diagramos....	100

## **Summary**

### **Project Documentation System Based On MagicDaw UML CASE Tool**

Creation of documentation usually is inefficient - documents are being created slowly, the methodology of documentation is usually redundant, there is no common strategy for this. The paper proposes a way to boost the process of creating documentation by using documentation templates that give an opportunity to generate documentation directly from a CASE tool. The created template implements additional functionality - the estimation of a size of programming and testing tasks of a project and tracing the dependencies between project's elements. The implemented template of a document may be applied in more than one project.

Key words: projects documentation, document template, project estimation, Report Wizard, artifact traceability

## Terminų ir santrumpų žodynis

	Terminas/santrumpa	Paaīškinimas
1.	KIS	Kompiuterizuota informacinė sistema
2.	RUP	<i>Rational Unified Process</i> - kartotinio projektų kūrimo metodika, sukurta įmonės <i>Rational Software</i> .
3.	<i>Agile</i>	Projektų kūrimo metodologija, pasiūlyta nepelno organizacijos „Agile Alliance“. Metodai buvo kuriami tiems atvejams, kai programų kūrimas sunkiai valdomas, reikalavimai sparčiai kinta, tokiais atvejais lankstusis kūrimas supaprastina tradicinę projektavimo inžineriją.
4.	ISO/IEC	<i>International Organization for Standardisation/ International Electrotechnical Commission</i> - tarptautinės standartų organizacijos, atsakingos už kokybės standartus.
5.	PĮ	Programinė įranga.
6.	BEAM	<i>Basic Enterprise Architecture Methodology</i> .
7.	TOGAF	<i>The Open Group Architecture Framework</i> – verslo architektūros metodika, kurioje pateikiami išsamūs nurodymai informacijos architektūrai projektuoti, planuoti, realizuoti ir valdyti.
8.	IS	Informacinė sistema.
9.	XML	<i>Extensible Markup Language</i> - W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.
10.	IT	Informacinės technologijos.
11.	VTL	<i>Velocity Template Language</i> – JAVA pagrįsta šablonų kūrimo kalba, kurios pirminis tikslas yra generuoti internetinius puslapius iš šablonų.
12.	Projektas su fiksuota struktūra	Projektas, kuris kurimas pagal iš anksto numatytą struktūrą, kuri suformuojama remiantis ankstesnių projektų patirtimi.
13.	Projektas su kintančia struktūra	Projektas, kurio struktūra projektavimo metu keičiama ir negalima pritaikyti iš anksto sukurtos struktūros.
14.	<i>Volere</i> šablonas	Šablonas sistemos funkcinių ir nefunkcinių reikalavimų aprašymui.

## 1. Įvadas

Galutinio vartotojo informacija, arba dokumentacija, pateikiama kartu su KIS (kompiuterizuota informacine sistema) kaip neatsiejama jos dalis. Iš tikrųjų KIS negalima pristatyti, kol neišbaigta jos dokumentacija. Tačiau dauguma KIS kūrimo procesų nelaiko rimtomis veiklų, kurios sudaro vartotojo informacijos kūrimo procesą. Dėl to vartotojo informacija labai dažnai sukuriama paskubomis projekto pabaigoje, dėl ko kyla reali grėsmė, kad dokumentacija bus neišbaigta, netiksli, todėl paprasčiausiai nenaudinga.

**Tyrimo problema** – kad dokumentacijos kūrimas vykdomas neefektyviai – dokumentai kuriami rankiniu būdu, nėra vieningos strategijos dokumentacijos kūrimui, dėl ko dokumentacija kartais laikoma viena iš projekto rizikų – sunku rasti optimalų sprendimą, kaip paskirstyti laiką taip, kad dokumentacija būtų paruošta kuo greičiau ir kuo geresnės kokybės. Kadangi praktikoje didesnė projektų dalis nesutelpa į laiko rėmus, todėl ieškoma būdų, kaip paspartinti projekto darbų apimtį, o vienas iš būdų – aukojant dokumentacijos kokybę. Ironiška, tačiau dokumentacija, kartu su vartotojo sąsaja, yra labiausiai prieinama ir matoma viso projekto dalis, todėl prastesnės kokybės dokumentacija tikrai ne išeitis – radus būdą, kaip efektyviau išnaudoti laiką, skirtą dokumentacijai paruošti, būdų galima išsaugoti dokumentacijos kokybę. Šiam būdui surasti, darbo **tyrimo sritis** pasirinkta informacinių sistemų projektavimo ir dokumentavimo metodai, o **tyrimo objektas** – dokumentacijos rengimo procesas. Atlikus tyrimą, bus galima pasiekti darbe iškeltą **tikslą** – patobulinti ataskaitų kūrimo procesą, sudarant IS projektų dokumentavimo CASE įrankiu metodiką ir pritaikant ją UML CASE įrankiui „MagicDraw UML“.

Darbe buvo iškelti šie **uždaviniai**:

- Suformuluoti ataskaitų kūrimo procesą.
- Suprojektuoti pavyzdinę sistemą.
- Sukurti pavyzdinės sistemos ataskaitų šablonus ir pagal juos sugeneruoti dokumentaciją.
- Atlikti eksperimentą sukurtai Projektų dokumentavimo sistemai įvertinti.

Tam, kad būtų susieti dokumentavimo ir projektavimo procesai, galima arba išplėsti projektavimo etapus, t.y. į tuos etapus įtraukti už dokumentaciją atsakingus projekto narius ir dokumentacijos ruošimo veiklas, arba sukurti naujas atskiras

dokumentavimo veiklas, kurios būtų glaudžiai susiję su kūrimo veiklomis. Šis susiejimo procesas buvo detalizuotas nurodant atskirų projekto kūrimo etapų dalyvius bei jų kuriamus artefaktus ir lygiagrečiai pateikiant galimus atitinkamo etapo dokumentavimo proceso dalyvius bei jų kuriamus artefaktus, aktualius dokumentui.

Išanalizavus kūrimo procesų metodus, įsitikinta, kad dokumentavimo procesas nėra pakankamai reglamentuotas. Tradicinės projektų kūrimo strategijos, tokios kaip kaskadinis modelis[9], persidengimų [11], evoliucinis[12], laipsniškas ir kt. net neįtraukia dokumentacijos kūrimo proceso į bendrą projekto kūrimo ciklą, todėl gali būti naudingi tik kaip strategijos pačiai dokumentacijai ruošti, neįtraukiant dokumentacijos proceso į bendrą projekto gyvavimo ciklą. Kiti metodai, tokie kaip RUP [2][6] ar lankstieji (*Agile*)[1], nors ir pateikia gaires dokumentacijos kūrimui, tačiau, RUP atveju, dokumentacijos kūrimui pateikiami šablonai yra pertekliniai ir sunkiai pritaikomi konkrečiam projektui, tuo tarpu *Agile* dokumentacija yra nepakankama, nes, nors dokumentacija ir yra svarbi *Agile* projektų kūrimo strategijos dalis, tačiau, priešingai nei tradicinių projektų kūrimo strategijų šalininkai, kurie dokumentaciją supranta kaip rizikos mažinimo priemonę, *Agile* strategijoje dokumentacija paprastai yra veiksnys, padidinantis viso projekto riziką, ir todėl dokumentaciją stengiamasi paruošti kaip įmanoma greičiau, dėl ko dokumentacija nebūna pilna ir išsami. Egzistuojantys standartai, pavyzdžiui ISO/IEC 12207 [4], pateikia tik bendrus nurodymus, kokius procesus, jų aspektus, dokumentuoti, tačiau išsamesnių nurodymų dokumento sudėčiai, jau nekalbant apie kokius nors dokumentų šablonus, nėra pateikta.

Vienas iš būdų dokumentacijos procesui paspartinti yra dokumentacijos generavimas iš projektavimo įrankio, panaudojant dokumentų šablonus. Nors ir egzistuoja nemažai projektavimo įrankių, pavyzdžiui MICROSOFT OFFICE VISIO, VISUAL PARADIGM, ALTOVA UMODEL MAGICDRAW UML, suteikiančių galimybę generuoti dokumentaciją, tačiau, nepaisant to, kad pateikiamos naudotojo instrukcijos, trūksta teorinio pagrindo dokumentacijos generavimo ir šablonų naudojimo integravimo į bendrą projekto kūrimo procesą.

Atliekant galimų įgyvendinimo priemonių variantų analizę, buvo ištirtos keturių įrankių galimybės generuoti dokumentus tiesiai iš projektavimo įrankio. MICROSOFT OFFICE VISIO FOR ENTERPRISE ARCHITECTS įrankyje realizuotos dokumentų generavimo priemonės yra žemo lygio, yra tik minimali galimybė modifikuoti gamintojo sukurtus šablonus, todėl buvo nuspręsta, kad šio



įrankio teikiamų galimybių tikrai nepakaks sistemai įgyvendinti. VISUAL PARADIGM FOR UML 8.1 ir ALTOVA UMODEL 2011 įrankiuose dėl itin patogios vartotojo sąsajos galima nesunkiai sumodeliuoti norimą generuoti dokumentą. Tačiau, nors vartotojo sąsaja ir supaprastina dokumentų sudarymą, tačiau vartotojo veiksmai, susiję su dokumentų personalizavimu, lieka apriboti įrankio kūrėjo, todėl sistemos praktiniam panaudojimui iliustruoti buvo pasirinktas *MagicDraw UML* įrankis, kuriame dokumentų šablonams sukurti naudojama VTL kalba, kuria galima sudaryti individualesnius šablonus, nes VISUAL PARADIGMA ar ALTOVA UMODEL vartotojo sąsajos teikiamų galimybių gali nepakakti, norint sumodeliuoti specifinius dokumentus.

Atlikus kūrimo procesų metodų analizę, buvo suformuluoti nurodymai dokumentacijos kūrimui, įtraukiant dokumentų generavimą bei dokumentų šablonų panaudojimą. Praktinį metodikos panaudojimą buvo nuspręsta realizuoti sukūrus pavyzdinę sistemą, kuriai buvo sukurtas dokumento šablonas bei sugenerotas pilnas sistemos dokumentas. Be įprastinių dokumento sudedamųjų dalių, taip pat buvo realizuotas automatinis projekto kūrimo [13] ir testavimo [14] darbų apimčių skaičiavimas bei projekto elementų ryšių pilnumo analizė. Dėl to, kad projekto kūrimo ir testavimo darbai automatiškai apskaičiuojami generuojant dokumentą, išvengiama būtinumo įsigyti arba kurti papildomus programinius produktus šiems darbas atlikti. Elementų ryšių pilnumo analizė padeda išvengti klaidų arba neatitikimų pereinant nuo vieno projektavimo etapo prie kito – taip užtikrinama, kad visi elementai, esantys viename projektavimo etape, bus įtraukti į kita etapą bei neliks nereikalingų elementų.

Tam, kad galimam Projektų dokumentavimo sistemos vartotojui būtų paprasčiau adaptuoti sistemą savo darbinėje aplinkoje, pateikti nurodymai, kaip importuoti šabloną, koreguoti šablono kintamuosius, generuoti dokumentą, viską iliustruojant *MagicDraw* įrankio vartotojo sąsajos vaizdais.

**Darbas suskirstytas į 9 skyrius.** Pirmasis skyrius – įvadas. Antrame skyriuje buvo išanalizuoti projektavimo procesai, atliktas jų palyginimas, tyrinėjami dokumentų tipai, apžvelgiamos priemonės, kurių pagalba būtų galima realizuoti sistemos praktinį panaudojimą, taip pat suformuluotas siekiamos sistemos apibrėžimas, nustatyti nefunkciniai reikalavimai bei apribojimai bei suformuluoti dokumentavimo proceso ir dokumentacijos kokybės reikalavimai. Trečiajame skyriuje pateikiama reikalavimų specifikacija, apibrėžta pavyzdinė sistema, panaudojant

*Volere* šabloną [5], suformuluoti reikalavimai pavyzdinei sistemai bei sukurtas pavyzdinės sistemos projektas. Ketvirtame skyriuje, aprašius sprendimą, buvo iliustruota, kaip turėtų vykti pradinis bei pakartotinis sistemos panaudojimas. Penktame skyriuje, aprašant sprendimo realizaciją, detalizuojamas *MagicDraw* įrankio ataskaitų generavimo procesas bei pateikiamos iliustracijos su *MagicDraw* įrankio vartotojo sąsajos vaizdais, paaiškinančiais, kaip panaudoti pagrindinį Projektų dokumentavimo sistemos artefaktą – šabloną. Be to, paaiškinta logika, kuria remiantis buvo realizuotas sistemos kūrimo bei testavimo darbų apimčių skaičiavimas bei projekto elementų ryšių pilnumo nustatymas. Šeštame skyriuje pateiktas eksperimentinis tyrimas. Septintame skyriuje suformuluotos darbo išvados, apibendrinančios tyrimo dokumentacijos rengimo procesų ir dokumentų tipų analizės, sprendimų reikalavimų specifikacijos, sprendimo aprašo ir sprendimo realizacijos rezultatus. Aštuntame skyriuje – darbe naudotos literatūros sąrašas, o devintame – priedai.

Darbo tema atspausdintas straipsnis ir perskaitytas pranešimas konferencijoje „Mag&Dok IT2011“ [15].

## 2. Dokumentų rengimo procesų ir dokumentų tipų analizė

### 2.1. Dokumentacijos rengimo proceso analizė

Analizės tikslas – išsiaiškinti ir išanalizuoti sistemų projektavimo ir dokumentavimo procesus, kad būtų galima geriau įvertinti jų trūkumus bei rasti būdus tiems trūkumams pašalinti, siekiant paspartinti informacinių sistemų projektų dokumentavimo procesą bei jo realizavimą.

Šio darbo **tyrimo sritis** – informacinių sistemų projektavimo ir dokumentavimo metodai, tyrimo objektas – dokumentacijos rengimo procesas.

Kaip matyti 2.1 paveiksle, dokumentavimo procesą galima išskaidyti į smulkesnes veiklas, kurios atspindi skirtingų dokumentų rengimą sistemos gyvavimo ciklo metu.



2.1 pav. Dokumentacijos ruošimo proceso analizė

Kiekviena iš šių veiklų dokumentuoja tam tikrą sistemos gyvavimo ciklo etapą. Nors visos veiklos orientuotos į dokumentų ruošimą, tačiau priemonės, kurios naudojamos skirtingų dokumentų ruošimui, gali skirtis, nes dokumentai skiriasi savo formalumo lygiu – jei vieniems yra aktualiau informacija, pateikiama tekstine forma

(reikalavimų dokumentavimas), kitiems aktualiau formalios (sistemos projekto dokumentavimas) arba vaizdinės (vartotojo instrukcija) informacijos formos.

Dokumentavimo procesai aktualūs vartotojams, kurių aibę galima suskaidyti į dvi dalis pagal tai, kurią dokumentacijos dalį išskirtoji vartotojų grupė naudos. Akivaizdu, kad viena dokumentacijos dalis – vartotojo vadovas – labiau orientuota į galutinį sistemos vartotoją, kuris gali būti tiek dalykinės srities specialistas, tiek eilinis vartotojas, priklausomai nuo to, kokia kuriamos sistemos paskirtis. Kitai vartotojų daliai – sistemos užsakovams – orientuota visa likusi dokumentacijos dalis, kuri paprastam vartotojui gali būti per daug sudėtinga dėl specifinių terminų ar dalykinės srities žinių stokos. Tačiau sistemos užsakovai suinteresuoti kuo geresne vartotojo vadovo kokybe, nes nuo to priklauso sistemos naudojamumas bei visos sistemos komercinė sėkmė (jei sistema kuriama ne savo reikmėms).

Taigi, jei iš dokumentacijos  $D$  išskirtume jos poaibį  $VV$  (vartotojos vadovas), t.y.  $VV \subset D$  bei išskirtume dvi vartotojų grupes  $GV$  (galutinei vartotojai) bei  $SU$  (sistemos užsakovai), tuomet turime, kad  $D \rightarrow SU$ , o  $VV \rightarrow GV$ , kur  $\rightarrow$  reiškia skirta.

Dokumentavimo procesų vartotojų problemos pateikiamos lentelėje, kurios dešinėje pusėje pateikiama problema, o kairėje – dabartinė situacija.

**2.1 lentelė. Problema ir dabartinė situacija**

<b>Problema</b>	<b>Kaip viskas vyksta dabar</b>
Dokumentavimas vykdomas lėtai ir neefektyviai.	Dokumentavimo procese naudojamos neefektyvios priemonės, dokumentus tenka sudarinėti rankiniu būdu.
Esama dokumentavimo metodika dažniausiai yra perteklinė, nepritaikyta konkrečiam dokumentavimo metodui.	Siūlomi dokumentavimo būdai, tokie kaip RUP, Volere šablonas, Agile procesų dokumentacija yra pertekliniai, nepritaikyti konkrečiam dokumentavimo metodui.
Dėl to, kad dokumentavimas vykdomas lėtai ir neefektyviai, pačios dokumentacijos kokybė dažniausiai prasta.	Dėl laiko stokos dokumentacija dažniausiai vykdoma projekto pabaigoje ir paskubomis.

Viena iš problemų, kuri kyla modeliuojant projektų dokumentavimo sistemą, yra identifikavimas dokumentų, kurie aktualūs konkrečiam projektui, ir paties dokumentacijos rengimo proceso bei jo etapų identifikavimas.

Dokumentacijos ir pačios KIS kūrimo procesai naudoja nemažai tų pačių artefaktų. Dažniausiai išskiriamos tokios pagrindinės veiklos, kurios charakterizuoja pagrindinį darbą, kurį reikia atlikti kuriant KIS:

- Projekto apibrėžimas;
- Reikalavimų specifikavimas;
- Analizė ir projektavimas;
- Realizavimas;
- Testavimas;
- Diegimas;

Nepaisant to, kad šios veiklos kartojasi skirtingose KIS kūrimo metodikose, tačiau trūksta vieningos dokumentavimo sistemos, kaip dokumentuoti KIS kūrimą – tos metodikos apibrėžia, kaip kurti, bet ne kaip dokumentuoti KIS galutiniams vartotojams.

Dokumentavimas turėtų būti vykdomas lygiagrečiai su KIS kūrimu. Iš esmės, dokumentacijos kūrimo procesas turi tokias pačias pagrindines veiklas, kaip ir pats kūrimas – projektavimas, rašymas, testavimas, pristatymas ir valdymas.

Susiejant dokumentavimą su tokiu pat tikslu procesu, kaip ir paties kūrimo, neabejotinai atsiranda valdymo poreikis bei poreikis pakartoti kai kuriuos proceso žingsnius. Tačiau vien to neužtenka – dokumentavimas nėra atskiras procesas – jis labai priklauso nuo kūrimo proceso. Susietas dokumentavimo ir projekto kūrimo procesas turi ne tik identifikuoti paralelias dokumentavimo veiklas, bet ir nustatyti, kaip dokumentavimas ir kūrimas yra susijęs. Dėl to atsiranda nauji tikslai, tokie kaip:

- Dokumentacija turi kuo tiksliau apibūdinti KIS;
- KIS turi vykdyti tai, kas aprašyta dokumentacijoje.

Galima išskirti tokius dokumentacijos tikslus:

- Dokumentaciją sukurti kaip išbaigtą, nuoseklų ir svarbiausia tikslų informacijos rinkinį;
- Užtikrinti, kad informacija atitiks vartotojo poreikius naudojantis KIS;
- Pateikti kokybišką informaciją, sunaudojant kuo mažiau finansinių ir laiko išteklių.

## **2.2. KIS kūrimo ir dokumentavimo veiklų susiejimas**

Kadangi dokumentavimo ir kūrimo veiklos sutampa, todėl yra glaudus ryšys tarp egzistuojančių kūrimo veiklų ir veiklų, kurios reikalingos dokumentavimui.

Dokumentavimo ir kūrimo veiklų susiejimas gali būti įgyvendintas arba išplečiant kūrimo veiklas, t.y. į tas veiklas įtraukiant dokumentavimo aktorius ir veiklas, arba sukuriant naujas atskiras dokumentavimo veiklas, kurios būtų glaudžiai susiję su keliomis kūrimo veiklomis.

Tam, kad būtų lengviau suprasti KIS kūrimo ir dokumentavimo veiklų galimą susiejimą, pateikiamas kiekvienos kūrimo veiklos aktorius ir jo artefaktai, kurie bus susieti su dokumentavimo veiklomis [2].

**2.2 lentelė** KIS kūrimo ir dokumentavimo veiklų susiejimas

<b>Projekto kūrimo aktorius</b>	<b>Aktoiaus kuriamas artefaktas</b>	<b>Projekto dokumentavimo aktorius</b>	<b>Aktoiaus kuriamas artefaktas</b>
<b>Projekto apibrėžimas</b>			
<i>Projekto analitikas</i>	Žodynas, nurodantis projekto krypties terminologiją.	<i>Informacijos architektas</i>	<b>Analizės artefaktai:</b>
	Projekto panaudojimo atvejų modelis, apibūdinantis pagrindines kuriamos KIS veiklas.		Žodynas, apibūdinantis dokumentacijoje naudojamus terminus.
<i>Projektuotojas</i>	KIS aktoriai, projekto esybės ir organizaciniai vienetai, apibūdinantys KIS vartotojus ir jų organizacijas.		Tam tikrų veiklų sekų analizė, apibūdinanti veiklas, kurias turi atlikti KIS vartotojai, bei ryšį tarp tų veiklų.
	Projekto panaudojimo atvejai, apibrėžiantys, kokie vartotojai ir esybės susiję su atskirais panaudojimo atvejais.		<b>Projektavimo artefaktai:</b>
	Projekto panaudojimo atvejų realizacijos, pritaikytos tam tikroms vartotojų ir esybių grupėms.		Vartotojų analizė, apibrėžianti vartotojus, kuriems skirtas produktas, bei tų vartotojų grupes.
			Preliminarūs veiklų modeliai (dažniausiai tekstinis panaudojimo atvejo apibūdinimas). Preliminarios instrukcijos, kaip elgtis tam tikrais produkto naudojimo atvejais.
Vienas iš svarbiausių susiejimų yra tarp panaudojimo atvejų ir jų atitiktens dokumentavime – veiklų sekų. Panaudojimo atvejai apibūdina galimas veiksmų sekas atliekant konkrečią veiklą. Kai visų panaudojimo atvejų informacija			

sujungiama, gaunama viso produkto veiklų seka (bei ryšiai tarp šių veiklų), tiksliau veiklų dokumentacija (žingsniai, kuriuos reikia atlikti, norint įvykdyti konkrečią užduotį).

<b>Reikalavimų specifikavimas</b>			
<i>Sistemų analitikas</i>	Žodynas	<i>Informacijos architektas</i>	<b>Analizės artefaktai:</b>
	Projekto panaudojimo atvejų modelis.		Žodynas.
	Panaudojimo atvejų modelis, sukurtas pagal projekto panaudojimo atvejų modelį, tačiau įtraukiant numatomą funkcionalumą pasiūlytam produktui.		Veiklų sekų analizė.
	Reikalavimų savybės, susijusios su konkrečiais vartotojo reikalavimais ir panaudojimo atvejais, kurie juos įgyvendina.		Reikalavimai dokumentacijai (iš kūrimo reikalavimų – kiekvienas reikalavimas kūrimui, kuris įtakoja vartotojo sąsają arba veiklų seką, turi atitinkamą reikalavimą dokumentacijai).
	Užsakovo reikalavimų keitimo fiksavimas.		Užsakovo reikalavimų keitimas (pagal reikalavimų kūrimui kitimą – kiekvienas pakitęs reikalavimas, įtakoiantis vartotojo sąsają arba veiklų seką, turi atitinkamą pasikeitusį reikalavimą dokumentacijai).
<i>Vartotojo sąsajos projektuotojas</i>	Aktoriai, apibūdinantys sistemos vartotojus, ir kurie panaudojimo atvejai susiję su šiais aktoriais.	<i>Informacijos architektas</i>	<b>Vartotojai sąsajos projektavimo artefaktai:</b>
	Pirminis vartotojo sąsajos prototipas.		Vartotojų analizė, pridėdant, kurios veiklų sekų dalys skirtos konkreitiems vartotojams.
	Panaudojimo atvejų vykdymo sekos fiksavimas, apibūdinantis, koku būdu bei koks panaudojimo atvejis bus įgyvendintas vartotojo sąsajoje.		Dokumentacijos prototipas.
			Apibūdinimas, koku būdu dokumentacija bus naudojama dirbant su KIS.
			Veiklų pavyzdžiai su vartotojos sąsajos elementais.

<b>Analizė ir projektavimas</b>			
<b>Architektas</b>	Analizės modelis, apibūdinantis klasių hierarchiją, kuri gali būti numanoma iš panaudojimo atvejų analizės.	<b>Informacijos architektas</b>	Principinis modelis ir principai (iš klasių hierarchijos ir žodyno).
	Projekto modelis, analogiškas analizės modeliui, tik iš projektavimo pusės.		Veiklų ir principų susiejimas (iš panaudojimo atvejų realizacijos).
<b>Projektuotojas</b>	Klasių projektavimas ir analizė (klasių pavadinimai, ryšiai tarp klasių, klasių metodai, klasių duomenys).		Principų ir atitinkamos informacijos susiejimas (iš klasių duomenų arba duomenų modelio).
	Panaudojimo atvejų realizavimas, apibūdinantis, kiek klasių siesis įgyvendinant konkretų panaudojimo atvejį.		Veiklų žingsnių ir klasių metodų susiejimas (iš klasių metodų ir panaudojimo atvejų realizavimo).
<b>Duomenų bazės projektuotojas</b>	Duomenų modelis, nusakantis, kokie duomenys turi būti išsaugomi arba kurie duomenys būtini.		
<p>Šiame žingsnyje architektas ir projektuotojai kuria klasių ir duomenų modelius, kuriuos dauguma kūrimo metodikų siūlo nedokumentuoti, nes projektavimo elementai, tokie kaip klasių hierarchija, nėra svarbūs vartotojo sąsajai. Tačiau procesas, pagal kurį analizuojama ir projektuojama klasių hierarchija bei duomenų modelis, yra gana panašus į procesą, pagal kurį informacijos architektas nustato pagrindinius principus, pagal kuriuos atrenkama, ką reikia dokumentuoti, ir reikalavimus atitinkamai informacijai. Nors ne visos klasės atitinka dokumentavimo principus, didelė jų dalis kyla iš panaudojimo atvejo ir žodyno susiejimo. Tai tie patys informacijos šaltiniai (veiklų sekų analizė ir žodynas), kuriuos informacijos architektas galėtų panaudoti identifikuodamas minėtus principus.</p>			



## **Realizavimas, testavimas ir diegimas**

Likusios trys veiklos, kurios taip pat yra labai svarbios, neturi tokio griežto susiejimo tarp dokumentavimo ir kūrimo procesų.

Dokumentavimo procesui reikalingi trys papildomi aktoriai:

### **Realizavimas**

- Rašytojas, kuris įgyvendina informacijos architektūrą.
- Dizaineris, kuris ieško arba kuria iliustracijas.
- Redaktorius, kuris prižiūri kokybę.

### **Testavimas**

- Dokumentacijos testuotojas, kuris tikrina informacijos funkcionalumą ir integralumą.

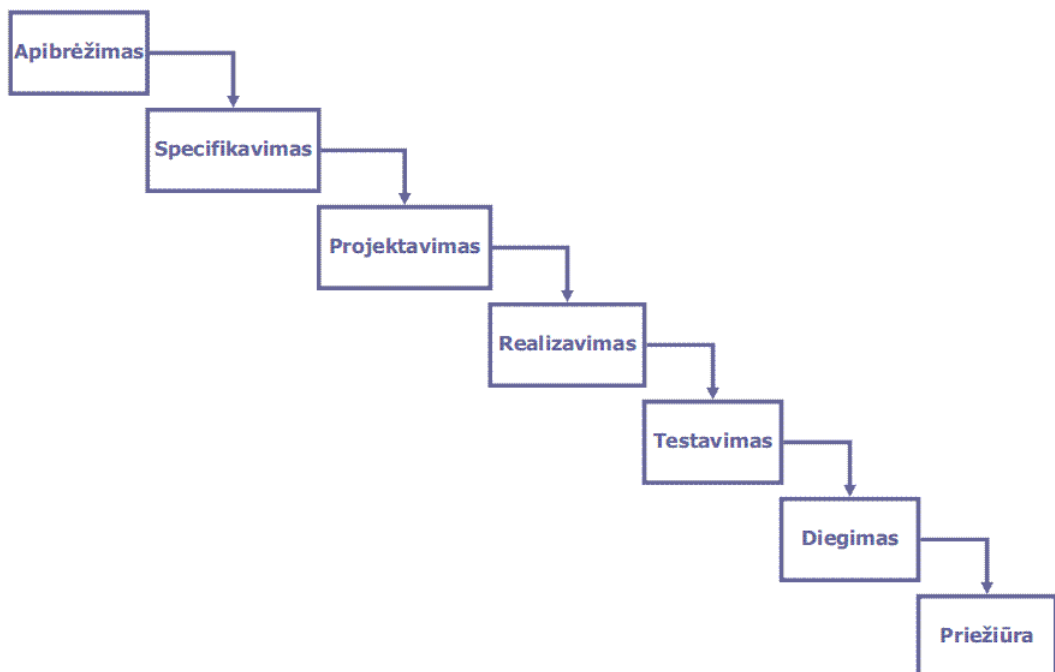
### **Diegimas**

- Produkcijos specialistas, kuris prižiūri įdiegtus automatizuotus dokumentavimo procesus.

## **2.3. Projektavimo procesų ir su jais susijusių dokumentų analizė**

Kaip buvo minėta, dokumentavimo ir paties KIS kūrimo procesai naudoja nemažai tų pačių artefaktų, todėl esamų sprendimų analizė atliekama KIS kūrimo tipų ir metodų pagrindu.

### **2.3.1. Kaskadinis (Software Development Life Cycle - SDLC)**

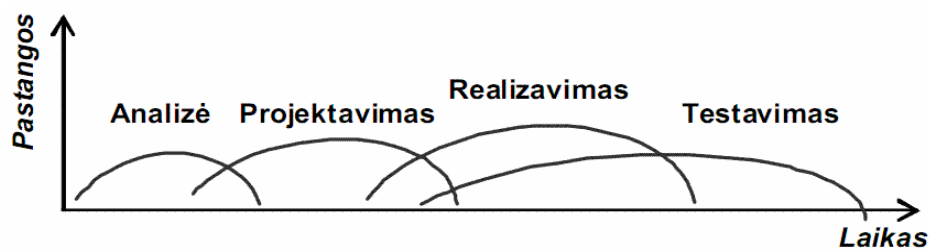


**2.2 pav. Kaskadinis modelis**

Naudojant kaskadinį modeli, kyla sunkumų skaidant projektą į skirtingas stadijas. Tai kelia sunkumų norint reaguoti į vartotojų reikalavimų pasikeitimus. Šis modelis taikomas tik tada, kai reikalavimai yra aiškiai suprantami.

Kaskadinis modelis gali būti praplėstas, papildant jį architektūra pagrįstomis veiklomis [9].

### 2.3.2. Persidengimų (Overlap) ciklas

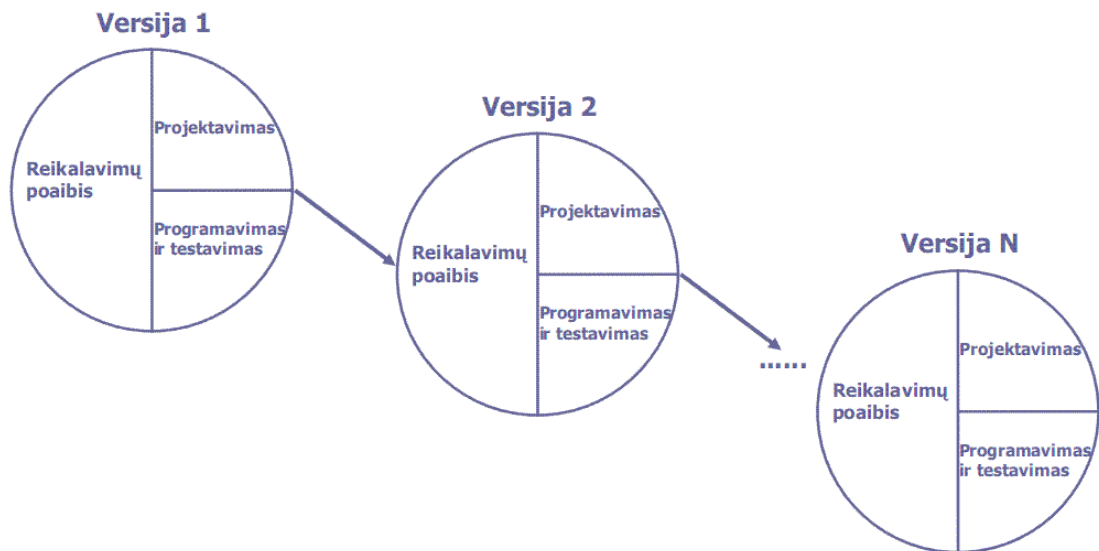


2.3 pav. Persidengimų ciklas

Persidengiantis metodas leidžia fazių persidengimą – bet kuri fazė gali trukti tol, kol nepasibaigė nuo jos priklausančios fazės. Tuo pačiu tai reiškia, kad galime pradėti vykdyti kitų fazių veiklas nepabaigę ankstesnės fazės, jei tik tai veiklų priklausomybės. Tačiau sudėtingi komponentu ir veiklų ryšiai padidina galimybę, kad atsiras nenumatytos iteracijos, dėl ko gali vėluoti projekto pristatymas [11].

### 2.3.3. Evoliucinis ciklas

Tai tiriamasis kūrimas. Tikslas - dirbti su užsakovu nuo pradinių specifikacijų iki galutinės sistemos. Turi būti pradedama nuo aiškių reikalavimų, kurių pagrindu sukuriama pradinė sistemos versija (dažniausiai tik prototipas). Analizė, projektavimas, realizavimas ir testavimas pagal *Overlap* metodo principus ir be jokios formalios dokumentų peržiūros. Pirmoji versija pateikiama vartotojui, kuris įvertina produktą ir apibrėžia tolesnius reikalavimus; tada kuriama antroji versija ir vėl pateikiama vartotojui. Taip kartojama tol, kol gaunamas vartotoją tenkinantis produktas.



**2.4 pav.** Evoliucinis ciklas

Dokumentavimas, produkto įvertinimas, mokymai atliekami tik tada, kai didžioji užsakovo reikalavimų patenkinta. Šis ciklas žymiai greitesnis už kaskadinį ir truputį greitesnis už *Overlap*. Tačiau egzistuoja šie trūkumai:

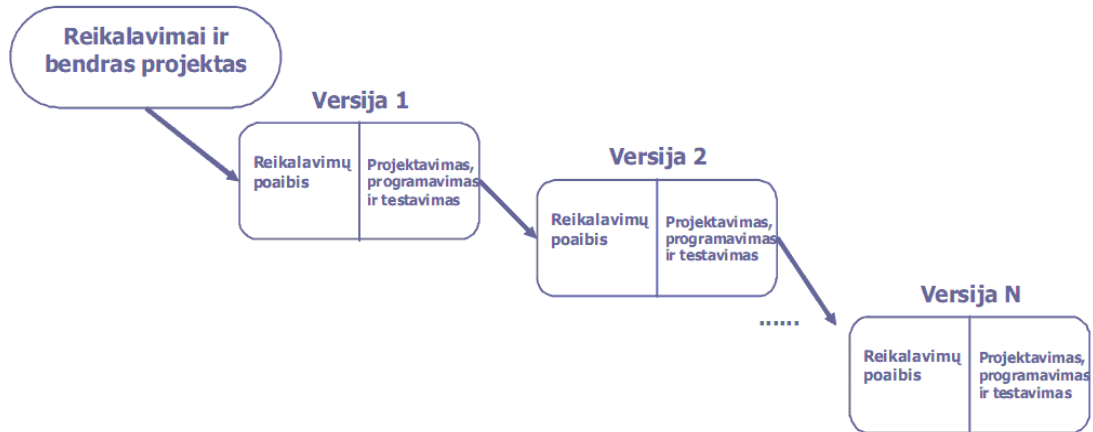
- Trūksta informacijos tarpiniam valdymui.
- Negalima pradžioje numatyti fiksuotos kainos, kadangi neaiški apimtis.
- Sudėtingas palaikymas ir tobulinimas, nes specifikacija ir projektas gali būti prastai dokumentuoti.
- Neskaidrus procesas.
- Dažnai sistemos mažai struktūrizuotos.
- Gali būti reikalingi specialūs įgūdžiai ir žinios ( specialios maketavimo kalbos).

Evoliucinis kūrimas labiausiai tinka mažoms ar vidutinėms sistemoms, didelių sistemų dalims arba trumpaamžėms sistemoms.

Evoliucinis metodas taip pat gali būti labai veiksmingas vykdant testavimo procesą.[12].

#### **2.3.4. Laipsniškas kūrimas**

Jis pradamas žinant visus reikalavimus, tačiau jų apibrėžimas nėra detalus. Reikalavimams prioritetizuojami pagal jų svarbą. Kiekvienas reikalavimų rinkinys sudaro išbaigtą sistemos dalį, kurią galima naudoti.

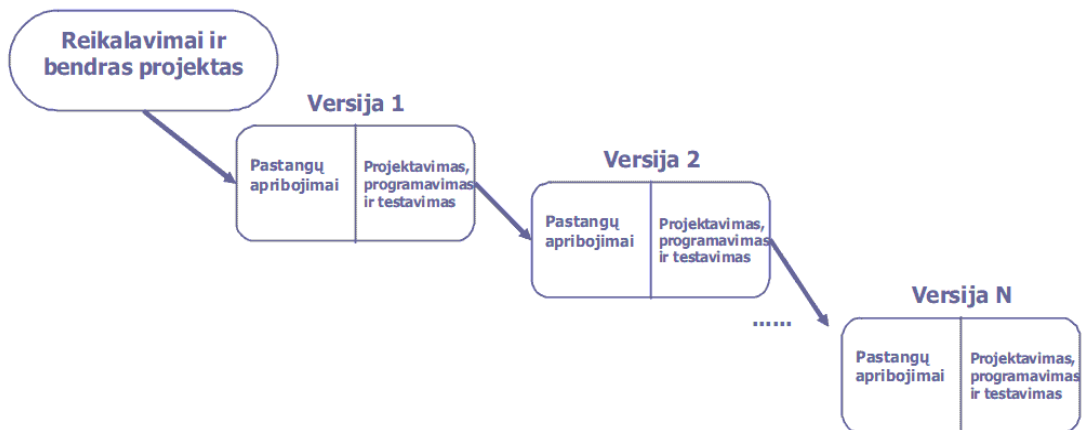


**2.5 pav.** Laipsniškas kūrimas

Laipsniškas kūrimas yra lėtesnis negu evoliucinis. Jis išvengia blogos projekto kokybės, tačiau išlieka kontraktais susiję problemos dėl neaiškių kainų ir projekto apimtys bei prioritetai laikui bėgant keičiasi.

### 2.3.5. Apribojimais grindžiamas (*Bounding Box*) kūrimas

Ciklas panašus į iteracinį, tačiau kiekvienoje iteracijos apribojama ne reikalavimų aibė, bet pastangos arba laikas. Labai dažnai sudaromas kontraktas tam tikram biudžetui.

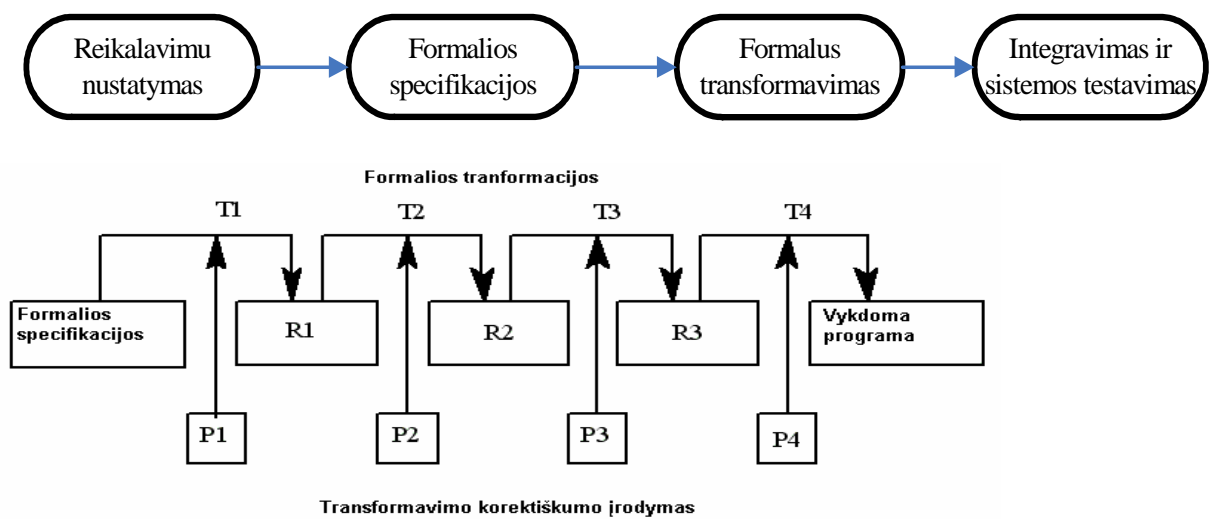


**2.6 pav.** Apribojimais grindžiamas kūrimas

Apribojimais grindžiamas kūrimas neturi problemų dėl kontraktų – klientai dažnai yra linkę sudaryti fiksuotus kontraktus. Tačiau įgyvendinamų reikalavimų aibės gali būti neišbaigtos iki naudojimo tinkamų sistemos dalių.

### 2.3.6. Formalus sistemų kūrimas

Aprašoma matematiškai ir transformuojama į realizuojamą programą. Transformacija išsaugo teisingumą, svarbu yra parodyti, kad programa atitinka specifikaciją. Šis kūrimas įgyvendintas “Cleanroom” programinės įrangos kūrimo modelyje [10].



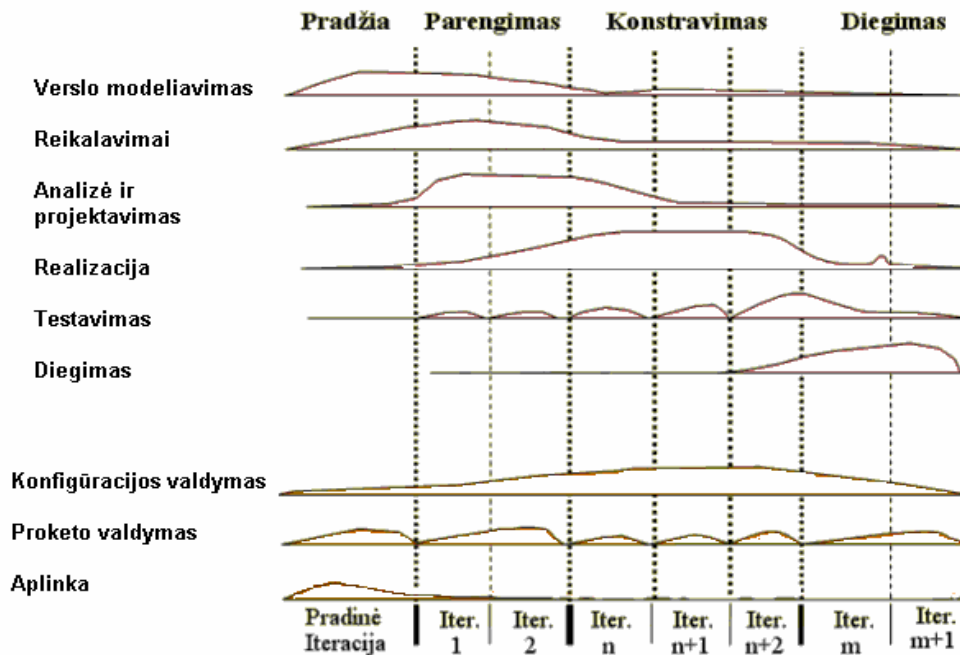
2.7 pav. Formalus sistemų kūrimas

Naudojant formalų sistemų kūrimą, kyla tokios problemos:

- Reikalingos specialios žinios ir pasirengimas.
- Sudėtinga formaliai aprašyti kai kuriuos sistemos aspektus, tokius kaip vartotojo sąsaja.

Formalus sistemų kūrimas labiausiai tinka kritinėms sistemoms, reikalaujančioms saugumo ar apsaugos užtikrinimo.

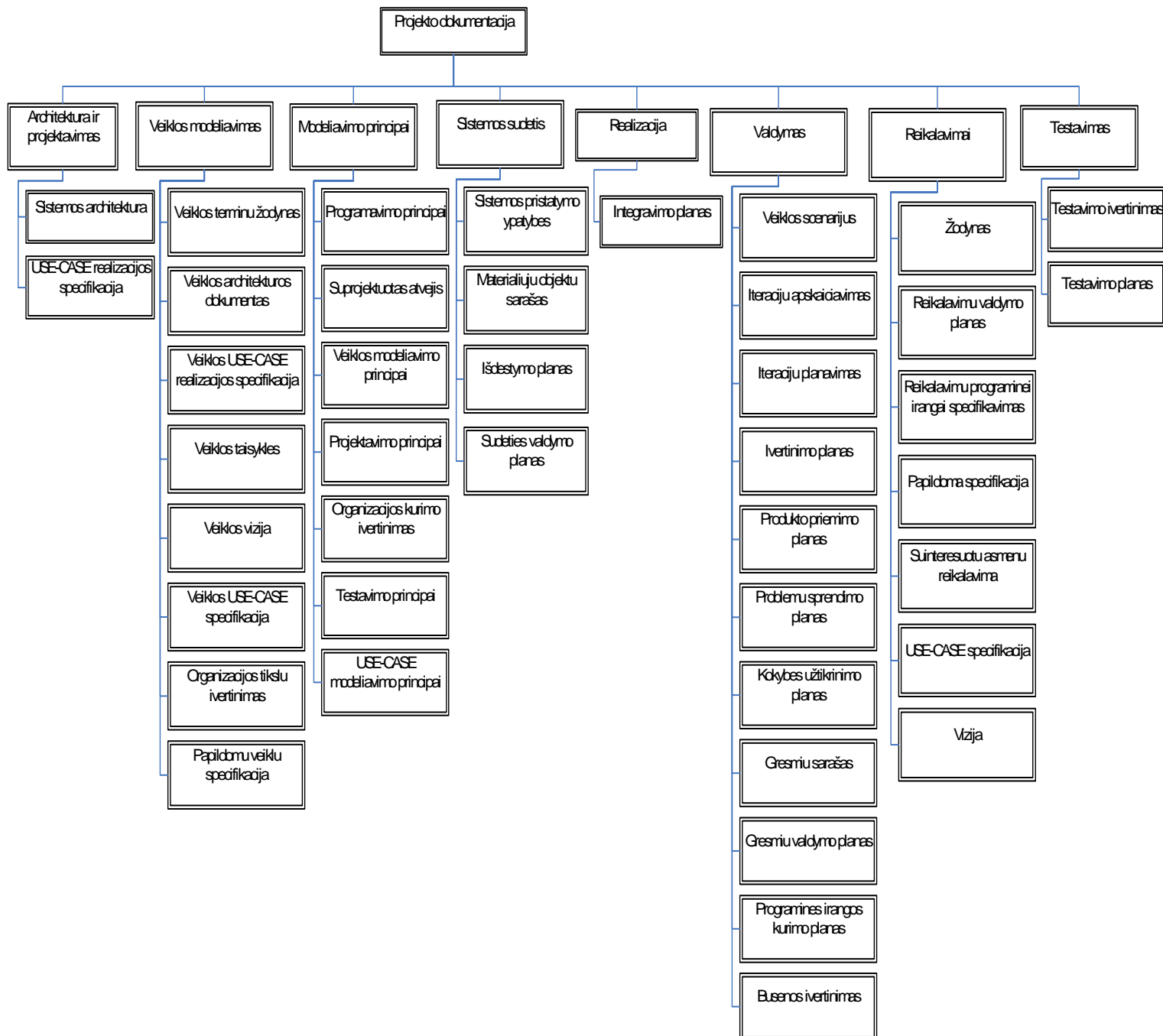
### 2.3.7. Unifikuotas kūrimo procesas (RUP)



2.8 pav. RUP

RUP remiasi persidengimų ir laipsniškuoju kūrimo būdais. Dešimt pagrindinių RUP principų [6]:

1. Sukurti viziją.
2. Valdyti projekto planą.
3. Identifikuoti ir mažinti riziką.
4. Paskirstyti darbus ir sekėti rezultatus.
5. Atlikti ekonominę analizę.
6. Projektuoti komponentinę architektūrą.
7. Palaipsniui konstruoti ir testuoti produktą.
8. Testuoti ir įvertinti rezultatus.
9. Kontroliuoti ir valdyti pakeitimus.
10. Teikti vartotojui paramą.



**2.9 pav.** RUP šablonų klasifikacija

2.9 paveiksle pavaizduota RUP šablonų klasifikacija, kur žemiausias lygmuo atitinka tam tikrą dokumentą. Nepaisant to, kad šie dokumentai apima visus projekto dokumentavimo aspektus, tačiau ne visiems projektams visi šie aspektai reikalingi. Be to, sukurti šablonai yra per daug detalūs, dėl ko dokumentavimo procesas tampa sunkiai valdomas.

EUP (*The Enterprise Unified Process*) yra RUP tęsinys. Procesas kuriamas pradedant nuo įrankio ir įtraukia veiklos architektūros kūrimą į UML pagrįstą PĮ procesą [7].

Dar verta paminėti BEAM (*Basic Enterprise Architecture Methodology*) metodologiją, kurioje modeliavimo procesas panašus į TOGAF 7 ir ADM 8. BEAM yra rinkinys įvairių veiklos architektūrų būdų [8].

### **2.3.8. Lankstieji (*Agile*) kūrimo procesai**

*Agile* kūrimo esmė – paprastumas. Akcentuojamas testavimas (automatiniai testai, testai kuriami prieš programavimą, jo metu ir po jo, radus naujų klaidų, kuriami nauji testai), palaikomi pokyčiai (grįžtamasis ryšys, kintami reikalavimai), nuolatinis integravimas bei komandinis darbas.

*Agile* kūrimo procese individai ir sąveikos svarbiau už įrankius ir procesus. Šiame kūrimo procese dokumentacijos kokybė dažniausiai būna prasta, nes veikianti programa yra daug svarbiau nei išsami dokumentacija. Bendravimas su klientu svarbiau už kontrakto derinimą, o reakcija į pokyčius svarbiau už plano sekimą [1].

### **2.3.9. *Volere* šablonas**

*Volere* šablonas yra reikalavimų konteineris. Analizuojant reikalavimų aprašus ir suskirstant juos pagal tipus, reikalavimų atpažinimas (užfiksavimas) ir įgijimas tampa daug paprastesnis. Šabloną sudaro keturios dalys [5]:

- apribojimai sistemai - tai nėra tiesioginiai reikalavimai, tačiau turi didelę įtaką kuriamai sistemai bei jos teikiamiems veikimo rezultatams;
- funkciniai reikalavimai;
- nefunkciniai reikalavimai;
- projekto išėiga (rezultatai, įvertinimas, problemos) .

### **2.3.10. ISO/IEC 12207 standarte apibrėžti procesai ir dokumentų tipai**

ISO (*the International Organization for Standardization*) ir IEC (*the International Electrotechnical Commission*) sudaro specializuotą sistemą, skirtą pasaulinio masto standartizacijai. Šalių organizacijos, kurios yra ISO arba IEC narės, priklauso techniniams komitetams, kurie kuria ir tobulina tarptautinius standartus. ISO ir IEC techniniai komitetai bendradarbiauja įvairiose srityse [4]. Yra atitinkamas standartas, kaip dokumentuoti testavimo dokumentus [3], tačiau dominantė sritis yra ne vien testavimo dokumentavimas, o viso kūrimo proceso, todėl šiam dokumentui nebuvo skirta itin didelis dėmesys.



Pagrindinė jungtinio techninio komiteto užduotis – paruošti tarptautinius standartus. Šie tarptautiniai standartai gali būti naudojami:

- Organizacijoje – padėti sudaryti norimų procesų aplinką. Šie procesai gali būti palaikomi infrastruktūros metodais, procedūromis, metodikomis, įrankiais ar padedant kvalifikuotam personalui. Organizacija gali panaudoti šia aplinką tam, kad atliktų ir valdytų projektų ir progreso sistemas per jų gyvavimo ciklo etapus. Šiuo atveju tarptautiniai standartais naudojami tam, kad įvertintų atitikimą deklaruotų, nustatytų gyvavimo ciklo procesų rinkinį su jų nuostatomis.
- Projekte – padėti parinkti, struktūrizuoti ir panaudoti nustatyto gyvavimo ciklo procesų elementus tam, kad suteikti produktus ir paslaugas. Šiuo atveju tarptautiniai standartai naudojami įvertinti projekto atitikimą deklaruotai ir sukurtai aplinkai.
- Kaip pagalba tiekėjui ir gavėjui – tobulinti susitarimą, susijusį su procesais ir veiklomis. Susitarimo metu, tarptautiniame standarte parenkamos, suderinamos ir atliekamos veiklos bei procesai. Šiuo atveju tarptautinis standartas naudojamas plėtojant susitarimą.
- Kaip priemonės, naudojamos organizacijų ir vertintojų – atliekant vertinimus, kurie gali būti panaudoti palaikant organizacinių procesų tobulinimą.

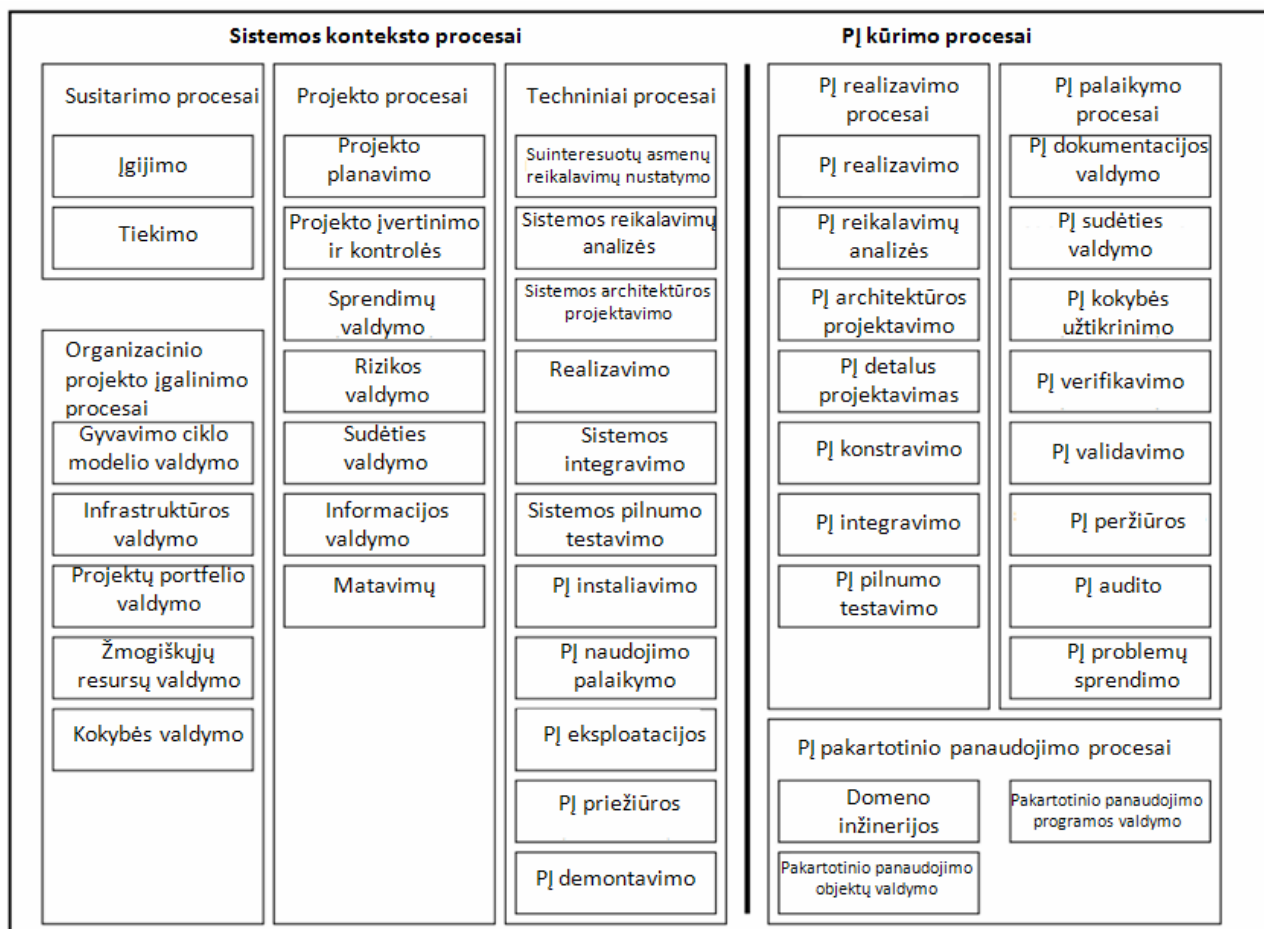
Tarptautiniai standartai nurodo reikalavimus daugeliui procesų, tinkamų naudoti PĮ produkto ar paslaugos gyvavimo ciklo metu. Tačiau paprasti projektai ar organizacijos nebūtinai turi naudoti visus procesus, pateikiamus tarptautiniuose standartuose. Todėl tarptautinių standartų panaudojimas apima procesų, tinkamų organizacijai ar projektui, atranką.

Tarptautinis standartas sugrupuoja veiklas, kurios gali būti atliekamos PĮ sistemos gyvavimo ciklo metu, į 7 procesų grupes [4]:

- 1) Susitarimo procesai – 2 procesai.
- 2) Organizacinio projekto įgalinimo procesai – 5 procesai.
- 3) Projekto procesai – 7 procesai.
- 4) Techniniai procesai – 11 procesų.
- 5) PĮ kūrimo procesai – 7 procesai.

- 6) PĮ palaikymo procesai – 8 procesai.  
 7) PĮ pakartotinio panaudojimo procesai – 3 procesai.

Bus pateikiama kiekvieno proceso numatoma išėja.



2.10 pav. ISO/IEC 12207 standarte pateikiami procesai ir jų klasifikacija

2.3 lentelė. ISO/IEC 12207 standarte analizuojami procesai ir jų išėigos

Procesas	Numatoma išėja	Dokumentuoti aspektai
Įgijimo	<ul style="list-style-type: none"> <li>Nurodoma įgijimo poreikiai, tikslai, produktas ir/arba paslaugų priėmimo kriterijai ir įgijimo strategija;</li> <li>Tobulinamas susitarimas, kuris aiškiai nurodo lūkesčius, įsipareigojimus ir atsakomybes tiek tiekėjo, tiek užsakovo;</li> <li>Parenkamas vienas ar keli tiekėjai;</li> <li>Užsakoma produktas ir/arba paslauga, kuri patenkina užsakovo norus;</li> <li>Stebimas užsakymo vykdymas, kad būtų patenkinti kainos, tvarkaraščio ir kokybės apribojimai;</li> <li>Užsakymo priėmimas.</li> </ul>	<ul style="list-style-type: none"> <li>Konceptas arba poreikiai sukurti naują arba patobulinti egzistuojančią sistemą, PĮ arba paslaugą;</li> <li>Sistemos reikalavimų specifikavimas ir analizė;</li> <li>Informacijos apie tiekėją specifikavimas.</li> </ul>
Tiekimo	<ul style="list-style-type: none"> <li>Identifikuojamas produkto arba paslaugos užsakovas;</li> <li>Atliekamas atsakas į užsakymą;</li> <li>Užsakovas ir tiekėjas susitaria dėl produkto arba</li> </ul>	<ul style="list-style-type: none"> <li>Informacijos apie užsakovą specifikavimas;</li> <li>Produkto gyvavimo ciklo modelis;</li> </ul>

	<p>paslaugos tobulinimo, priežiūros, valdymo, pakavimo, pristatymo ir instaliavimo;</p> <ul style="list-style-type: none"> <li>• Tiekėjas sukuria produktą ar paslaugą, kuri atitinka susitartus reikalavimus;</li> <li>• Pagal nustatytus reikalavimus produktas arba paslauga pristatoma užsakovui;</li> <li>• Pagal nustatytus reikalavimus produktas įdiegiamas pas užsakovą.</li> </ul>	<ul style="list-style-type: none"> <li>• Projekto kūrimo planas;</li> </ul>
Gyvavimo ciklo modelio valdymo	<ul style="list-style-type: none"> <li>• Pateikiama gyvavimo ciklo modelio valdymo ir panaudojimo strategijos bei procedūros;</li> <li>• Nurodoma gyvavimo ciklo valdymo atsakomybė bei galiojimo ribos;</li> <li>• Nurodoma, valdoma ir tobulinama organizacijos naudojami gyvavimo ciklo procesai, modeliai ir procedūros;</li> <li>• Įgyvendinama prioritetizuotų procesų tobulinimai.</li> </ul>	<ul style="list-style-type: none"> <li>• Produkto gyvavimo ciklo modelio valdymo ir naudojimo planas;</li> </ul>
Infrastruktūros valdymo	<ul style="list-style-type: none"> <li>• Nurodomi procesų palaikymo reikalavimai infrastruktūrai;</li> <li>• Nurodomi ir specifikuojami infrastruktūros elementai;</li> <li>• Įgyvendinami infrastruktūros elementai;</li> <li>• Prižiūrima ir tobulinama stabili ir patikima infrastruktūra.</li> </ul>	<ul style="list-style-type: none"> <li>• Infrastruktūra ir jos sudėtis;</li> </ul>
Projektų portfelio valdymo	<ul style="list-style-type: none"> <li>• Apibūdinamos, prioritetizuojamos ir parenkamos veiklos rizikingos galimybės, investicijos ir būtinybės;</li> <li>• Nustatoma ir priskiriama kiekvieno projekto resursai ir biudžetai;</li> <li>• Nustatoma projekto valdymo atsakingos esybės;</li> <li>• Palaikomi projektai, atitinkantys susitarimų ir suinteresuotų asmenų reikalavimus;</li> <li>• Projektai, neatitinkantys susitarimų arba suinteresuotų asmenų reikalavimų, persvarstomi arba sustabdomi.</li> </ul>	<ul style="list-style-type: none"> <li>• Naujos veiklos galimybės;</li> <li>• Kiekvieno projekto atsakomybė;</li> <li>• Numatomos projektų išeigos;</li> <li>• Projektų resursai;</li> <li>• Vykstančių projektų eigos įvertinimas;</li> </ul>
Žmogiškųjų resursų valdymo	<ul style="list-style-type: none"> <li>• Nustatomi projektui reikalingi įgūdžiai;</li> <li>• Suteikiami projektui reikalingi žmogiškieji resursai;</li> <li>• Tobulinami, prižiūrimi arba gerinami personalo įgūdžiai;</li> <li>• Sprendžiami projektų konfliktai dėl žmogiškųjų resursų;</li> <li>• Surenkami, pakartotinai panaudojami, dalinami ir gerinami individualūs įgūdžiai, informacija ir žinios visoje organizacijoje.</li> </ul>	<ul style="list-style-type: none"> <li>• Įgūdžių tobulinimo planas;</li> <li>• Tobulinimo instrukcijos;</li> <li>• Naujo personalo įdarbinimo kriterijai bei planai;</li> </ul>
Kokybės valdymo	<ul style="list-style-type: none"> <li>• Nustatoma organizacijos kokybės valdymo metodikos ir procedūros;</li> <li>• Nustatomi organizacijos kokybės tikslai;</li> <li>• Nustatoma už kokybės valdymą atsakingos esybės;</li> <li>• Stebima kliento patenkinimo būsena;</li> <li>• Imamasi atitinkamų veiksmų, jei kokybės tikslai nepasiekiami.</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• Kokybės valdymo tikslai bei kas atsakingas, kad šie tikslai būtų pasiekti;</li> <li>• Periodiniai projekto kokybės tikrinimo planai;</li> </ul>
Projekto planavimo	<ul style="list-style-type: none"> <li>• Nustatoma projekto darbo apimtis;</li> <li>• Įvertinamas projekto tikslų pasiekimo galimumas,</li> </ul>	

	<p>įvertinant prieinamus resursus bei nustatytus apribojimus;</p> <ul style="list-style-type: none"> <li>• Įvertinamos užduotys ir resursai, reikalingi darbui įvykdyti;</li> <li>• Nustatomi projekto elementų ryšiai su kitais projektais bei organizaciniais vienetais;</li> <li>• Tobulinami projekto įvykdymo planai;</li> <li>• Aktyvuojami projekto įvykdymo planai.</li> </ul>	
Projekto įvertinimo ir kontrolės	<ul style="list-style-type: none"> <li>• Stebimas ir fiksuojamas projekto progresas;</li> <li>• Stebimi projekto elemento ryšiai su kitais projektais bei organizaciniais vienetais;</li> <li>• Imamasi veiksmų, skirtų koreguoti nukrypimus nuo plano bei išvengti pasikartojančių problemų, nustatytų projekte, kai nepasiekiami projekto tikslai;</li> <li>• Pasiekiami ir užfiksuojami projekto tikslai.</li> </ul>	<ul style="list-style-type: none"> <li>• Nukrypimo nuo projekto vykdymo planų priežastys ir jų sprendimo būdai;</li> </ul>
Sprendimų valdymo	<ul style="list-style-type: none"> <li>• Nustatoma sprendimų priėmimo strategija;</li> <li>• Nustatomo alternatyvūs veiksmai;</li> <li>• Pasirenkamas pageidautinas veiksmų planas;</li> <li>• Užfiksuojami ir paskelbiami sprendimai, sprendimų loginiai pagrindimai ir prielaidos.</li> </ul>	<ul style="list-style-type: none"> <li>• Sprendimų priėmimo planas;</li> <li>• Sprendimų pasekmės;</li> </ul>
Rizikų valdymo	<ul style="list-style-type: none"> <li>• Nustatoma rizikų valdymo, kurį reikės atlikti, apimtis;</li> <li>• Nustatomos ir pritaikomos tinkamos rizikų valdymo strategijos;</li> <li>• Nustatomos rizikos, kai jos auga, arba projekto vykdymo metu;</li> <li>• Atliekami atitinkami veiksmai tam, kad pakoreguoti arba išvengti rizikos.</li> </ul>	<ul style="list-style-type: none"> <li>• Numatomas įgyvendinti rizikos valdymo procesas;</li> <li>• Už rizikos valdymą atsakingos esybės;</li> <li>• Rizikos valdymo proceso kontekstas;</li> </ul>
Sudėties valdymo procesas	<ul style="list-style-type: none"> <li>• Nustatoma sudėties valdymo strategija;</li> <li>• Nustatomi objektai, kuriems reikalingas sudėties valdymas;</li> <li>• Nustatomas sudėties pradinė būseną;</li> <li>• Kontroliuojama sudėties valdymo įtaka objektų kaitai;</li> <li>• Kontroliuojama išleistų objektų sudėtis;</li> </ul>	<ul style="list-style-type: none"> <li>• Sudėties valdymo planas;</li> </ul>
Informacijos valdymo	<ul style="list-style-type: none"> <li>• Nustatoma informacija, kurią reikės valdyti;</li> <li>• Nustatomi informacijos vaizdavimo būdai;</li> <li>• Prireikus informacija transformuojama ir perleidžiama;</li> <li>• Užfiksuojama informacijos būseną;</li> <li>• Informacija naujausia, pilna ir galiojanti;</li> <li>• Informacija tampa prieinama pageidaujančioms esybėms.</li> </ul>	<ul style="list-style-type: none"> <li>• Informacijos vienetai, kuriems numatoma taikyti informacijos valdymą;</li> </ul>
Matavimų	<ul style="list-style-type: none"> <li>• Identifikuojama techninių ir valdymo procesų informacijos poreikis;</li> <li>• Identifikuojamo ir/arba plėtojami atitinkami matavimai informacijos poreikiams;</li> <li>• Identifikuojamos ir planuojamos matavimų veiklos;</li> <li>• Surenkama, saugojama ir analizuojama reikalingi duomenys bei interpretuojami rezultatai;</li> <li>• Įvertinamas matavimų procesas bei matavimai.</li> </ul>	<ul style="list-style-type: none"> <li>• Organizacijos charakteristikos, kurioms svarbūs matavimai;</li> <li>• Matavimų rezultatai;</li> </ul>

<p>Suinteresuotų asmenų reikalavimų nustatymo</p>	<ul style="list-style-type: none"> <li>• Nurodoma paslaugų panaudojimo charakteristikos ir kontekstas;</li> <li>• Nurodomi apribojimai sistemos sprendimui;</li> <li>• Užfiksuojamas ryšys tarp suinteresuotų asmenų reikalavimų ir pačių suinteresuotų asmenų ir jų poreikių;</li> <li>• Apibūdinami pagrindai sistemos reikalavimų specifikacijai;</li> <li>• Apibūdinami pagrindai paslaugų validavimo formai;</li> <li>• Pateikiami pagrindai deryboms ir susitarimui dėl paslaugų ar produkto tiekimo.</li> </ul>	<ul style="list-style-type: none"> <li>• Apribojimai sistemos sprendimui;</li> <li>• Numatomų sistemos vartotojų ir kitų suinteresuotų asmenų reikalavimai bei reikalavimų svarbos įvertinimas;</li> </ul>
<p>Sistemos reikalavimų analizės</p>	<ul style="list-style-type: none"> <li>• Sukuriama nustatytas rinkinys funkcinių ir nefunkcinių reikalavimų, apibūdinančių problemą, kurią reikia išspręsti;</li> <li>• Atliekamos metodikos, skirtos optimizuoti norimą projekto sprendimą;</li> <li>• Analizuojama sistemos reikalavimų korektiškumas ir testuojamumas;</li> <li>• Išsiaiškinama sistemos reikalavimų įtaka vykdymo aplinkai;</li> <li>• Prireikus atliekama reikalavimų prioritetizavimas, patvirtinimas ir atnaujinimas;</li> <li>• Užtikrinamas sistemos reikalavimų ir pradinių kliento reikalavimų vientisumas ir atsekamumas;</li> <li>• Įvertinama pokyčio pradinei būsenai įtaka kainai, tvarkaraščiui ir techninėms detalėms;</li> <li>• Visos susijusios esybės supažindinamos su sistemos reikalavimais.</li> </ul>	<ul style="list-style-type: none"> <li>• Funkcinių ir nefunkcinių reikalavimų specifikavimas;</li> </ul>
<p>Sistemos architektūros projektavimo</p>	<ul style="list-style-type: none"> <li>• Nustatomas sistemos architektūros projektas, kuriame identifikuojami visi sistemos elementai ir kuris atitinka nustatytus reikalavimus;</li> <li>• Priskiriami sistemos funkciniai ir nefunkciniai reikalavimai;</li> <li>• Paskirstomi reikalavimai sistemos elementams;</li> <li>• Atliekamas sistemos reikalavimų ir sistemos architektūros verifikavimas;</li> <li>• Reikalavimai, paskirstyti sistemos elementams ir jų sąsajoms, susiejami su užsakovo pradiniais reikalavimais;</li> <li>• Prižiūrimas sistemos reikalavimų ir sistemos architektūros projekto vientisumas ir susietumas;</li> <li>• Užfiksuojami sistemos reikalavimai, sistemos architektūros projektas bei jų tarpusavio ryšys bei pranešama apie tai visoms susijusioms pusėms;</li> <li>• Į sistemos projektą įtraukiami žmogiškieji faktoriai bei ergonominės žinios.</li> </ul>	<ul style="list-style-type: none"> <li>• Sistemos architektūros projektas;</li> <li>• Kiekvieno sistemos elemento vidiniai ir išoriniai ryšiai;</li> </ul>
<p>Sistemos integravimo</p>	<ul style="list-style-type: none"> <li>• Sukuriama sistemos integravimo strategija pagal sisteminių reikalavimų prioritetus;</li> <li>• Sukuriami kriterijai, skirti verifikuoti vientisumą sistemos reikalavimų, paskirstytų sistemos elementams, įtraukiant sąsajas tarp sistemos elementų;</li> </ul>	<ul style="list-style-type: none"> <li>• Kriterijai, kaip nustatyti sistemos atitikimą vartotojo reikmėms;</li> </ul>

	<ul style="list-style-type: none"> <li>• Pagal sukurtus kriterijus verifikuojamas sistemos integravimas;</li> <li>• Užtikrinamas sistemos projekto ir integruotos sistemos elementų vientisumas ir susietumas;</li> <li>• Sukuriamas integruota sistema, kuri atitinka sistemos projektą;</li> <li>• Sukuriama integruota sistema, kurioje matyti, kad egzistuoja baigtinė seka pritaikomų sistemos elementų.</li> </ul>	
Sistemos pilnumo testavimo	<ul style="list-style-type: none"> <li>• Sukuriami kriterijai, skirti įvertinti sistemos reikalavimų išpildymą;</li> <li>• Pagal sukurtus kriterijus testuojama integruota sistema;</li> <li>• Užfiksuojami testų rezultatai;</li> <li>• Užtikrinamas sistemos paruoštumas pristatymui.</li> </ul>	<ul style="list-style-type: none"> <li>• Sistemos atitikimo vartotojo reikmėms testavimo rezultatai;</li> </ul>
PĮ diegimo	<ul style="list-style-type: none"> <li>• Sukuriama PĮ diegimo strategija;</li> <li>• Sukuriami PĮ diegimo kriterijai, skirti įvertinti PĮ diegimo reikalavimų išpildymą;</li> <li>• PĮ produktas instaliuojamas numatytoje aplinkoje;</li> <li>• Užtikrinamas PĮ produkto paruoštumas naudoti numatytoje aplinkoje.</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ diegimo strategija bei jos kriterijai;</li> <li>• Sistemos diegimo instrukcija;</li> </ul>
PĮ naudojimo palaikymo	<ul style="list-style-type: none"> <li>• Produktas pabaigtas ir pristatytas užsakovui;</li> <li>• Palaikomi užsakovo priėmimo testai ir peržiūros;</li> <li>• Produktas paruošiamas naudojimui vartotojo aplinkoje;</li> <li>• Priėmimo metu užfiksuotos problemos ir ieškoma šių problemų sprendimų.</li> </ul>	
PĮ eksploatacijos	<ul style="list-style-type: none"> <li>• Nustatoma eksploataavimo strategija;</li> <li>• Identifikuojama ir įvertinama korektiškos PĮ eksploatacijos sąlygos eksploataavimo aplinkoje;</li> <li>• PĮ testuojama ir nustatoma operuoti eksploataavimo aplinkoje;</li> <li>• PĮ eksploatuojama realiomis sąlygomis;</li> <li>• Pagal susitarimą teikiami apmokymai ir konsultacijos.</li> </ul>	<ul style="list-style-type: none"> <li>• Eksploatavimo planas;</li> </ul>
PĮ priežiūros	<ul style="list-style-type: none"> <li>• Sukuriama priežiūros strategija tam, kad būtų galima valdyti pakeitimus bei produkto srautus pagal išleidimo pobūdį;</li> <li>• Identifikuojama pokyčių įtaka organizacijoje egzistuojančiai sistemai, operacijoms arba sąsajoms;</li> <li>• Prireikus atnaujinama paveiktosios sistemos ar PĮ dokumentacija;</li> <li>• Modifikuoti produktai tobulinami kartu su testais, skirtais pademonstruoti, kad reikalavimai nepažeisti;</li> <li>• Teikiami produkto atnaujinimai klientui;</li> <li>• Apie sistemos pakeitimus pranešama visoms suinteresuotoms pusėms.</li> </ul>	<ul style="list-style-type: none"> <li>• Priežiūros planas;</li> </ul>
PĮ demontavimo	<ul style="list-style-type: none"> <li>• Nustatom PĮ demontavimo strategija;</li> <li>• Pateikiami demontavimo apribojimai kaip reikalavimų įvestys;</li> <li>• Sistemos PĮ elementai sunaikinami arba patalpinami į saugyklą;</li> <li>• Nustatoma darbinės aplinkos sutarta būseną;</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ demontavimo planas;</li> <li>• Reikalavimai demontavimui;</li> </ul>

	<ul style="list-style-type: none"> <li>• Informacija apie demontavimo veiksmus tampa prieinama.</li> </ul>	
PĮ realizavimo	<ul style="list-style-type: none"> <li>• Apibūdinama realizavimo strategija;</li> <li>• Identifikuojami projekto realizavimo technologijos apribojimai;</li> <li>• Realizuojama PĮ elementas;</li> <li>• PĮ elementas supakuojamas ir saugomas pagal tiekimo susitarimą.</li> </ul>	<ul style="list-style-type: none"> <li>• Realizavimo planas;</li> <li>• Reikalavimai projekto realizavimo technologijai;</li> </ul>
PĮ reikalavimų analizės	<ul style="list-style-type: none"> <li>• Nurodomi reikalavimai, priskirti PĮ elementams ir jų sąsajoms;</li> <li>• Analizuojamas PĮ reikalavimų korektiškumas ir testuojamumas;</li> <li>• Išsiaiškinama PĮ reikalavimų įtaka darbinei aplinkai;</li> <li>• Nustatomas PĮ reikalavimų ir sistemos reikalavimų vientisumas ir susietumas;</li> <li>• PĮ reikalavimų įvykdymo prioritizavimas;</li> <li>• Prireikus atnaujinami ir patikrinami PĮ reikalavimai;</li> <li>• Įvertinama pokyčių PĮ reikalavimuose įtaka kainai, tvarkaraščiui ir techniniams aspektams;</li> <li>• PĮ reikalavimai užfiksuojami ir pranešami visoms susijusioms pusėms.</li> </ul>	<ul style="list-style-type: none"> <li>• Vykdytojo reikalavimai kuriamam produktui;</li> </ul>
PĮ architektūros projektavimo	<ul style="list-style-type: none"> <li>• Sukuriamas ir užfiksuojamas PĮ architektūros projektas, kuris apibūdina PĮ elementus, kurie išpildys PĮ reikalavimus;</li> <li>• Nustatomi PĮ elementų vidiniai ir išoriniai ryšiai;</li> <li>• Nustatomas PĮ reikalavimų ir PĮ projekto vientisumas ir susietumas.</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ architektūros projektas; PĮ elementų vidiniai ir išoriniai ryšiai;</li> <li>• Pradinis duomenų bazės projektas;</li> <li>• Preliminarūs testavimo reikalavimai;</li> </ul>
PĮ detalus projektavimas	<ul style="list-style-type: none"> <li>• Nustatomas kiekvieno numatomo kurti PĮ komponento detalus projektas;</li> <li>• Nurodomos kiekvieno PĮ elemento išorinės sąsajos;</li> <li>• Nustatoma detalaus projekto bei reikalavimų ir architektūros projekto vientisumas ir susietumas.</li> </ul>	<ul style="list-style-type: none"> <li>• Kiekvieno numatomo sukurti PĮ komponento detalus projektas;</li> <li>• Kiekvieno komponento vidinių ir išorinių ryšių modelis;</li> <li>• Detalus duomenų bazės modelis;</li> </ul>
PĮ konstravimo	<ul style="list-style-type: none"> <li>• Nustatomi visų PĮ elementų reikalavimų verifikavimo kriterijai;</li> <li>• Sukuriami visi suprojektuoti PĮ elementai;</li> <li>• Nustatoma PĮ elementų, reikalavimų bei projekto vientisumas ir susietumas.</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ elementų reikalavimų verifikavimo kriterijai;</li> </ul>
PĮ integravimo	<ul style="list-style-type: none"> <li>• Sukuriama integravimo strategija PĮ elementams, atitinkančius PĮ projektą bei prioritetizuotus PĮ reikalavimus;</li> <li>• Nustatomi PĮ elementų verifikavimo kriterijai, užtikrinantys atitikimą PĮ reikalavimams, priskirtiems šioms elementams;</li> <li>• PĮ elementai verifikuojami pagal šiuos kriterijus;</li> <li>• Sukuriami PĮ elementai su nustatyta integravimo strategija;</li> <li>• Užfiksuojami integravimo testavimo rezultatai;</li> </ul>	<ul style="list-style-type: none"> <li>• Integravimo planas;</li> <li>• Regresijos planas;</li> </ul>

	<ul style="list-style-type: none"> <li>• Nustatomas PĮ projekto ir PĮ elementų vientisumas ir susietumas;</li> <li>• Sukuriama ir pritaikoma regresijos strategija pakartotiniam PĮ elementų verifikavimui, kai atsiranda pokyčių PĮ elementuose (įskaitant susijusius reikalavimus, pakitimus projekte ar programos kode).</li> </ul>	
PĮ pilnumo testavimo	<ul style="list-style-type: none"> <li>• Nustatomi suintegruotos PĮ kriterijai, nurodantys atitikimą PĮ reikalavimams;</li> <li>• Integruota sistema verifikuojama pagal nustatytus kriterijus;</li> <li>• Užfiksuoja testų rezultatai;</li> <li>• Sukuriama pakartotinio testavimo regresijos strategija, kai atsiranda pokyčių PĮ elementuose.</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ pilnumo testavimo rezultatai;</li> </ul>
PĮ dokumentacijos valdymo	<ul style="list-style-type: none"> <li>• Sukuriama strategija, identifikuojanti numatomą sukurti dokumentaciją PĮ produkto arba paslaugos gyvavimo ciklo metu;</li> <li>• Identifikuojami standartai, kuriuos numatoma pritaikyti kuriant PĮ dokumentaciją;</li> <li>• Identifikuojama dokumentacija, kurią turėtų sukurti procesas arba projektas;</li> <li>• Specifikuojama, peržiūrima ir patvirtinama visos dokumentacijos turinys ir tikslas;</li> <li>• Dokumentacija sukuriama ir tampa prieinama pagal visus identifikuotus standartus;</li> <li>• Dokumentacija palaikoma pagal nustatytus kriterijus.</li> </ul>	<ul style="list-style-type: none"> <li>• Dokumentavimo planas;</li> <li>• Kiekvieno identifikuoto dokumento projektas;</li> </ul>
PĮ sudėties valdymo	<ul style="list-style-type: none"> <li>• Nustatoma PĮ sudėties valdymo strategija;</li> <li>• Identifikuojami, apibūdinami ir užfiksuoja projekto arba proceso sugeneruoti elementai;</li> <li>• Kontroliuojama elementu modifikacijos ir pristatymai;</li> <li>• Modifikacijos ir pristatymai tampa prieinami visoms susijusioms pusėms;</li> <li>• Elementų ir modifikacijų būsenos užfiksavimas ir paskelbimas;</li> <li>• Užtikrinamas elementų vientisumas ir pilnumas;</li> <li>• Kontroliuojamas elementų saugojimas ir pristatymas.</li> </ul>	<ul style="list-style-type: none"> <li>• PĮ sudėties valdymo planas;</li> </ul>
PĮ kokybės užtikrinimo	<ul style="list-style-type: none"> <li>• Nustatoma kokybės užtikrinimo strategija;</li> <li>• Sukuriami ir palaikomi PĮ kokybės užtikrinimo įrodymai;</li> <li>• Identifikuojama ir užfiksuoja problemos ir/arba neatitikimai reikalavimams</li> <li>• Verifikuojamas produktų, procesų ir veiklų griežtas standartų, procedūrų ir reikalavimų laikymasis.</li> </ul>	<ul style="list-style-type: none"> <li>• Kokybės užtikrinimo planas;</li> </ul>
PĮ verifikavimo	<ul style="list-style-type: none"> <li>• Nustatoma ir įgyvendinama verifikavimo strategija;</li> <li>• Identifikuojami visų reikalingų PĮ darbo produktų verifikavimo kriterijai;</li> <li>• Atliekamos reikalingos verifikavimo veiklos;</li> <li>• Identifikuojami ir užfiksuoja defektai;</li> <li>• Veiklų verifikavimo rezultatai pateikiami</li> </ul>	<ul style="list-style-type: none"> <li>• Verifikavimo planas;</li> <li>• Verifikavimo kriterijai;</li> </ul>



	užsakovui ir kitoms suinteresuotoms pusėms.	
PĮ validavimo	<ul style="list-style-type: none"> <li>Nustatoma ir įgyvendinama validavimo strategija;</li> <li>Identifikuojami visų reikalingų PĮ darbo produktų validavimo kriterijai;</li> <li>Atliekamos reikalingos validavimo veiklos;</li> <li>Identifikuojamos ir užfiksuojamos problemos;</li> <li>Pateikiami įrodymai, kad sukurti PĮ darbo produktai yra tinkami naudojimui;</li> <li>Veiklų validavimo rezultatai pateikiami užsakovui ir kitoms suinteresuotoms pusėms.</li> </ul>	<ul style="list-style-type: none"> <li>Validavimo strategija;</li> <li>Validavimo kriterijai;</li> </ul>
PĮ peržiūros	<ul style="list-style-type: none"> <li>Atliekamos valdymo ir techninės peržiūros, pagrįstos projekto poreikiais;</li> <li>Proceso veiklų būsenos ir produktai įvertinami atliekant peržiūros veiksmus;</li> <li>Pateikiami peržiūros rezultatai visos suinteresuotoms pusėms;</li> <li>Identifikuojamos ir užfiksuojamos rizikos bei problemos.</li> </ul>	<ul style="list-style-type: none"> <li>PĮ peržiūros planas</li> </ul>
PĮ audito	<ul style="list-style-type: none"> <li>Nustatoma ir pritaiko audito strategija;</li> <li>Pagal audito strategiją nustatoma parinktų PĮ produktų ir/arba paslaugų ar procesų atitikimas reikalavimams, planams ir susitarimams;</li> <li>Atitinkama nešališka pusė atlieka auditą;</li> <li>Ieškoma būdų audito metu aptiktoms problemoms išspręsti.</li> </ul>	<ul style="list-style-type: none"> <li>Audito vykdymo planas;</li> </ul>
PĮ problemų sprendimo	<ul style="list-style-type: none"> <li>Nustatoma problemų sprendimo strategija;</li> <li>Problemos identifikuojamos, užfiksuojamos ir klasifikuojamos;</li> <li>Problemos analizuojamos ir identifikuojamo priimtini sprendimai;</li> <li>Įvykdomas problemų sprendimas;</li> </ul>	<ul style="list-style-type: none"> <li>PĮ problemų sprendimo planas;</li> </ul>
Domeno inžinerijos	<ul style="list-style-type: none"> <li>Parenkama domeno modelių ir domeno architektūrų atvaizdavimo būdai;</li> <li>Nustatomas domeno modelis, kuris užfiksuoja esminius bendrus ir skirtingus domeno ypatumus, galimumus, principus ir funkcijas;</li> <li>Nustatoma domeno architektūra, apibūdinanti domeno sistemų grupes, apimant jų bendrumus ir skirtumus;</li> <li>Specifikuojamos esminės domeno ypatybės;</li> <li>Šios domeno ypatybės sužinomos arba sukuriamos ir palaikomos per visą jų gyvavimo ciklą;</li> <li>Domeno modeliai ir architektūros palaikomos per visą jų gyvavimo ciklą.</li> </ul>	<ul style="list-style-type: none"> <li>Domeno modelis;</li> <li>Domeno architektūra;</li> <li>Domeno inžinerijos planas;</li> <li>Domeno terminų žodynas;</li> </ul>
Pakartotinio panaudojimo objektų valdymo	<ul style="list-style-type: none"> <li>Dokumentuojama objektų valdymo strategija;</li> <li>Nustatomo objektų priimtimumo, sertifikavimo ir netinkamumo naudoti kriterijai;</li> <li>Vykdomas objektų saugojimo ir pakartotinio panaudojimo mechanizmas;</li> <li>Užfiksuojama objektų teikiama nauda;</li> <li>Kontroliuojami objektų pokyčiai;</li> <li>Objektų naudotojai informuojami apie problemas,</li> </ul>	<ul style="list-style-type: none"> <li>Pakartotini panaudojimo objektų valdymo planas;</li> <li>Pakartotinio panaudojimo apribojimai;</li> </ul>

	pakeitimus, naujas versijas ar objektų pašalinimą iš saugyklų.	
Pakartotinio panaudojimo programų valdymo	<ul style="list-style-type: none"> <li>• Nustatoma organizacijos pakartotinio panaudojimo strategija, apimant jos tikslus, ir apimtis;</li> <li>• Identifikuojami galimi pakartotinio panaudojimo domenai;</li> <li>• Nustatoma organizacijos gebėjimai susistemintam pakartotiniam panaudojimui;</li> <li>• Įvertinami panaudojimo pasiūlymai tam, kad užtikrinti, jog pakartotinai panaudotas produktas tinkamas pasiūlytai sričiai;</li> <li>• Įgyvendinama organizacijos pakartotinio panaudojimo strategija;</li> <li>• Sukuriami mechanizmai atsiliepimams pateikti, bendradarbiauti suinteresuotoms pusėms;</li> <li>• Stebima ir įvertinama pakartotinai panaudojama programa.</li> </ul>	<ul style="list-style-type: none"> <li>• Organizacijos pakartotinio panaudojimo plano sudarymas;</li> </ul>

### 2.3.11. Kūrimo procesų dokumentavimo metodų palyginimas

*RUP* – nepaisant to, kad, naudojant *RUP* projektų kūrimui, galima panaudoti sukurtus šablonus projektų dokumentavimui, tačiau, nors šie šablonai ir apima visą projekto kūrimo ciklą, tačiau jie orientuoti ne į visą projektą, o į atskiras jo dalis, todėl, norint sėkmingai panaudoti šiuos šablonus ruošiant dokumentaciją, tektų atlikti nemažai patobulinimų, modifikacijų, apibendrinimų tam, kad būtų gautos naudotinos priemonės dokumentacijos ruošimui.

*Agile* dokumentacija yra nepakankama. Dokumentacija yra svarbi *Agile* projektų kūrimo strategijos dalis, tačiau, priešingai nei tradicinių projektų kūrimo strategijų šalininkai, kurie dokumentaciją supranta kaip rizikos mažinimo priemonę, *Agile* strategijoje dokumentacija paprastai yra veiksnys, padidinantis viso projekto riziką, ir todėl dokumentaciją stengiamasi paruošti kaip įmanoma greičiau, dėl to dokumentacija nebūna pilna ir išsami.

ISO/IEC 12207 standarte pateikiami bendri nurodymai, kokius procesus, jų aspektus, dokumentuoti, taip pat galimus kiekvieno proceso dokumentavimo artefaktus, tačiau išsamesni nurodymai kiekvieno artefakto sudėčiai, nekalbant apie kokius nors dokumentų šablonus, nepateikiami.

Tradicinėse projektų kūrimo strategijose, tokiose kaip kaskadinis, persidengimų, evoliucinis, laipsniškas ir kt., dokumentavimas laikomas viena iš projektų kūrimo proceso sudėtinių dalių. Nors pačias strategijas galima pritaikyti dokumentacijos ruošimui, tačiau tai tebus gairės arba idėjos apie patį dokumentacijos ruošimo procesą, ne apie jos sudėtinę dalis ar reikalavimus galutiniam produktui – dokumentui.

## 2.4. Galimų įgyvendinimo priemonių variantų analizė

Tam, kad būtų pilnai išnaudota dokumentavimo sistemos teikiama nauda, dokumentacijos ruošimui numatyta panaudoti priemones, teikiančias galimybę generuoti dokumentus.

MICROSOFT OFFICE VISIO FOR ENTERPRISE ARCHITECTS - nors įrankyje yra realizuotos dokumentacijos generavimo priemonės, tačiau pati dokumentacija yra žemo lygio su minimaliomis galimybėmis modifikuoti realizuotus dokumentų šablonus. Be to, įrankyje nėra galimybės saugoti visos projekto struktūros vienoje byloje – skirtingos projekto diagramos turėtų būti saugomos atskirose bylose, todėl ir dokumentacija būtų generuojama kiekvienai diagramai atskirai, todėl galiausiai dokumentacija būtų apjungiamą iš atskirų dalių, dėl ko dokumentacijos ruošimas nebūtų našus.

VISUAL PARADIGM FOR UML 8.1– dokumentaciją galima kurti dviem būdais – arba redaguojant šablono bylą XML formatu, arba Visual paradigm įrankio “*drag and drop*” priemonės, kai tiesiai iš projektavimo įrankio norimi dokumentuoti elementai įdedami į dokumento šablona. Visual paradigm yra galingas įrankis projektuoti ir generuoti dokumentacijai su itin patogią vartotojo sąsaja, kurios pagalba galima sudaryti įvairius dokumentus, tačiau, norint sudaryti įvairiapusiškesnes ataskaitas, reikia turėti patirties dirbant su XML dokumentais.

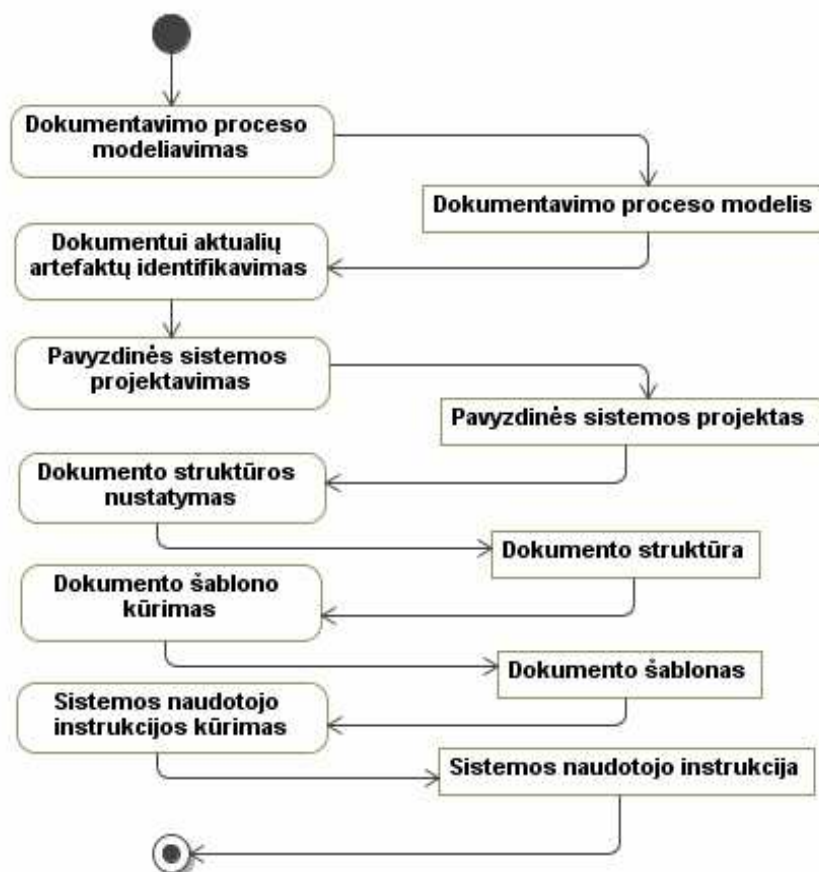
ALTOVA UMODEL 2011 įrankyje, kaip ir Visual Paradigma, vartotojo sąsajos pagalba galima sudaryti norimą dokumento struktūrą. Nors abiejų šių įrankių dokumentų sudarymo idėja panaši, tačiau Visual Paradigma vartotojo sąsaja žymiai funkcionalesnė, turinti daugiau funkcijų, be to, savo dizainu žymiai patrauklesnė nei Altova UModel įrankyje.

MAGICDRAW UML 16.8 įrankyje ataskaitų generavimą atliekamas ataskaitų vedlio (*Report Wizard*) pagalba. Nors *MagicDraw UML* įrankyje nėra realizuota vartotojo sąsaja dokumentams sudaryti, tačiau dokumentų šablonams sukurti naudojama VTL kalba galima sudaryti individualesnius šablonus, nes, nors vartotojo sąsaja ir supaprastina dokumentų sudarymą, tačiau vartotojo veiksmai, susiję su dokumentų personalizavimu, lieka apriboti įrankio kūrėjo. Dėl to, kad, kuriant Projektų dokumentavimo sistemą, reikės kurti specifinius dokumentus, kuriems sudaryti gali nepakakti išnagrinėtų įrankių, tokių kaip Visual Paradigma ar Altova

UModel, vartotojo sąsajos teikiamų galimybių, todėl Projektų dokumentavimo sistemą nutarta realizuoti būtent *MagicDraw* įrankiu.

## 2.5. Dokumentavimo priemonių kūrimo formuluotė

Siekiamos sistemos apibrėžimas iliustruojamas 2.11 paveiksle.



2.11 pav. Siekiamos sistemos apibrėžimas

2.11 paveiksle matomos sistemos kūrimo pagrindinės veiklos ir objektai, kurie gaunami kaip atitinkamų veiklų išeišos. Todėl, sukūrus sistemą, vartotojui bus pateikiama dokumentavimo proceso modelis, pavyzdinės sistemos projektas, pavyzdinės sistemos dokumentacijos struktūra bei dokumento šablonas ir sistemos naudotojo instrukcija.

Tolesnio tyrimo tikslas – sudaryti IS projektų dokumentavimo ir jos realizavimo CASE įrankiu metodiką ir pritaikyti ją UML CASE įrankiui „Magic Draw“.

Sukūrus tinkamą KIS projektų dokumentavimo ir realizavimo metodiką, pats dokumentavimo procesas vyktų sparčiau ir efektyviau, kas įtakotų kuriamos dokumentacijos kokybės augimą. Be to, pasinaudodami sukurtąja metodika, projektuotojai sugebėtų pasiruošti sau patogias dokumentavimo priemones CASE įrankiu.

## **2.6. Nefunkciniai reikalavimai ir apribojimai**

### **2.6.1. Reikalavimai standartams**

Esamu momentu galima surasti ne vieną dokumentacijos ruošimo ir projektavimo standartą, pvz.: RUP, ISO/IEC 12207, *Agile* procesų standartai, todėl nebus stengiamasi sukurti visiškai naują standartą, o, išanalizavus esamus standartus, suformuoti bendrus nurodymus dokumentacijos kūrimui.

Sistemos pritaikymo standartai priklausys nuo naudojamo įrankio – vienuose įrankiuose dokumentacijos generavimui panaudotas XML, kituose pseudo programavimo kalbos. Kaip ir nurodymų dokumentacijai kūrimui, taip ir praktiniam sistemos pritaikymui standartai nebus kuriami, o panaudojami priklausomai nuo pasirinkto projektavimo įrankio.

### **2.6.2. Reikalavimai veikimui**

Viena iš Projektų dokumentavimo sistemos kūrimo priežasčių – nepakankamai efektyvus dokumentavimo proceso vykdymas. Tam, kad padidėtų našumas, pageidautina, kad, sukūrus sistemos projektą, būtų sugeneruota pilna projekto dokumentacija.

Tam, kad būtų galima pakartotinai panaudoti dokumento ruošimo priemones, pageidautina, kad būtų sukurti projekto dokumentacijos parametrai, kurių reikšmės būtų galima koreguoti tiesiai iš projektavimo įrankio.

### **2.6.3. Reikalavimai sąveikai su kitomis sistemomis**

Pageidautina, kad dokumentacija, paruošta pagal Projektų dokumentavimo sistemos nurodymus, būtų pateikiama tokiu formatu, kuris suprantamas pagrindinio teksto redaktoriaus – MS Office Word.

### **2.6.4. Reikalavimai panaudojamumui**

Kadangi asmenys, atsakingi už dokumentacijos ruošimą, nebūtinai turi būti patyrę IT specialistai, todėl pageidautina, kad Projektų dokumentacijos sistema būtų pritaikoma kuo paprasčiau ir nurodymai, kaip naudotis sistema, turi būti suformuluoti kuo išsamiau ir aiškiau.

Projektų dokumentavimo sistemos praktinį pritaikymą nuspręsta realizuoti *MagicDrawUML* įrankio 16.8 versija. Pastebėta, jog įrankio versijos išleidžiamos labai dažnai. Kyla rizika, jog, praktinį Projektų dokumentavimo sistemos realizavus su 16.8 versija, kils sunkumų, kai sistemos funkcionalumą reikės pritaikyti naujesnei *MagicDraw UML* versijai. Tačiau mažai tikėtina, kad *MagicDraw UML* tiekėjai taip pakeistų įrankio architektūrą, kad nebūtų įmanoma dirbti su projektais, sukurtais senesnėmis įrankio versijomis, todėl šio rizikos faktoriaus įtaka taip pat neturi žymios įtakos viso Projektų dokumentavimo sistemos kūrimo sėkmei.

## 2.7. Dokumentavimo proceso ir dokumentacijos kokybės kriterijai

Dokumentavimo procesui ir dokumentams iškelti kokybės reikalavimai matomi 2.4 lentelėje.

**2.4 lentelė.** Dokumentavimo proceso ir dokumentacijos kokybės reikalavimai

Kokybės reikalavimas	Paaiškinimas
Išsamumas	Dokumentacija turi išsamiai atspindėti atitinkamo etapo rezultatus.
Nepertekliškumas	Dokumentacija turi atspindėti tik tuos rezultatus, kurie reikalingi atitinkam etape.
Funkcionalumas	Dokumentacija turi ne tik aprašyti projekto artefaktus, bet ir padėti juos įvertinti ir priimti sprendimus, pateikti projekto patikrinimo, apimties įvertinimo duomenis.
Savalaikiškumas	Dokumentacija turi būti pateikiama pagal pareikalavimą.
Patogumas vartotojui	Dokumentacija pateikianti priemonė turi būti tokia, kokios tikisi vartotojas.
Lankstumas, pritaikomumas	Priemonės, pateikiančios dokumentaciją, turi būti nesunkiai pritaikomos pakitusiems vartotojo poreikiams.
Išmokstamumas	Vartotojas turi sugebėti ne tik naudotis, bet ir, prireikus, modifikuoti dokumentaciją pateikiančias priemones.

## **2.8. Analizės išvados**

1. Atlikta projektavimo procesų ir įrankių analizė parodė, kad dokumentavimo procesas nėra pakankamai reglamentuotas, nėra vieningos strategijos dokumentams rengti, todėl dokumentacija dažnai ruošama neefektyviai, paskubomis, dėl ko nukenčia jos kokybė.

2. Nors esami dokumentacijos generavimo įrankiai suteikia galimybę generuoti dokumentaciją, pateikiamos naudotojo instrukcijos, tačiau trūksta metodikos dokumentacijos generavimo ir šablonų taikymo integravimui į bendrą projekto kūrimo procesą; neišnaudojamos įrankių galimybės, kurias būtų galima įtraukti į ataskaitų rengimą.

3. Suformulavus nurodymus dokumentacijos rengimo procesui, integravus dokumentacijos generavimą bei realizavus papildomą funkcionalumą, dokumentaciją būtų galima ruošti efektyviau.

4. Ataskaitų kūrimo įrankius būtų galima pagerinti sukuriant papildomas funkcijas:

- Projektavimo ir testavimo darbų apimties skaičiavimą;
- Skirtingų stadijų projekto elementų ryšių pilnumo nustatymą.

### 3. Projekto dokumentų kūrimo metodika

Nepaisant to, kad dauguma sistemų projektavimo metodų dokumentavimą laiko pagalbine veikla ir rekomenduoja skirti kuo mažiau projektui skirtu laiko dokumentacijos ruošimui, tačiau kokybiškos dokumentacijos teikiamą naudą sunku paneigti.

Mažai tikėtina, kad iškart pavyks sukurti sistemą be trūkumų, tačiau tie trūkumai gali išryškėti po savaitės, mėnesio ar dar vėliau. Kuo vėliau išryškėja trūkumas, tuo sudėtingiau rasti būdą, kodėl tas trūkumas galėjo atsirasti bei kaip jį panaikinti. Kokybiška dokumentacija gali būti išsigelbėjimas.

Kuriamo sprendimo – projektų dokumentavimo sistemos – dalykinė sritis priklauso nuo dokumentuojamo projekto – tai gali būti nuo studento magistrinio darbo iki įmonės vykdomo projekto.

Ištyrus pagrindines esamas projektų kūrimo metodikas, galima išskirti tokius pagrindinius projektų kūrimo etapus:

1. Projekto apibrėžimas – labai priklauso nuo dalykinės srities, todėl sudėtinga nustatyti kokius nors vieningus reikalavimus ar taisykles šio etapo dokumentavimui.
2. Reikalavimų specifikavimas – patys reikalavimai užrašomi arba laisva forma, arba panaudojant egzistuojančius šablonus, tokius kaip *Volere*. Šiame etape braižomos panaudojimo atvejų diagrama su detalio panaudojimo atvejų specifikacija, sekų, veiklos diagramos svarbios kuriamai sistemai – šias diagramas numatoma įtraukti į kuriamą sistemą, sukuriant joms šablonus.
3. Analizė ir projektavimas – vienas iš svarbiausių etapų kuriamai projektų dokumentavimo sistemai, nes būtent čia kyla didžiausias poreikis ataskaitų šablonams. Reikalingi elementai:
  - Duomenų bazės modelis su duomenų bazės lentelių bei jų atributų aprašais. Čia taip pat galima įtraukti esybių bei klasių diagramas.
  - Klasių diagrama.
  - Dokumentuojamos sistemos loginis modelis.



4. Realizavimas – maža tikimybė, kad realizavimo metu atsiras poreikis naujoms ataskaitoms. Šio etapo vykdymo metu vyksta egzistuojančių ataskaitų naudojimas, ir, prireikus, modifikavimas, pritaikant jas prie konkrečių realizavimo atvejų.
5. Testavimas – kaip ir realizavimo etapas, yra technologinio pobūdžio. Testavimo vykdymo metu kuriamam dokumentui – testavimo planui – taip pat sudėtinga nustatyti konkrečius apibendrintus reikalavimus, nes jie labai priklauso nuo konkrečios sistemos funkcionalumo.
6. Diegimas – iki diegimo etapo pradžios turi būti paruošta sistemos naudotojo instrukcija, sistemos administravimo instrukcija bei diegimo planas. Šie dokumentai naudojami diegimo etape. Tačiau šiems dokumentams ruošti ataskaitų šablonus, paremtus *MagicDraw UML*, taip pat nėra prasminga, nes nėra poreikio į šiuos dokumentus įtraukti kokias nors UML diagramas.

Po kiekvieno iš šių etapų galima užfiksuoti kontrolės tašką, kuriame turi būti suformuotas realus rezultatas. Kadangi kuriama dokumentavimo metodika, todėl aktualiausias rezultatas – dokumentas. Tačiau ne visiems kūrimo etapams dokumentacija yra svarbi – kai kurie iš jų yra grynai technologiniai, tokie kaip realizavimas, kurie dokumentuojami arba labai mažai, arba iš vis nedokumentuojami, todėl kuriamai dokumentavimo sistemai šie etapai nėra aktualūs.

### **3.1. Dokumentavimo procesas**

3.1 paveiksle vaizduojama bendrą dokumentų kūrimo schemą galima suprasti kaip grandinėlių procesą, kuriuos reikia atlikti, norint sukurti sistemos dokumentaciją.

Visų pirma, reikia išsiaiškinti patį projektavimo procesą bei jo etapus (dėl šios priežasties analizės dalyje buvo tiriami standartiniai projektavimo procesai), kad būtų galima suprasti, kokios srities dokumentų reikia, norint dokumentuoti kuriamą sistemą. Išskirti kontroliniai taškai galėtų būti tie taškai, kuriuose atsiranda poreikis dokumentavimui.



**3.1 pav.** Bendra metodikos kūrimo schema

Apibrėžiant kontrolinių taškų artefaktus, išskiriami galimi ataskaitų tipai bei jų prioritetai, nustatant svarbiausius artefaktus (nes nebūtinai visos ataskaitos yra vienodai reikalingos). Galimi ataskaitų tipai: „Veiklos analizės stadijos ataskaita“, „Reikalavimų specifikavimo ataskaita“, „Architektūros projekto ataskaita“, „Detalaus komponento projekto ataskaita“, „DB schemos projekto ataskaita“. Atskiru atveju „Magistro darbo projekto ataskaita“.

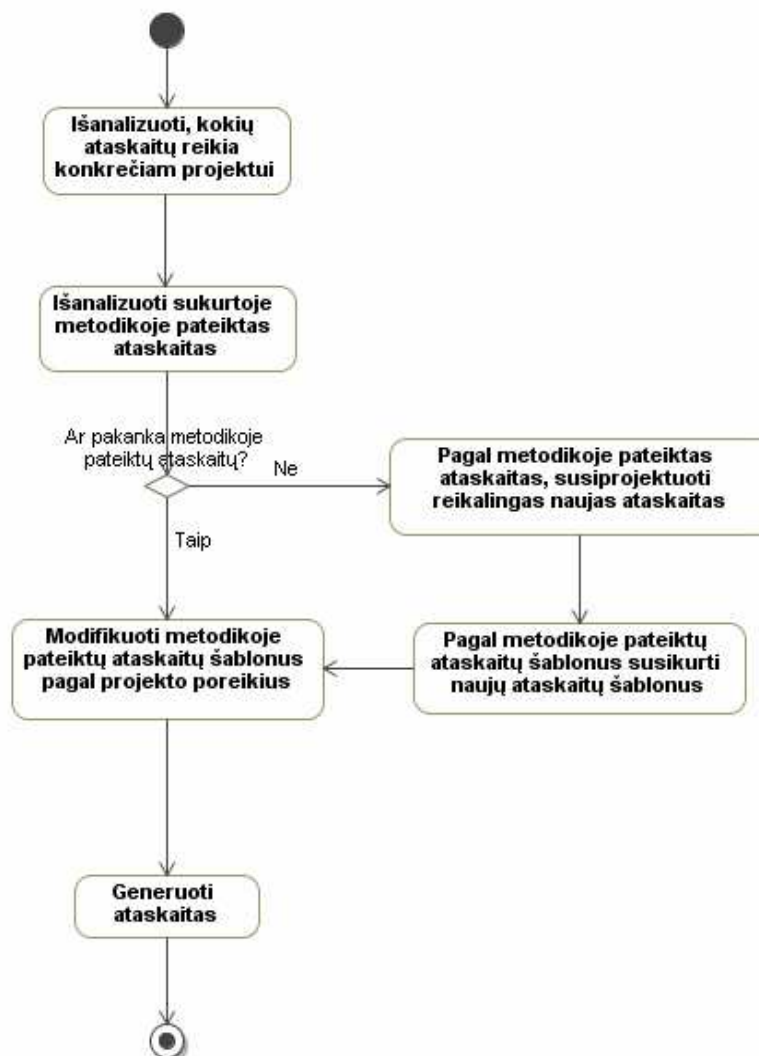
Vėliau suformuojama kontrolinio taško ataskaitos eskizas bei struktūra. Turint šią informaciją, galima formuoti ataskaitos šabloną, naudojant VTL (*Velocity Template Language*) kalbą – kalbą, kuria kuriama ataskaitų šablonai generavimui

*MagicDraw UML* įrankiu. Sukūrus ataskaitos šabloną, atliekamas ataskaitos generavimas *MagicDraw UML* įrankiu.

### 3.2. Projektų dokumentavimo metodikos taikymas

3.2 paveiksle pateikiama metodikos panaudojimo schema – t.y. ką turi atlikti vartotojas, norėdamas pasinaudoti metodikos teikiama nauda.

Visų pirma reikia, kad vartotojas gerai suprastų projektą, prie kurio dirbama – t.y. sugebėtų suprasti projekto reikmes dokumentacijai. Tik taip bus įmanoma išskirti projektui reikalingas ataskaitas. Išskyrus projektui reikalingas ataskaitas, galima analizuoti metodikoje pateiktas ataskaitas ir mėginti jas pritaikyti konkrečiam projektui. Kadangi mažai tikėtina, kad metodikoje bus pateikti visi galimi ataskaitų pavyzdžiai ir šablonai, todėl gali tekti pagal metodikoje pateiktus nurodymus susikurti naujas ataskaitas bei jų šablonus. Paskutiniame etape atliekamas ataskaitų generavimas.



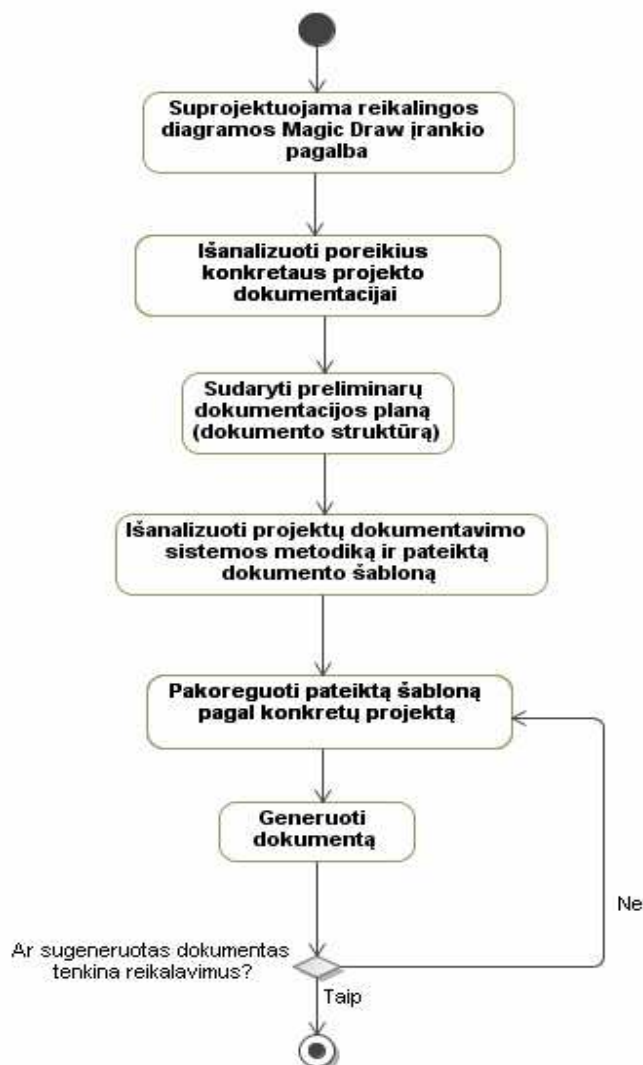
3.2 pav. Metodikos taikymo schema

Kaip jau buvo minėta analizės dalyje, dauguma projektų kūrimo metodikų dokumentavimo procesą laiko antraeilu dalyku projekto kūrime, todėl pačiam dokumentavimo proceso aprašymui arba skiriama labai mažai laiko, arba iš vis neskiriama. Kuriama projektų dokumentavimo sistema turėtų padėti asmenims, kuriantiems dokumentus projektams, atlikti šį darbą sparčiau ir paprasčiau.

Panaši dokumentų generavimo strategija, jei naudojama, tai naudojama įmonių vidinėms reikmėms, neviešinant dokumentų šablonų ar metodikos, kaip jais naudotis. Sukurtuoju projektų dokumentavimo šablonu bei nurodymais, kaip juo naudotis, galėtų naudotis visos suinteresuotos organizacijos ar įmonės, nes šablonas kuriamas nekomerciniais tikslais.

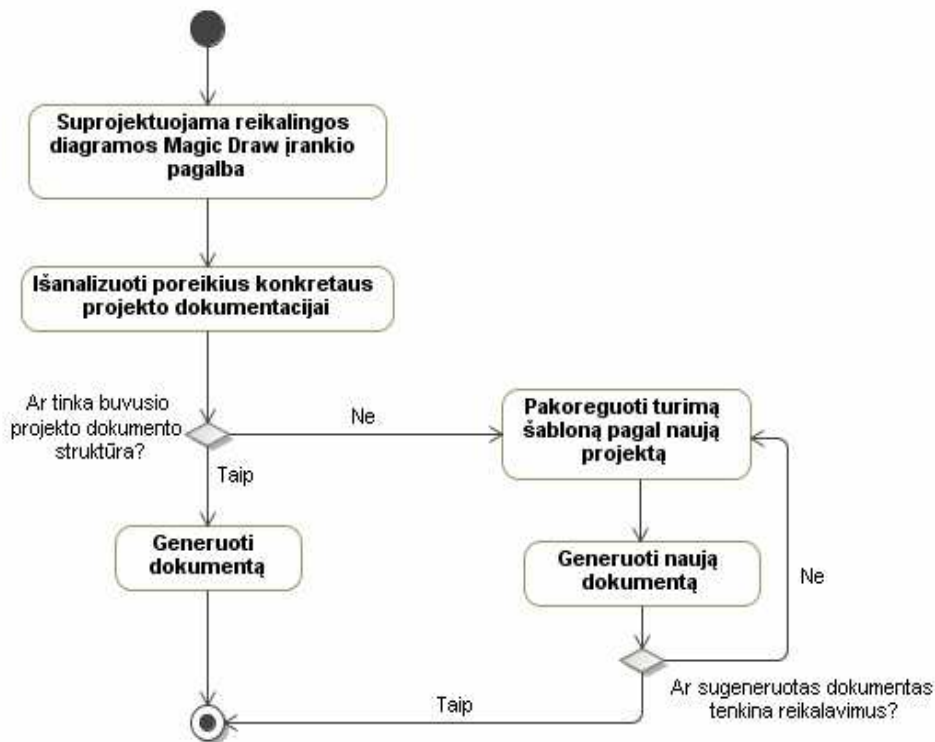
Nepaisant to, kad *MagicDraw UML* įrankyje yra nemažai sukurtų ataskaitų šablonų, tačiau tos ataskaitos yra gremėzdiškos, jose gausu mažai naudojamų elementų, todėl tokias ataskaitas neracionalu naudoti konkrečiam projektui. Be to, nėra šablono, kuris būtų skirtas visam projektui, o ne atskiroms jo dalims. Projektų dokumentavimo sistemos dokumento šablonas skirtas konkrečiam projektui ir apima visą projekto kūrimo procesą – nuo reikalavimų identifikavimo iki realizavimo. Nepaisant to, kad pradžioje gali užtrukti, kol vartotojas pritaikys sistemų projektavimo ir dokumentavimo metodiką, tačiau, kuriant dokumentus kitam projektui, dokumentavimo procesas turėtų paspartėti.

Pradinis projektų dokumentavimo sistemos panaudojimas pateiktas 3.3 paveiksle.



**3.3 pav.** Pradinis projektų dokumentavimo sistemos panaudojimas

Kadangi, dokumentuojant projektus, dokumento struktūra išlieka panaši ar net tokia pati, todėl, kaip matoma 3.4 paveiksle, dokumentuojant kitą projektą, projektų dokumentavimo sistemos panaudojimo žingsnių gali sumažėti. Jei naujojo projekto dokumento struktūra labai skiriasi nuo buvusiojo, tuomet gali prireikti pritaikyti turimą šabloną naujam projektui. Bet kuriuo atveju, kuo daugiau šablonų turės dokumentavimą atliekantys asmenys, tuo didesnė tikimybė, jog tarp tų šablonų bus toks, kuris bus tinkamas naujam projektui ir, jei prireiks atlikti kokius nors pakeitimus šablone, tie pakeitimai bus minimalūs.



3.4 pav. Pakartotinis projektų dokumentavimo sistemos naudojimas

### 3.3. Dokumento šablono koregavimas

Projektų dokumentavimo sistemos panaudojimo atžvilgiu bene svarbiausias žingsnis yra šablono koregavimas, nes būtent šiame žingsnyje naudojamas sistemos svarbiausias artefaktas – šablonas. Šablono koregavimo procesas smulkiau detalizuotas 3.5 paveiksle.



3.5 pav. Šablono koregavimo procesas

Projektų dokumentavimo sistemos ataskaitos šablonas bus eksportuotas \*.mrzip formatu. Norint naudoti šabloną, jį reikės importuoti į konkretaus vartotojo aplinką *MagicDraw UML* įrankyje.

Šablone bus sukurti kintamieji, kurių reikšmės atspindės sugeneruotoje ataskaitoje. Labai maža tikimybė, kad vartotojui reikės visų kintamųjų, todėl nereikalingi kintamieji turėtų būti pašalinti tam, kad būtų lengviau naudoti šabloną bei nebūtų kintamųjų dubliavimosi.

Sukūrus naujus kintamuosius bei pašalinus nereikalingus, galima redaguoti patį šablono dokumentą, saugomą \*.rtf formatu, kurį supranta dauguma tekstinių redaktorių, įskaitant MS WORD. Norint, kad kintamųjų reikšmės būtų sugeneruotos ataskaitoje, reikės pagal projektų dokumentavimo sistemoje esančius nurodymus įterpti kintamuosius į ataskaitą.

Visa tai atlikus, galima generuoti dokumentą, ir, prireikus, atlikti korekcijas.

### 3.4. Projekto programavimo ir testavimo darbų apimčių skaičiavimas

Skaičiuojant programavimo darbų apimtį, vertinami tokie techniniai faktoriai:

Faktorius	Apibūdinimas	Svoris
T1	Išskaidytoji sistema	2.0
T2	Atsako greitis	1.0
T3	Našumo reikalavimai	1.0
T4	Sudėtingas apdorojimas	1.0
T5	Pakartotinis kodo panaudojimas	1.0
T6	Lengvai instaliuojama	0.5
T7	Lengvai naudojama	0.5
T8	Nešiojama	2.0
T9	Lengvai keičiama	1.0
T10	Panaudotas lygiagretaus kūrimo principai	1.0
T11	Saugumo reikalavimai	1.0
T12	Prieiga tretiesiems asmenims	1.0
T13	Reikalingi specialūs apmokymai	1.0

Skaičiuojant testavimo darbų apimtį, vertinami tokie techniniai faktoriai:

Faktorius	Apibūdinimas	Svoris
T1	Testavimo įrankiai	5.0
T2	Įvedamų reikšmių dokumentavimas	5.0
T3	Kūrimo aplinka	2.0

T4	Testavimo aplinka	3.0
T5	Testavimo įrankių pakartotinis panaudojimas	3.0
T6	Išskaidytoji sistema	4.0
T7	Našumo reikalavimai	2.0
T8	Saugumo reikalavimai	4.0
T9	Sudėtinga vartotojo sąsaja	5.0

Vertinami aplinkos faktoriai:

Faktorius	Apibūdinimas	Svoris
E1	Susipažinęs su RUP	1.5
E2	Ssuipažinęs su kuriama sistema	0.5
E3	Susipažinęs su objektiniu kūrimu	1.0
E4	Kompetetingi vadovai	0.5
E5	Motyvacija	1.0
E6	Stabilūs reikalavimai	2.0
E7	Laikini darbuotojai	-1.0
E8	Sudėtinga programavimo kalba	-1.0

Svoris žymi kiekvieno faktoriaus svarbą. Pažymėkime  $T_i$  – i-tajam techniniam faktoriui projektuotojo priskirta svorinė reikšmė,  $E_i$  – i-tajam aplinkos faktoriui priskirta reikšmė, o  $W_i$  – atitinkamo faktoriaus svoris. Pažymėkime  $T_{Factor}$  – bendra techninių faktorių vertė,  $E_{Factor}$  – bendra aplinkos faktorių reikšmė.

$T_{Factor} = \sum (T_i * W_i)$ , kur  $i=1..13$  (programavimo darbų apimties skaičiavimo atveju)

arba  $i=1..9$  (testavimo darbų apimties skaičiavimo atveju)

$E_{Factor} = \sum (E_i * W_i)$ , kur  $i=1..8$

Techninio sudėtingumo vertė TCF gaunama pagal formulę:

$TCF = 0,6 + (0,01 * T_{Factor})$  - programavimo darbų skaičiavimo atveju

$TCF = 0,65 + (0,01 * T_{Factor})$  - testavimo darbų skaičiavimo atveju

Aplinkos faktoriaus vertė ECF gaunama pagal formulę:

$ECF = 1,4 + (-0,03 * E_{Factor})$ .

Tarkime, kad  $U_i$  yra vieno panaudojimo atvejo sudėtingumo svorinė vertė, o  $A_j$  – vieno aktoiaus sudėtingumo svorinė reikšmė. Tuomet UUCP apskaičiuojama taip:

$UUCP = \sum U_i + \sum A_j$ , kur  $i = 1..n$ ,  $j = 1..m$ .

Suradus UUCP, TCF ir ECF, galima nustatyti apskaičiuojamų panaudojimo atvejų taškų reikšmę UCP (*adjusted Use Case Points*):

$UCP = UUCP * TCF * ECF$



Tarkime, kad  $H$  yra pagal ankstesnių projektų patirtį nustatytas valandų skaičius, kuris sugaištamam vienam panaudojimo atvejui sukurti/ištvestuoti. Tuomet viso projekto kūrimo/testavimo darbų apimtis žmogaus darbo valandomis apskaičiuojama UCP reikšmę padauginus iš  $H$ , t.y.  $UCP * H$ .

Tarkime, projekto kūrimo dalyvaujam  $d$  programuotojų/testuotojų ir jie dirba  $dh$  darbo valandų per dieną, o mėnesyje yra 20 darbo dienų. Tuomet, pagal projekto komandos sudėtį, projekto kūrimas/testavimas užtruks:

$$UCP * H \div d \div dh \text{ dienų arba } UCP * H \div d \div dh \div 20 \text{ mėnesių.}$$

### 3.5. Projekto elementų ryšių pilnumo nustatymas

Projekto elementų ryšių pilnumo nustatymas turėtų įvertinti, ar, pereinant nuo skirtingų projekto kūrimo etapų, t.y. nuo reikalavimų prie analizės, nuo analizės prie realizavimo, visi projekto elementai perkeliama į kitą etapą ir ar pilnai realizuoti. Turėtų būti tikrinami tokie ryšiai:

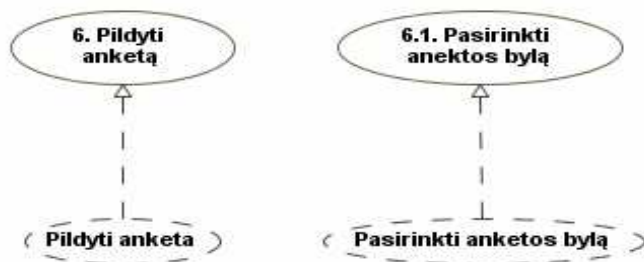
- ar panaudojimo atvejai realizuoti projekto klasėmis;
- ar panaudojimo atvejai iš reikalavimų identifikavimo realizuoti *Collaboration* elementais;
- ar *Collaboration* elementai susieti *Dependency* ryšiu su klasėmis iš reikalavimų analizės;
- ar *Collaboration* elementai realizuoti klasėmis;
- ar komponentai realizuoti artefaktais;
- ar komponentai realizuoti klasėmis.

Tarkime, reikalavimų identifikavimo metu buvo sudarytas toks veiklos modelis:



3.6 pav. Reikalavimų identifikavimo metu sudaryto veiklos modelio pavyzdys

Pereinant nuo reikalavimų identifikavimo prie analizės, sudarytas veiklos modelis turi būti perkeltas į analizės dalį, t.y. panaudojimo atvejai turi būti realizuoti *Collaboration* tipo elementais. Tarkime, jei per klaidą projektuotojas neperkėlė panaudojimo atvejo „6.2. Prisijungti prie sistemos“ iš reikalavimų identifikavimo į analizę ir turime tokią situaciją, kad tik du panaudojimo atvejai iš trijų buvo realizuoti *Collaboration* tipo elementais:



**3.7 pav.** Perėjimas nuo reikalavimų identifikavimo prie analizės

Tuomet būtų naudinga, kad, sugeneravus dokumentą, būtų suformuojama dokumento dalis, kurioje būtų matyti, kuriems elementams trūksta tam atitinkamo tipo ryšių.

Projekto elementų ryšių pilnumo tikrinimas	
Panaudojimo atvejis, nerealizuotas <i>Collaboration</i> elementu:	
Elemento pavadinimas	Tėvinis elementas
6.2. Prisijungti prie anketos	1.1. Panaudojimo atveju modelis

**3.8 pav.** Dokumento fragmentas, kuris galėtų būti sugeneruojamas, aptikus trūkstamą ryšį tarp elementų

## **4. Pavyzdinės apklausų informacinės sistemos projekto aprašas**

Tam, kad būtų ištirtas dokumentacijos pobūdis, išskirti reikalingi dokumentacijos atributai bei diagramos bei išbandyta ir patikrinta sukurtoji metodika, projektuojama pavyzdinė sistema, kuriai, realizavus projektą dokumentavimo sistemą, bus sugeneruota dokumentacija.

Tarkime, kad kuriama Apklausų projektavimo ir vykdymo sistema, skirta palengvinti ir paspartinti apklausų kūrimą bei vykdymą. Šią sistemą planuojama panaudoti visose srityse, kur kuriamos ar vykdomos apklausos, pavyzdžiui, mokyklose, universitetuose ir pan. Sistemos pagalba vartotojas gali pilnai valdyti apklausas – sukurti anketas, jas redaguoti bei šalinti, teikti respondentams pildyti anketas, surinkti užpildytas respondentų anketas, peržiūrėti užpildytų anketų duomenis matricos bei grafiko pavidalu. Sistemos panaudojimas įmonės, teikiančios viešosios nuomonės ir rinkos tyrimų paslaugas, leidžia sumažinti apklausų vykdymo kaštus, visiškai atsisakant „popierinių apklausų“.

### **4.1. Pavyzdinės apklausų informacinės sistemos funkciniai reikalavimai**

- Sistemoje turi būti trys vartotojų grupės – administratoriai, vartotojai ir respondentai.
- Asmuo, turintis prisijungimo vardą ir slaptažodį, vadinamas vartotoju. Jis gali:
  - Kurti anketas;
  - Redaguoti anketas;
  - Peržiūrėti anketas;
  - Šalinti anketas;
  - Eksportuoti anketos duomenis internetiniam anketavimui.
  - Peržiūrėti apklausos rezultatus matricos arba grafiniu pavidalu.
- Administratoriai – tai privilegijuotas vartotojas, kuris, be paprasto vartotojų galimybių, taip pat:
  - Suteikia vartotojams prisijungimo vardus ir slaptažodžius, kad šie galėtų projektuoti ir vykdyti apklausas;
  - Keičia vartotojų duomenis;
  - Šalina vartotojų duomenis iš sistemos.

- Respondentas – asmuo, pildantis anketą. Priklausomai nuo anketos pobūdžio, anketa gali būti anoniminė (respondentu gali būti bet kuris asmuo) arba skirta konkrečiam ratui respondentų (kuriems anketų kūrimo metu sukuriama prisijungimo vardai ir slaptažodžiai).

Be to, sistemoje turi būti realizuota vartotojų prisijungimo prie sistemos funkcija bei naujų vartotojų registravimo funkcija. Vartotojų prisijungimo funkcija tikrina, ar vartotojas turi prisijungimo teises ir pagal jo tipą nukreipia arba į vartotojų administravimo, arba anketų kūrimo dalį. Prisijungęs administratorius turi matyti savo sukurtas anketas, turėti galimybę jas kurti, redaguoti, šalinti arba eksportuoti respondentų atsakinėjimui, peržiūrėti apklausų rezultatus matricos pavidalu ir/arba grafiškai.

Jei norima, kad anketas pildytų konkretūs asmenys, turi būti galimybė suteikti šiems respondentams unikalius prisijungimo vardus bei slaptažodžius. Slaptažodžiai įrašomi ranka arba sugeneruojami automatiškai. Respondentų sąrašas gali būti eksportuojamas į failą. Jei suformuojamas konkretus respondentų sąrašas, tuomet, prieš pildant anketą, reikia įvesti suteiktą respondento prisijungimo vardą ir slaptažodį.

## 4.2. Pavyzdinės apklausų informacinės sistemos reikalavimų specifikavimas, naudojant *Volere* šabloną

Apklausų informacinės sistemos reikalavimus galima suklasifikuoti panaudojant *Volere* šabloną, kur „F“ žymi funkcinius, „NF“ – nefunkcinius reikalavimus:

<b>Reikalavimas #:</b>	<b>1</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	2.; 4.3
<b>Aprašymas:</b>	Jei anketa ne anoniminė, turi būti galimybė priskirti respondentams prisijungimo vardus ir slaptažodžius.				
<b>Pagrindimas:</b>	Kartais kyla poreikis identifikuoti respondentus, kai jų skaičius nedidelis.				
<b>Šaltinis:</b>	Apklausų projektuotojas				
<b>Tinkamumo kriterijus:</b>	Respondentas informuojamas apie neteisingai įvestus prisijungimo duomenis.				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>2</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	2.; 4.3
<b>Aprašymas:</b>	Anketos redagavimo vartotojo sąsajos aplinka turi būti tokia pat, kaip ir jos projektavimo.				
<b>Pagrindimas:</b>	Siekiami kuo paprastesnės ir vieningesnės vartotojo sąsajos.				
<b>Šaltinis:</b>	Apklausų projektuotojas				
<b>Tinkamumo kriterijus:</b>	Vienoda vartotojo sąsaja anketų projektavime ir redagavime.				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>3</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	1; 1.1; 1.3
<b>Aprašymas:</b>	Įkeliamų anketų rezultatai sinchronizuojami.				
<b>Pagrindimas:</b>	Automatiškai sinchronizuojant rezultatus, sutaupoma laiko.				
<b>Šaltinis:</b>	Analitikas				
<b>Tinkamumo kriterijus:</b>	Matomi visų įkeltų anketų rezultatai.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>4</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>4</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	1; 1.1
<b>Aprašymas:</b>	Turi būti galimybė peržiūrėti anketų rezultatus matricos arba grafikų pavidalu.				
<b>Pagrindimas:</b>	Toks pavidalas vaizdžiausias anketų rezultatų peržiūrai.				
<b>Šaltinis:</b>	Analitikas				
<b>Tinkamumo kriterijus:</b>	Korektiškai atvaizduoti rezultatai matricos arba grafiko pavidalu.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>5</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	2.; 4.3
<b>Aprašymas:</b>	Kuriant arba redaguojant anketą, turi būti galimybė kurti ir pakartotinai panaudoti klausimų šablonus.				
<b>Pagrindimas:</b>	Sutaupoma laiko kuriant ir redaguojant anketas, nes nereikia dažnai pasikartojančių klausimų kurti iš naujo.				
<b>Šaltinis:</b>	Apklausų projektuotojas.				
<b>Tinkamumo kriterijus:</b>	Galimybė kurti klausimų šablonus.				
<b>Užsakovo patenkinimas:</b>	<b>4</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					

<b>Reikalavimas #:</b>	<b>6</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	2.; 4.3
<b>Aprašymas:</b>	Turi būti galimybė respondentų prisijungimo vardus ir slaptažodžius generuoti automatiškai.				
<b>Pagrindimas:</b>	Sutaupoma laiko, jei būtina priskirti respondentams prisijungimo vardus ir slaptažodžius.				
<b>Šaltinis:</b>	Apklausų projektuotojas.				
<b>Tinkamumo kriterijus:</b>	Generatorius respondentų prisijungimo vardams ir slaptažodžiams.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>2</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					

<b>Reikalavimas #:</b>	<b>7</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	2.; 4.3
<b>Aprašymas:</b>	Turi būti galimybė respondentų sąrašą su prisijungimo vardais ir slaptažodžiais eksportuoti į tekstinį failą.				
<b>Pagrindimas:</b>	Atspausdinta lapą su respondentų prisijungimo vardais ir slaptažodžiais galima sukarpyti ir išdalinti konkreitiems respondentams.				
<b>Šaltinis:</b>	Anketuotojas				
<b>Tinkamumo kriterijus:</b>	Galimybė eksportuoti minėtus duomenis į tekstinį failą.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>1</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					

<b>Reikalavimas #:</b>	<b>8</b>	<b>Reikalavimo tipas:</b>	F	<b>Įvykis/panaudojimo atvejis #:</b>	6
<b>Aprašymas:</b>	Galimybė pildyti anketas ne tik pelės, bet ir klaviatūros pagalba.				
<b>Pagrindimas:</b>	Kadangi anketų pildymo vieta gali būti mobili (gatvė, kavinė ir kt.), todėl ne visada patogiu anketas pildyti anketas.				
<b>Šaltinis:</b>	Anketuotojas				
<b>Tinkamumo kriterijus:</b>	Galima anketas pildyti ne tik pelės, bet ir klaviatūros pagalba.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>1</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>9</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Neįkyri, nereikalaujanti ką nors kelis kartus tvirtinti, sąsaja.				
<b>Pagrindimas:</b>	Sistemą su neįkyria vartotojo sąsaja palankiau vertins jos vartotojai.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Neįkyri vartotojo sąsaja.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>10</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Lengvai skaitoma sąsaja bei paprastas ir nesudėtingas panaudojimas.				
<b>Pagrindimas:</b>	Siekama, kad vartotojai kuo greičiau įsisavintų naująją sistemą.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Lengvai skaitoma sąsaja bei paprastas ir nesudėtingas panaudojimas.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>11</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	1.1
<b>Aprašymas:</b>	Vienos apklausos (iki 1000 anketų) duomenys turi būti įkelti greičiau nei per 2 sekundes.				
<b>Pagrindimas:</b>	Ilgesnis laiko tarpas neigiamai veikia.				
<b>Šaltinis:</b>	Anketuotojas				
<b>Tinkamumo kriterijus:</b>	Vienos apklausos (iki 1000 anketų) duomenys įkeliami greičiau nei per 2 sekundes.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>12</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	6
<b>Aprašymas:</b>	Kadangi tenka per trumpą laiką apklausti daug respondentų, todėl vartotojui tenka dirbti paskubomis. Todėl reikia, kad būtų galimybė kuo greičiau pildyti anketas.				
<b>Pagrindimas:</b>	Greitai pildant anketas, sutaupoma laiko ir galima apklausti daugiau žmonių.				
<b>Šaltinis:</b>	Anketuotojas				
<b>Tinkamumo kriterijus:</b>	Nepatyręs anketuotojas užpildo 5 anketas per 10 min.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>13</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Kadangi įmonė planuoja pereiti prie LINUX operacinės sistemos, todėl turi būti galimybė naudoti sistemą šioje operacinėje sistemoje.				
<b>Pagrindimas:</b>	Įmonės veiklos plėtros strategijoje numatyta pereiti prie LINUX operacinės sistemos.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Galimybė naudoti sistemą LINUX operacinėje sistemoje.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>1</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

<b>Reikalavimas #:</b>	<b>14</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Vartotojai su skirtingais prisijungimo vardais ir slaptažodžiais mato tik savo sukurtas anketas ir jų rezultatus.				
<b>Pagrindimas:</b>	Anketų informacija gali būti konfidenciali.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Vartotojai su skirtingais prisijungimo vardais ir slaptažodžiais mato tik savo sukurtas anketas ir jų rezultatus.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

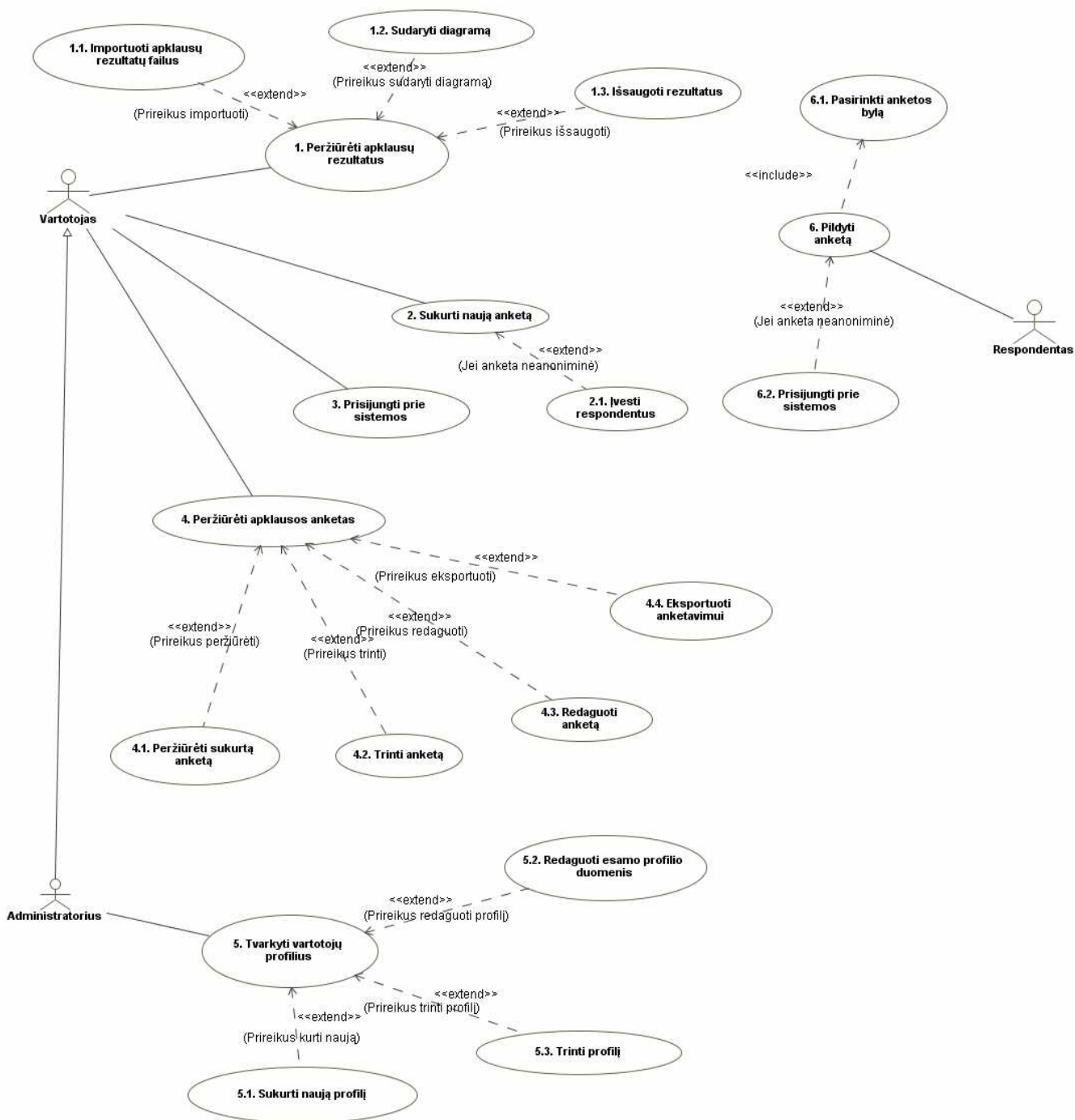
<b>Reikalavimas #:</b>	<b>15</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistemoje turi būti naudojamas lietuviškas alfabetas.				
<b>Pagrindimas:</b>	Anketos pildomos Lietuvių kalba.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Sistemoje naudojamas lietuviškas alfabetas.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					



<b>Reikalavimas #:</b>	<b>16</b>	<b>Reikalavimo tipas:</b>	NF	<b>Įvykis/panaudojimo atvejis #:</b>	Visi
<b>Aprašymas:</b>	Sistemoje kūrime neturi būti naudojama piratinė programinė įranga.				
<b>Pagrindimas:</b>	Piratinės įrangos naudojimas kenkia įmonės prestižui.				
<b>Šaltinis:</b>	Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Sistemos kūrime naudojama tik legali programinė įranga.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		
<b>Priklausomybės:</b>	<b>Nėra</b>	<b>Konfliktai:</b>	<b>Nėra</b>		
<b>Papildoma medžiaga:</b>					
<b>Istorija:</b>					

### 4.3. Pavyzdinės apklausų informacinės sistemos projektas

4.1 paveiksle pateikiama eksperimente naudojamos Apklausų informacinės sistemos panaudojimo atvejų modelis.



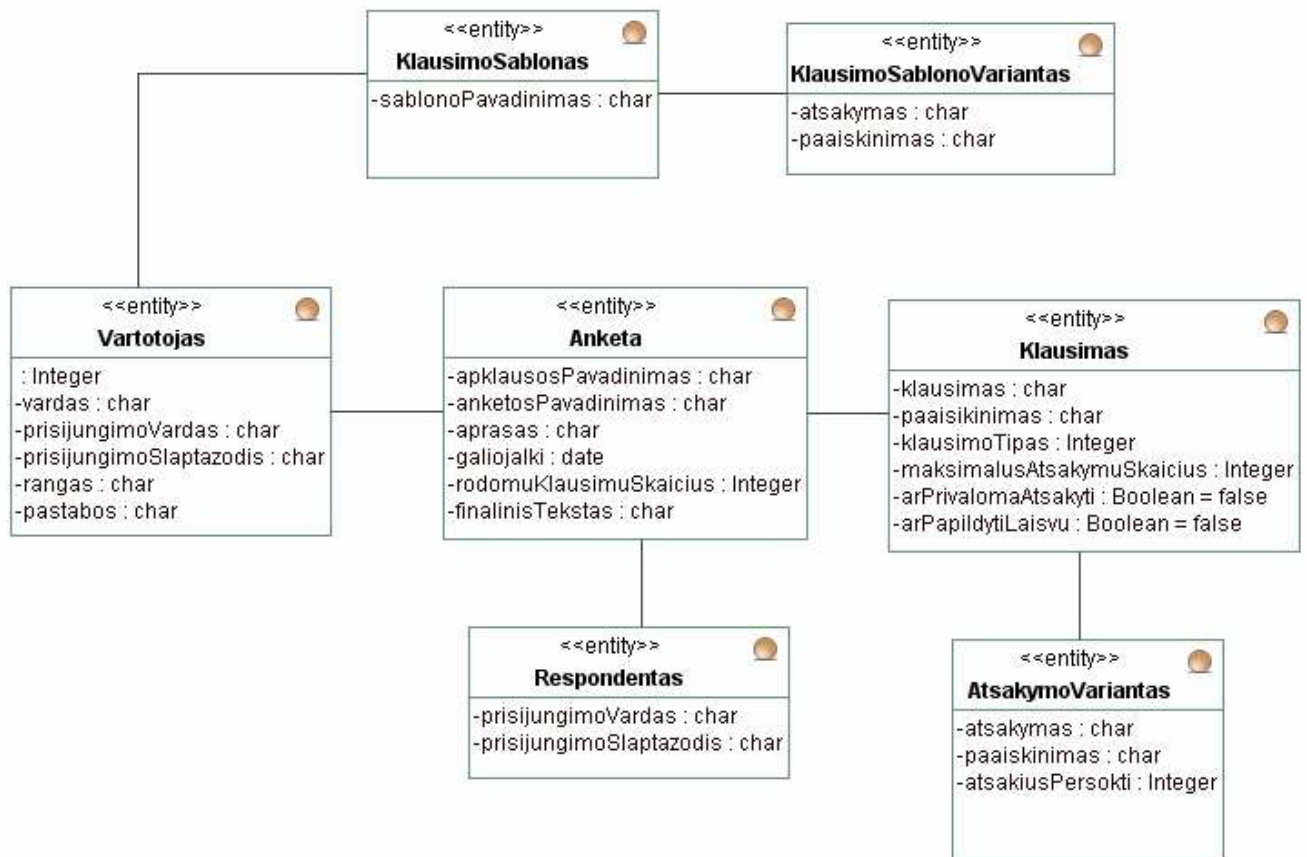
4.1 pav. Pavydinės sistemos panaudojimo atvejų modelis

4.1 lentelėje pateikiama panaudojimo atvejų, kurie matomi 4.1 paveiksle, aprašai. Pagrindiniai panaudojimo atvejai detalizuoti veiklos ir sekų diagramomis 9.3 priedo 9.1 – 9.9 paveiksluose.

#### 4.1 lentelė. Panaudojimo atvejų aprašai

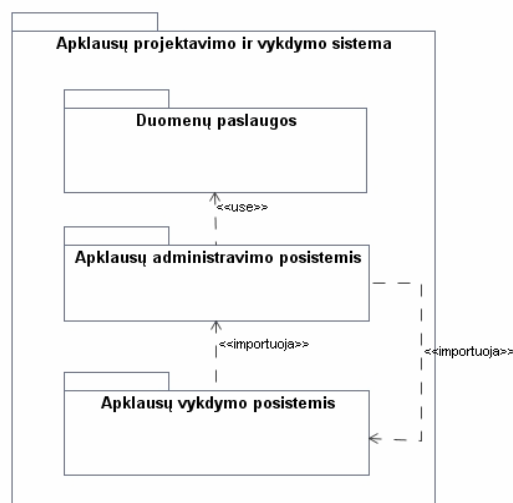
Pavadinimas	Aprašas
1. Peržiūrėti apklausos rezultatus	Apklausos rezultatų peržiūra.
1.1. Importuoti apklausos rezultatų failus	Kadangi gali būti daug failų, kuriuose saugomi užpildytų anketų rezultatai, todėl reikia pasirinkti, kurių anketų rezultatus norima vaizduoti (pagal nutylėjamą matricos pavidalą).
1.2. Sudaryti diagramą	Konkretaus klausimo rezultatų vizualizavimas grafiko pavidalu.
1.3. Išsaugoti rezultatus	Suimportuotų apklausos atskirų anketų rezultatų išsaugojimas viename faile.
2. Sukurti anketą	Kuriama nauja anketa su visais jos duomenimis.
2.1. Įvesti respondentus	Jei anketa neanoniminė, įvedami respondentų prisijungimo prie anketos vardai ir slaptažodžiai.
3. Prisijungti prie sistemos	Įvesti prisijungimo duomenis, pagal kuriuos nukreipiama į vartotojo arba administratoriaus darbo langą arba išmetamas informacinis pranešimas apie neteisingus prisijungimo duomenis.
4. Peržiūrėti apklausos anketas	Leidžia atlikti numatytus veiksmus su anketomis.
4.1. Peržiūrėti sukurtą anketą	Sukurtų anketų peržiūra.
4.2. Trinti anketą	Nereikalingų anketų šalinimas.
4.3. Redaguoti anketą	Sukurtos anketos redagavimui.
4.4. Eksportuoti anketą	Anketų duomenų eksportavimas į failą, kuris naudojamas anketos pildymui.
5. Tvarkyti paskyrą	Leidžia atlikti numatytus veiksmus su vartotojų duomenimis.
5.1. Sukurti naują paskyrą	Naujam vartotojui sukurti.
5.2. Redaguoti esamos paskyros duomenis	Informacijos apie egzistuojantį vartotojo duomenims keisti.
5.3. Trinti paskyrą	Egzistuojančio vartotojo paskyrai pašalinti.
6. Pildyti anketą	Atsakinėjama į anketos klausimus.
6.1. Pasirinkti anketos bylą	Pasirenkamas anketos, kurią norima pildyti, failas.
6.2. Prisijungti prie sistemos	Jei anketa neanoniminė, tuomet respondentas turi turėti prisijungimo vardą bei slaptažodį tam, kad galėtų pildyti anketą.

4.2 paveiksle pateiktas sistemos projektų naudojamų esybių diagrama. Pagal šią diagramą galima susidaryti bendrą sistemos esybių tarpusavio priklausomybių vaizdą..



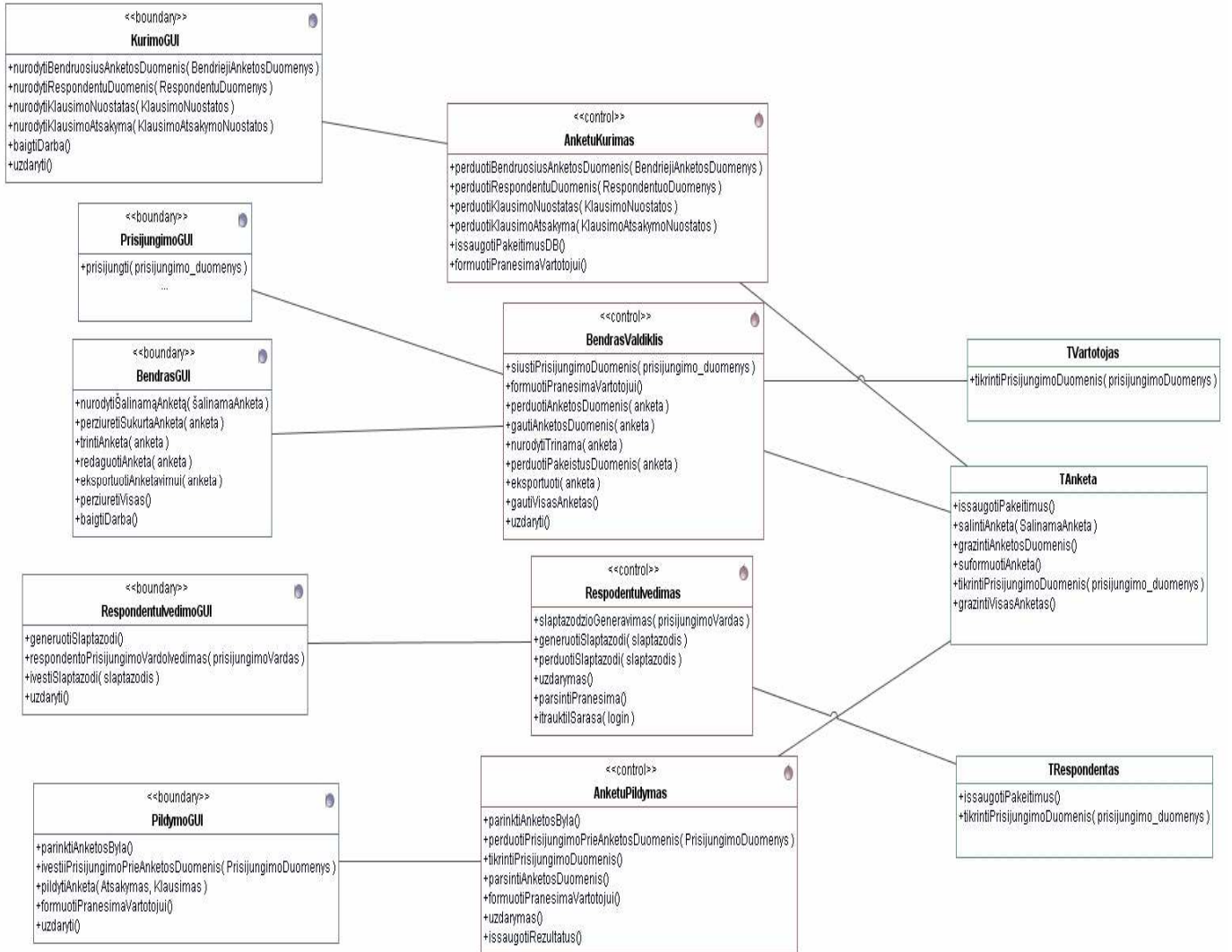
4.2 pav. Pavyzdinės sistemos esybių ryšių modelis

Sistemos loginė architektūra pateikta 4.3 paveiksle. Visą sistemą galima struktūrizuoti pagal skirtingų vartotojų kategorijų galimus atliktus veiksmus, t.y. administratorius administruoja sistemos vartotojus, paprasti vartotojai kuria anketas, o respondentai jas pildo.



4.3 pav. Loginė sistemos architektūra

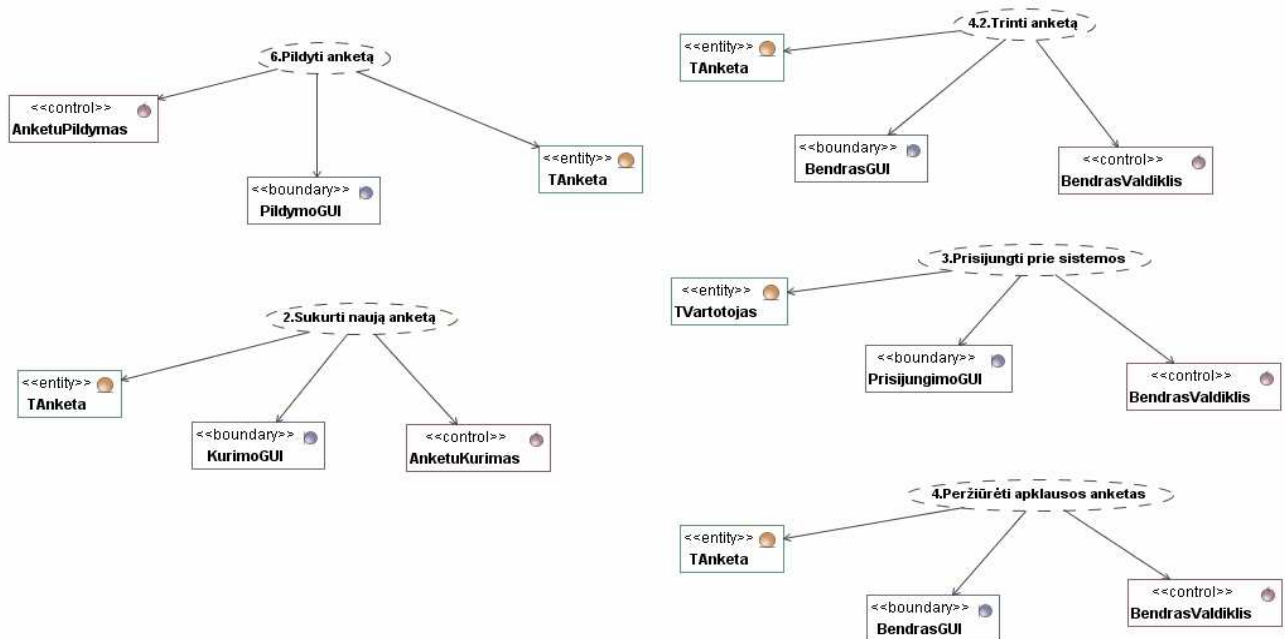
9.3 priedo 9.10 - 4.15 paveiksluose detaliai specifikuojami pagrindiniai sistemos panaudojimo atvejai. Stengiamasi išlaikyti vartotojo sąsaja→valdiklis→duomenų bazė architektūros principų.



4.4 pav. Trijų lygių klasių diagrama

4.4 paveiksle matoma panaudojimo atvejų realizacijos analizės klasėmis logika – projekto klases galima suskirstyti į tris sluoksnius – vartotojo sąsajos klasės, valdiklių klasės ir duomenų bazės klasės, kur kiekviena klasių grupė specifikuoja tam tikrų sistemos dalių paskirtį.

Panaudojimo atvejų realizacijos analizės klasėmis diagrama nurodo (4.5 paveikslas), kokiomis klasėmis realizuojami tam tikri panaudojimo atvejai. Kadangi naudojama trijų sluoksnių architektūra – vartotojo sąsaja→valdiklis→duomenų bazė – todėl kiekvienam panaudojimo atvejui realizuoti minimaliai pakanka trijų klasių.



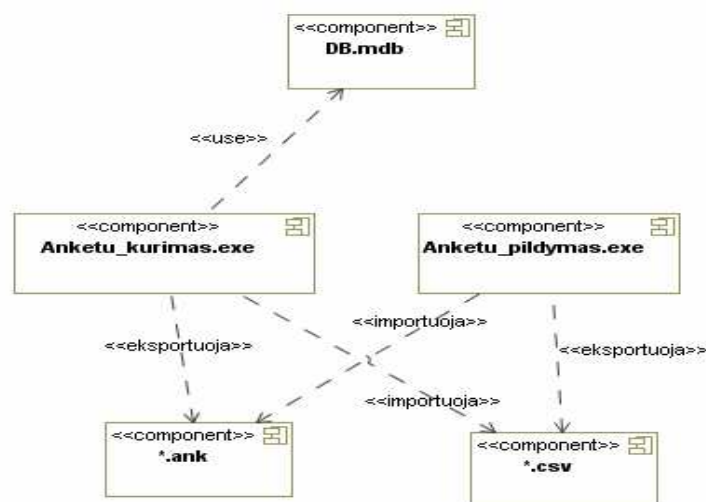
4.5 pav. Pavyzdinės sistemos panaudojimo atvejų realizacijų klasių diagrama

Realizavus sistemą, vartotojui būtų pateikiama du atskiri Anketu\_kurimas.exe ir Anketu\_pildymas.exe failai (vienas anketų kūrimui bei vartotojų administravimui, kitas anketų pildymui) bei DB.mdb failas, kuriame būtų realizuota sistemos duomenų bazė.

Sukūrus anketą, ji būtų išsaugoma \*.ank tipo faile, kuris naudojamas anketų pildymui. Užpildyta anketa išsaugoma \*.csv tipo faile, kuris importuojamas į sistemą.

Vartotojų administravimui nenumatyta atskiro komponento – vartotojų administravimas realizuojamas Anketu\_kurimas.exe faile, tiesiog pagal vartotojo teises bus nustatoma, rodyti ar ne vartotojų administravimo įrankius.

Principinis realizacijos modelis matomas 4.6 paveiksle.



4.6 pav. Realizacijos modelis

## 5. Pavyzdinės sistemos ataskaitų šablonų realizavimas *MagicDraw UML* įrankiu

### 5.1. *MagicDraw UML* ataskaitų vedlio (*Report Wizard*) taikymas šablonams kurti

Dokumentavimo sistemos metodikos panaudojimas iliustruojamas *MagicDraw UML* įrankio pagalba, panaudojant *MagicDraw UML* įrankio ataskaitų vedlį (*Report Wizard*).

Pats *MagicDraw UML* įrankis atitinka naujausius Java bei UML technologijų standartus, turi vieną iš patikimiausių išeities kodų inžinerijos mechanizmų Java, C#, C++ ir CORBA IDL programavimo kalboms bei gali vykdyti šių kalbų kodo atvirkštinę inžineriją, duomenų bazių schemų atvirkštinę inžineriją, kodo bei duomenų bazių schemų generavimą. *MagicDraw UML* panaudoja "roundtrip" technologiją, leidžiančią keisti tiek OO modelį, tiek programos kodą bet koku eiliškumu, juos nuolat sinchronizuojant. *MagicDraw UML* yra vienas iš nedaugelio rinkoje esančių paketų, leidžiančių greičiau nubraižyti UML diagramas bei turintis UML diagramų semantinio teisingumo tikrinimo ir modelio validavimo mechanizmą.

*MagicDraw UML* naudoja UML grafinio modeliavimo kalbą, nepriklausomą nuo konkrečios programavimo kalbos. UML naudoja vieningą terminologiją, taigi programuotojai, nepriklausomai nuo jų naudojamos programavimo kalbos, gali lengviau susišnekėti tarpusavyje. *MagicDraw UML* leidžia specifikuoti, vizualizuoti ir dokumentuoti programinės įrangos sistemų modelius – jų struktūrą ir projektus.

*MagicDraw UML* yra Java pagrindu veikiantis įrankis, kuriuo galima naudotis bet kurioje operacinėje sistemoje (Windows (98 ir naujesnėse), Mac OS X, UNIX (Solaris, HP-UX, AIX) ir Linux), palaikančioje Java 1.4 ir Java 1.5 versijas.

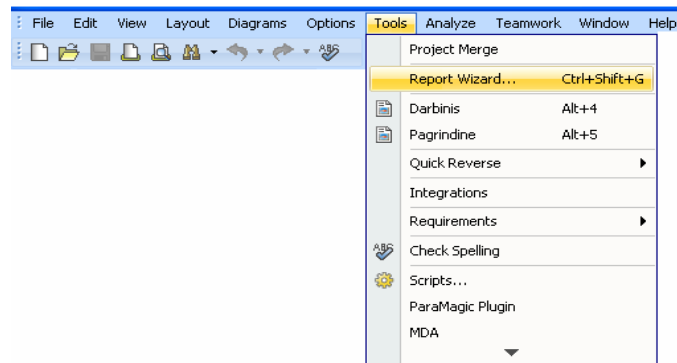
*MagicDraw UML* ataskaitų vedlys yra ataskaitų generavimo priemonė, sutinkama nuo *MagicDraw UML* 14.0 versijos. *MagicDraw UML* ataskaitų vedlys skirtas išspręsti senesnių panašaus pobūdžio priemonių, tokių kaip XSL/XSLT ir JPython, problemas.

Ataskaitų vedlys, sukurtas *Velocity Engine* (atviro kodo šablonų kūrimo įrankio) pagrindu, palaiko tekstinius šablonus ataskaitų generavimui. Ataskaitos formatas priklauso nuo šablono tipo. Ataskaitų vedlys palaiko paprasto teksto, RTF, HTML, Office Open XML (ISO/IEC 29500:2008), OpenDocument format (ISO/IEC 26300) ir XML šablonus (DocBook arba FO).

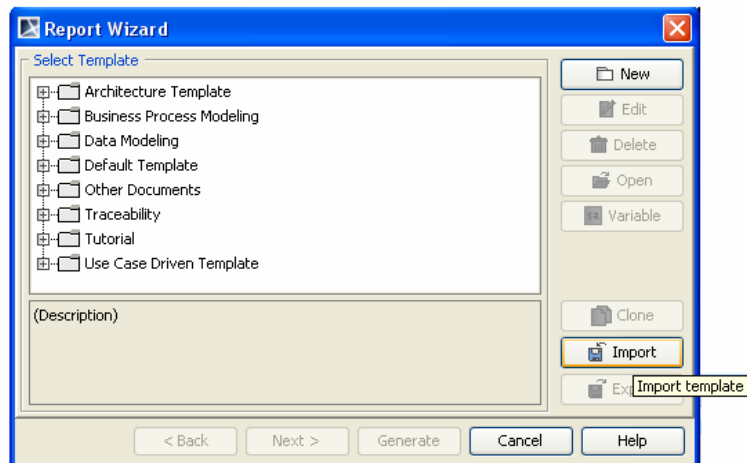
Dokumentavimo sistemos metodikos realizacija atlikta panaudojant *MagicDraw UML* 16.8 versiją. Šablonai bei sugeneruotos ataskaitos saugomos \*.rtf formatu.

## 5.2. Dokumento šablono importavimas

Ataskaitos šablonas sukurtas panaudojant VTL (*Velocity Template Language*) ir išsaugotas \*.mrzip formatu. Ataskaitos šablonas importuojamas į *MagicDraw UML* įrankį standartinėmis *MagicDraw UML* priemonėmis:



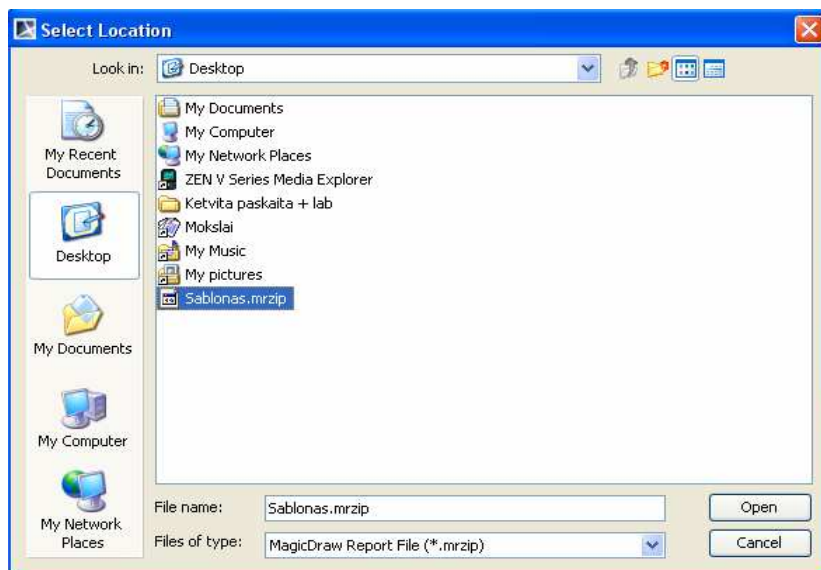
5.1 pav. Kelias iki *Report Wizard* įrankio



5.2 pav. *Report Wizard* įrankio langas

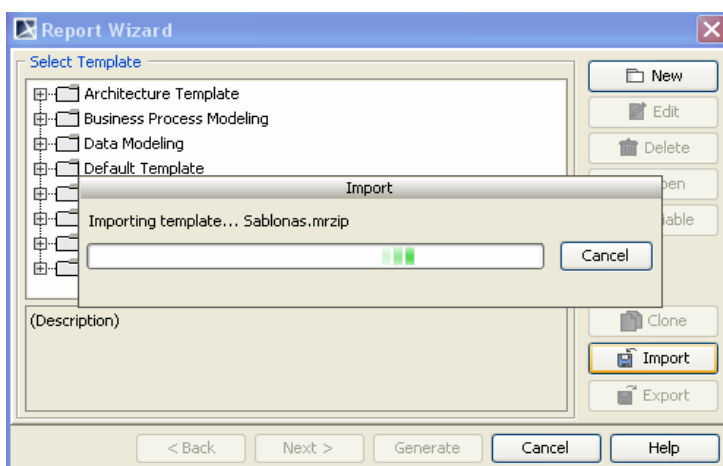
Paspaudus *Import* mygtuką, reikia nurodyti, kur saugojamas ataskaitos šablonas:





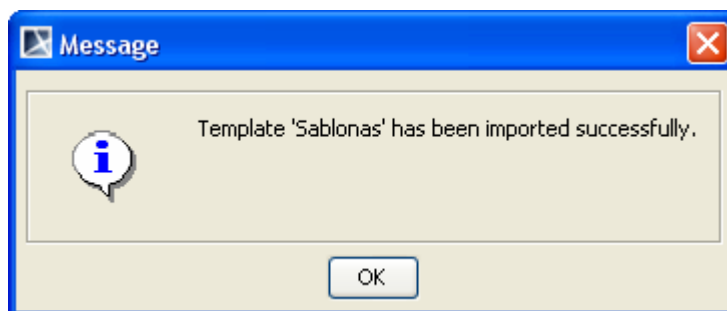
5.3 pav. Importuojamo šablono nurodymas

Paspaudus mygtuką *Open*, vyksta šablono importavimas:



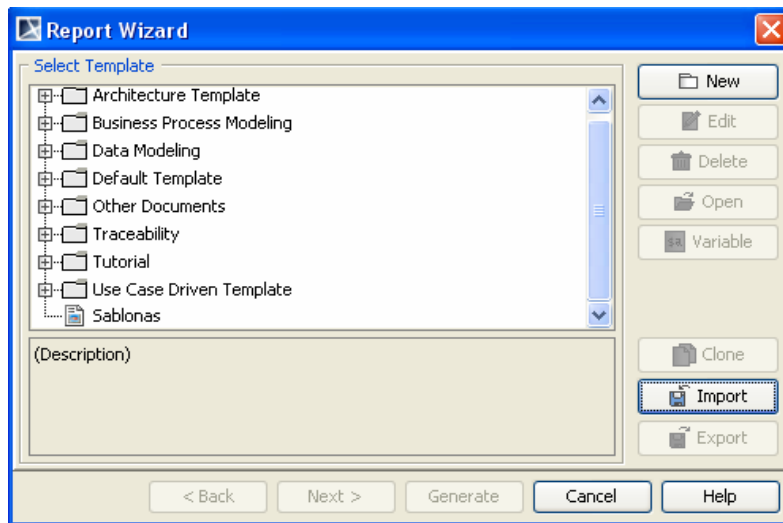
5.4 pav. Šablono importavimas

Pabaigus importavimą, rodomas pranešimas apie sėkmingai suimportuotą šabloną:



5.5 pav. Pranešimas apie sėkmingą šablono importavimą

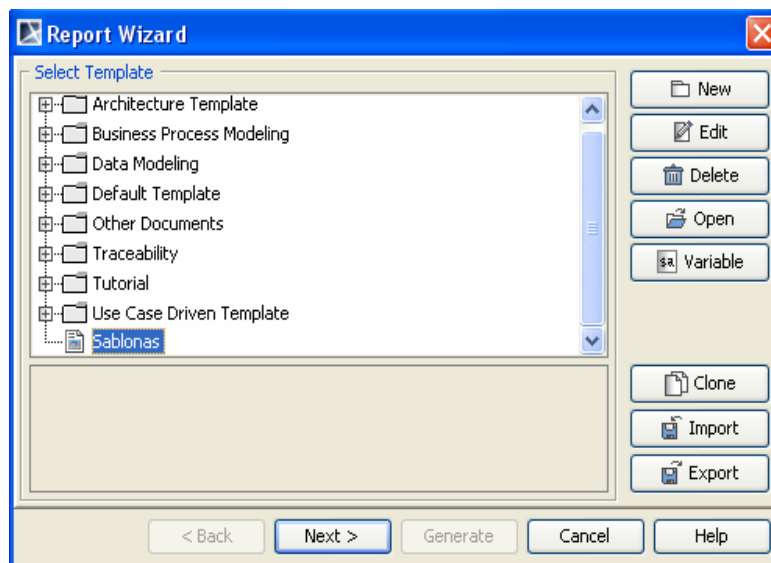
Importavus, šablonas atsiranda *MagicDraw UML* įrankyje:



**5.6 pav.** *Report Wizard* įrankio langas importavus sukurtą šabloną

Šis šablonas bus matomas visiems projektams. Kitą kartą, paleidus *MagicDraw UML*, atskaitos šablono importuoti nereikės.

Norint peržiūrėti šablono turinį, pasižymimas šablonas *ReportWizard* lange ir spaudžiamas mygtukas *Open*:

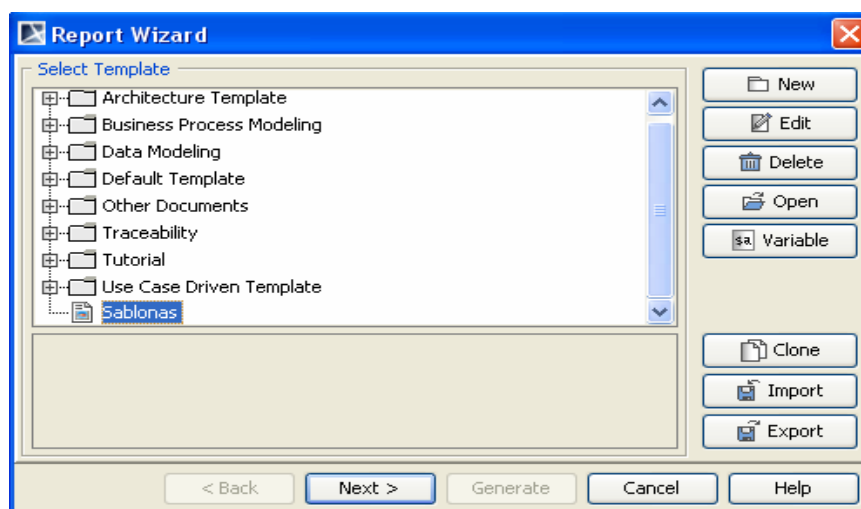


**5.7 pav.** *Report Wizard* įrankio langas, pažymėjus konkretų šabloną

Paprastam vartotojui šablono turinys nėra informatyvus, nes šabloną sudaro VTL sintakse sukurti elementai. Vartotojui aktualus sugeneruotas dokumentas.

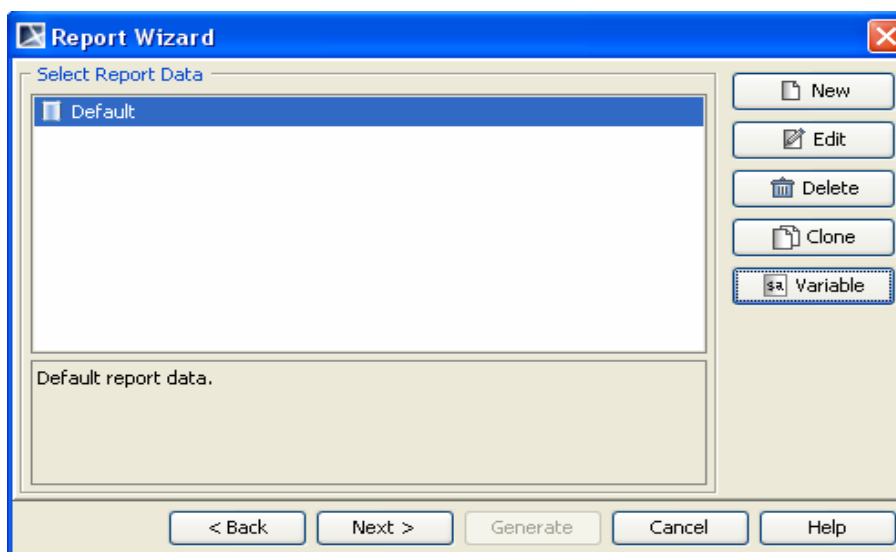
### 5.3. Dokumento generavimas

Dokumento generavimas atliekamas *ReportWizard* lange pasirinkus ataskaitos šabloną bei paspaudus *Next*:



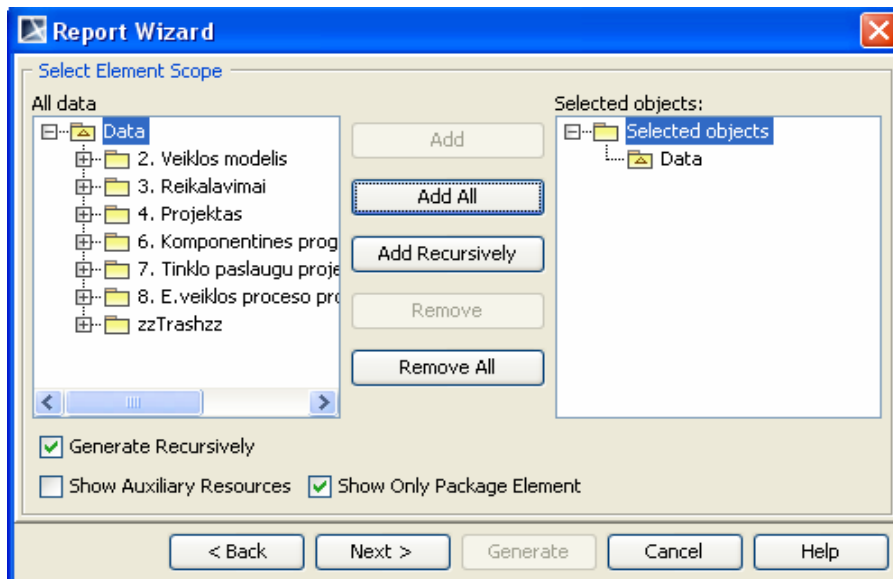
5.8 pav. *Report Wizard* įrankio langas, pažymėjus konkretų šabloną

Dar kartą spaudžiama *Next*:



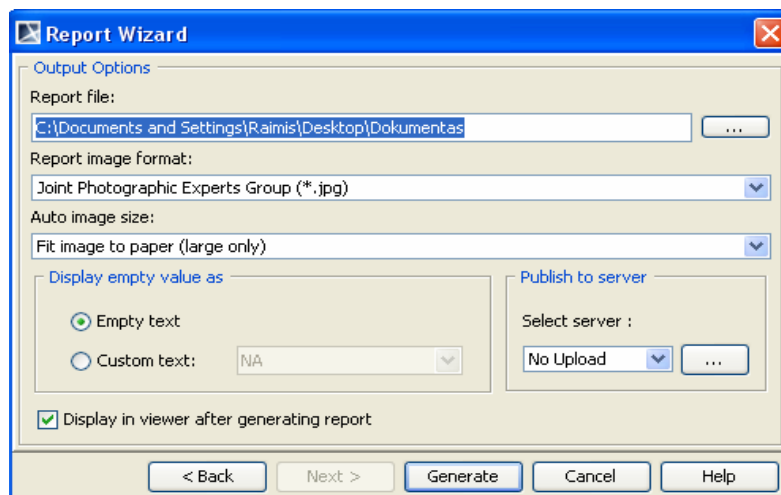
5.9 pav. *Report Wizard* įrankio langas, pasirinkus konkretų šabloną ir paspaudus *Next*

Kitame lange nurodoma, kurias projekto dalis įtraukti į dokumentą. Kadangi dokumentas generuojamas visam projektui, todėl įtraukiamas šaknis projekto katalogas *Data*:



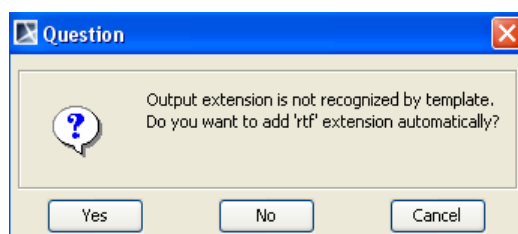
**5.10 pav.** Įtraukiamų į generuojamą dokumentą projekto dalių nurodymas

Nurodžius, kur išsaugoti sugeneruotą dokumentą, spaudžiamas mygtukas *Generate*:



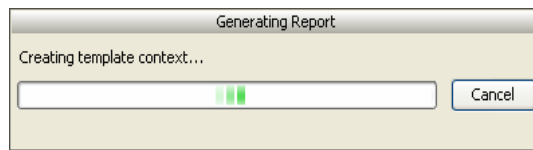
**5.11 pav.** Paskutinis langas prieš dokumento generavimo vykdymą

Išmetamas pranešimas, ar norime, kad prie dokumentas būtų sugeneruotas RTF format. Spaudžiama *Yes*:



**5.12 pav.** Pranešimas apie sugeneruoto dokumento formato nurodymą

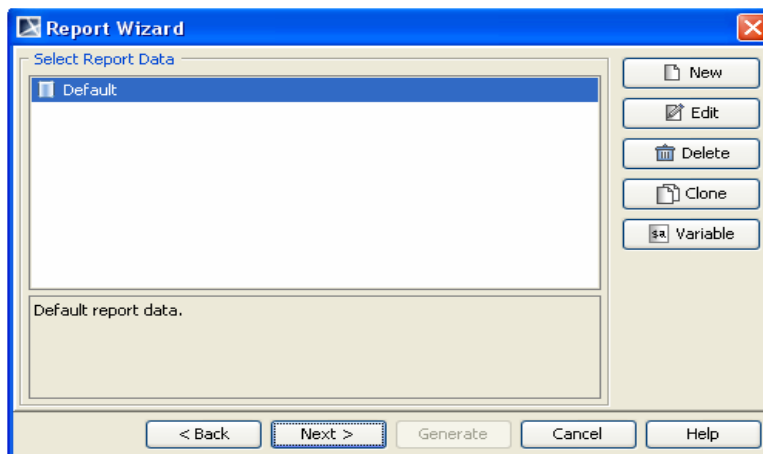
Vyksta dokumento generavimas:



5.13 pav. Būsenos apie dokumento generavimą juosta

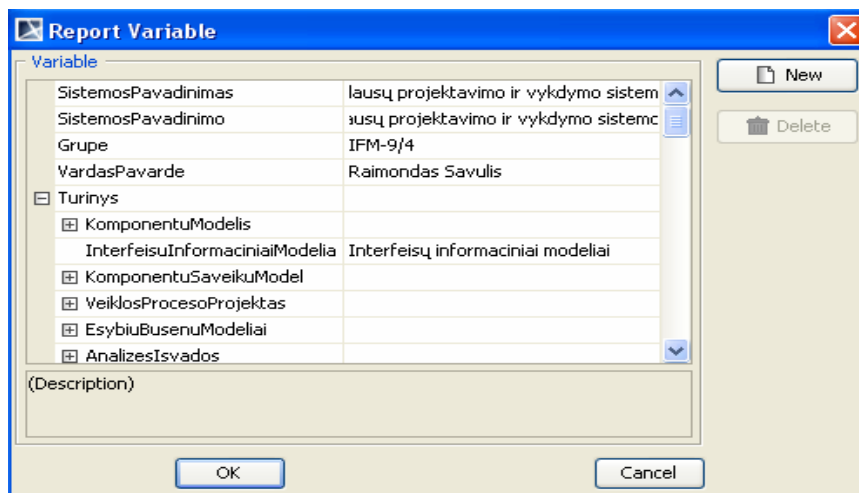
#### 5.4. Dokumento šablono kintamieji

Šablono kintamuosius galima rasti *Report Wizards* įrankyje pasirinkus šablona, bei paspaudus *Next* ir pasirinkus *Variable*:



5.14 pav. *Report Wizard* įrankio langas, pasirinkus konkretų šablona ir paspaudus *Next*

Šablono kintamąjį sudaro jo pavadinimas ir priskirta reikšmė. Šablono kintamojo pavadinimas naudojamas tam, kad, generuojant dokumentą, dokumentų generavimo variklis žinotų, kokią reikšmę įterpti į dokumentą.



5.15 pav. Sukurto šablono kintamieji

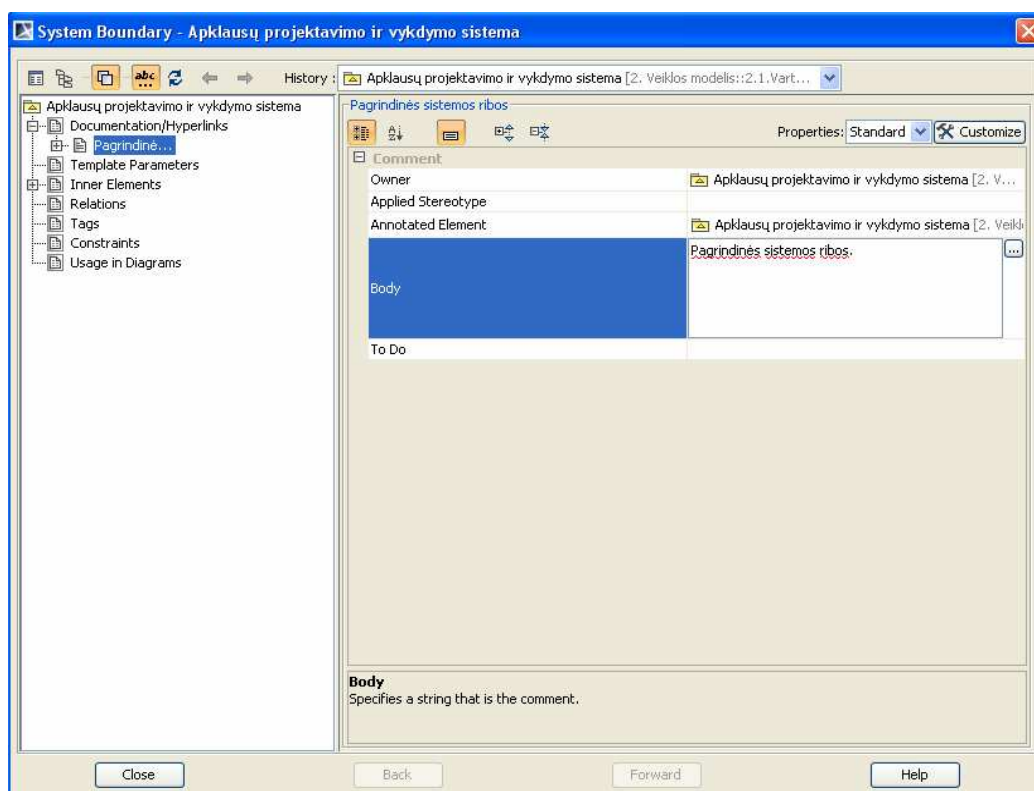
Tarkime, turime šablono kintamąjį „VardasPavarde“ su jam priskirta reikšme „Raimondas Savulis“. Tam, kad įterpti šablono kintamojo reikšmę į dokumentą, šablone įterpiamas simbolis \$ ir rašoma kintamojo pavadinimas: \$VardasPavarde.

Sugeneravus dokumentą vietoje „\$VardasPavarde“ bus sugeneruota „Raimondas Savulis“.

Labai svarbu nesuklysti ir šablone teisingai nurodyti kintamųjų pavadinimus – dėl vienos nesutampančios raidės nebus įterpta kintamojo reikšmė. Be to, nenaudotinos lietuviškos raidės kintamųjų varduose.

Galima sukurti šablono kintamųjų hierarchiją. Tarkime, turime pagrindinį kintamąjį „Turinys“ bei jo vidinį kintamąjį „Ivadas“. Tam, kad būtų sugeneruota kintamojo „Ivadas“ reikšmė, nurodoma pagrindinio kintamojo pavadinimas, dedamas taškas „.“ ir įterpiama dominančio kintamojo reikšmė: \$Turinys.Ivadas .

Projektuojant *MagicDraw* įrankio pagalba, rekomenduojama sudėti elementų komentarus. Elementų komentarai įrašomi elemento *Specification* lange užpildžius *Documentation/Hyperlinks* savybės *Body* dalį:

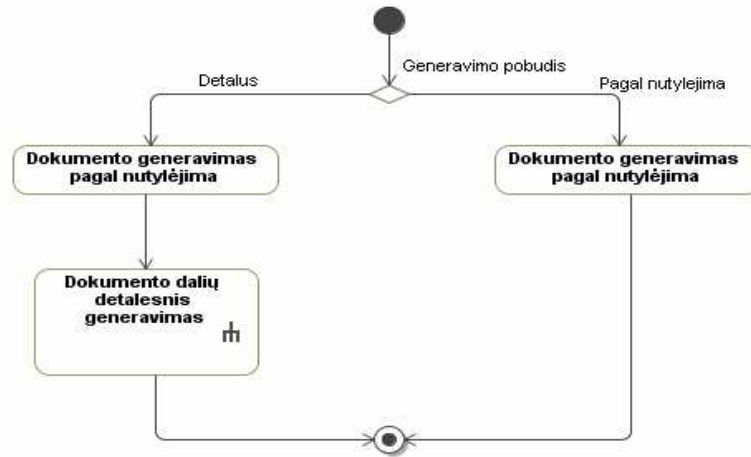


5.16 pav. Elemento komentarų pildymas

Tokiu būdu, sugeneravus dokumentą, nereikės pildyti elementų aprašymų, nes jie bus automatiškai įtraukiami į dokumentą generavimo metu.

Taip pat rekomenduojama užpildyti diagramų aprašymus *MagicDraw* įrankyje, nes dokumento šablone realizuotas diagramų aprašymų įtraukimas į sugeneruotą dokumentą.

Projekto dokumento generavimo, panaudojant dokumentavimo sistemos šabloną, eiga matoma 5.17 ir 5.18 paveiksluose.



**5.17 pav.** Projektų dokumento generavimo, panaudojant projektų dokumentavimo sistemos šabloną, eiga

5.18 paveiksle matomas detalesnis veiklos „Dokumento dalių detalesnis generavimas“ vaizdas.



5.18 pav. Detalesnis veiklos „Dokumento dalių detalesnis generavimas“ vaizdas



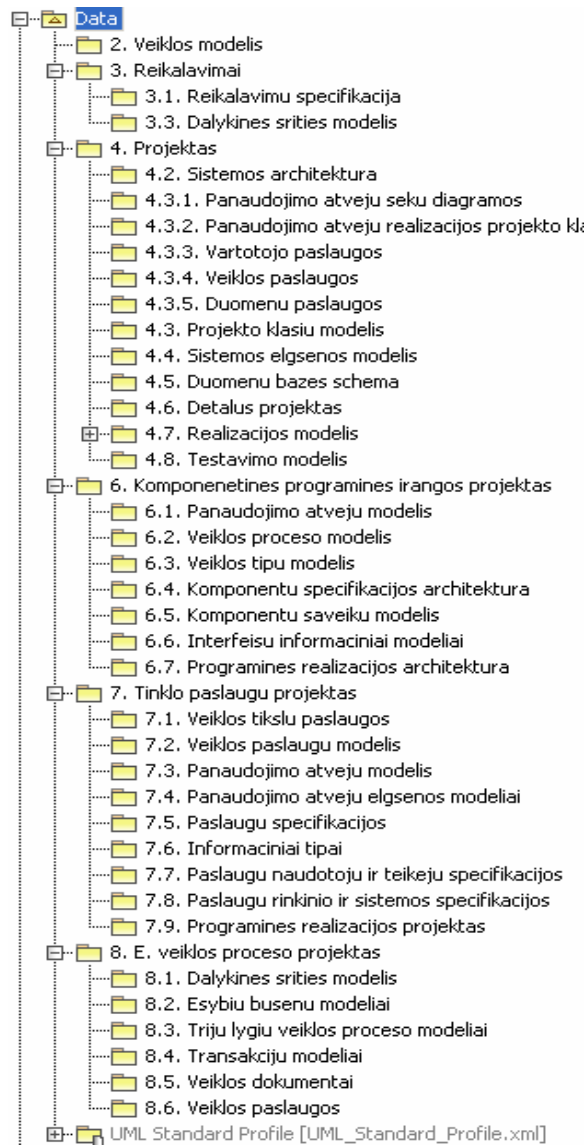
Kaip matyti 5.17 pav. ir 5.18 pav., panaudojant sukurtą šabloną, galima sugeneruoti dviejų tipų dokumentus – pagal nutylėjimą ir detalų. Detalus dokumentas nuo dokumento, sugeneruoto pagal nutylėjimą, skiriasi tuo, kad kai kurios dokumento vietos labiau detalizuojamos bei į dokumentą įtraukiami papildomi skaičiavimai, tokie kaip programavimo darbų apimtys skaičiavimas, testavimo darbų apimtys skaičiavimas bei projekto elementų ryšių pilnumo nustatymas.

## **5.5. Projektų dokumentavimo sistemos šablonas**

Tam, kad projektų dokumentavimo sistemą būtų galima panaudoti įvairesniuose projektuose, buvo sukurti du dokumentų šablonai – šablonas, skirtas projektams su fiksuota (pastovia) struktūra, ir projektams, kurių struktūra projektavimo eigoje kinta.

### **5.5.1 Šablonas su fiksuota projekto struktūra**

Sukurtas projektų dokumentavimo sistemos šablonas susietas su pavyzdinio projekto katalogų struktūra – pagal projekto katalogų pavadinimus nustatoma, kurio projekto katalogo duomenis įtraukti į konkrečią dokumento vietą.



5.19 pav. Pavyzdinio projekto katalogų struktūra

Tarkime, turime tokį fragmentą iš šablono:

```
#foreach ($p in $report.findElementByName($Package , "3.1.
Reikalavimu specifikacija"))
#if ($p.ownedDiagram != "" && $p.ownedDiagram != [])
#foreach($d in $sorter.sort($p.ownedDiagram, "name"))
    $d.image
    $d.name
#end
#end
```

Sugeneravus, vietoje šio fragmento būtų įterpta visos diagramos iš katalogo „3.1. Reikalavimu specifikacija“. Kaip matyti 5.19 paveiksle, katalogo pavadinimas šablone sutampa su katalogo pavadinimu pavyzdinio projekto *MagicDraw UML* įrankyje. Todėl, norint panaudoti pavyzdinį šabloną kitam projektui, reikia arba suderinti naujo projekto ir pavyzdinio projekto katalogų struktūrą, arba pakoreguoti katalogų pavadinimus šablone.

Nors, norint panaudoti šį šabloną su skirtinga nei pavyzdinės sistemos katalogų struktūra, tektų pakoreguoti šablono programinį kodą, tačiau toks generavimo būdas turi nemažai privalumų, palyginus su šablonu, kuris skirtas projektams su kintančia struktūra. Visų pirma, kadangi, generuojant dokumentus su *MagicDraw UML*, ribotos teksto redagavimo galimybės, todėl tas dokumento dalis, kurių tekstui reikia specifinio formatavimo, galima įrašyti tiesiai į šabloną ir atitinkamai suformatuoti – generuojant dokumentą, šios vietos bus praleidžiamos ir nekoreguojamos. Antra, šis šablonas nėra taip stipriai susietas su projektavimo įrankiu, todėl galima lengviau koreguoti dokumento struktūrą – pridėti naujus skyrius, į kuriuos nereikia įtraukti duomenų iš projektavimo įrankio, sukeisti dokumento skyrius vietomis ir kt.

Be to, tos dokumento dalys, kurios tekstinės - įvadas, išvados, reikalavimų specifikavimas - ir kurių neprasminga įtraukinėti į *MagicDraw UML*, galima įrašyti tiesiai į dokumento šabloną – dokumento generavimo variklis tas dalis praleidžia jų nekeisdamas. Todėl tokias dokumento dalis galima formatuoti pagal poreikius ir norus.

Tam, kad nereikėtų papildomai koreguoti sukurtojo šablono, projektuojant *MagicDraw UML* įrankiu, rekomenduojama 3 lygių katalogų struktūra, nes tik tokiam lygyje realizuota diagramų paieška. Tačiau, jei reikalinga daugiau nei 3 lygių katalogų struktūra, atlikus nesudėtingus pakeitimus šablone, funkcionalumas bus išplėstas. Tarkime, turime dviejų lygių katalogų struktūrą:

```
#foreach ($p in $report.findElementByName($Package , "Katalogas"))
#if ($p.ownedDiagram != "" && $p.ownedDiagram != [])
#foreach($d in $sorter.sort($p.ownedDiagram, "name"))
    $d.image
    $d.name

    $d.documentation
#end
#end
#foreach($i in $sorter.sort($report.getInnerElement($p), "name"))
#if ($i.ownedDiagram != "" && $i.ownedDiagram != [])
#foreach($d in $sorter.sort($i.ownedDiagram, "name"))
    $d.image
    $d.name

    $d.documentation
#end
#end
#end
#end
```

Šioje ištraukoje iš šablono realizuota diagramų paieška *Package* elemente, pavadinimu „Katalogas“, bei jo vaikiniuose elementuose (2 lygiai.)

Tam, kad įtraukti papildomą katalogo lygį, tereikia nukopijuoti 2-ojo lygio kodo gabalą, sukuriant naują ciklą, bei atlikti minimalius pakeitimus cikle:

```
#foreach ($p in $report.findElementByName($Package , "Katalogas"))
#if ($p.ownedDiagram != "" && $p.ownedDiagram != [])
#foreach($d in $sorter.sort($p.ownedDiagram, "name"))
    $d.image
    $d.name
    $d.documentation
#end
#end
#foreach($i in $sorter.sort($report.getInnerElement($p), "name"))
#if ($i.ownedDiagram != "" && $i.ownedDiagram != [])
#foreach($d in $sorter.sort($i.ownedDiagram, "name"))
    $d.image
    $d.name
    $d.documentation
#end
#end
#end

#foreach($i in $sorter.sort($report.getInnerElement($i), "name"))
#if ($i.ownedDiagram != "" && $i.ownedDiagram != [])
#foreach($d in $sorter.sort($i.ownedDiagram, "name"))
    $d.image
    $d.name
    $d.documentation
#end
#end
#end

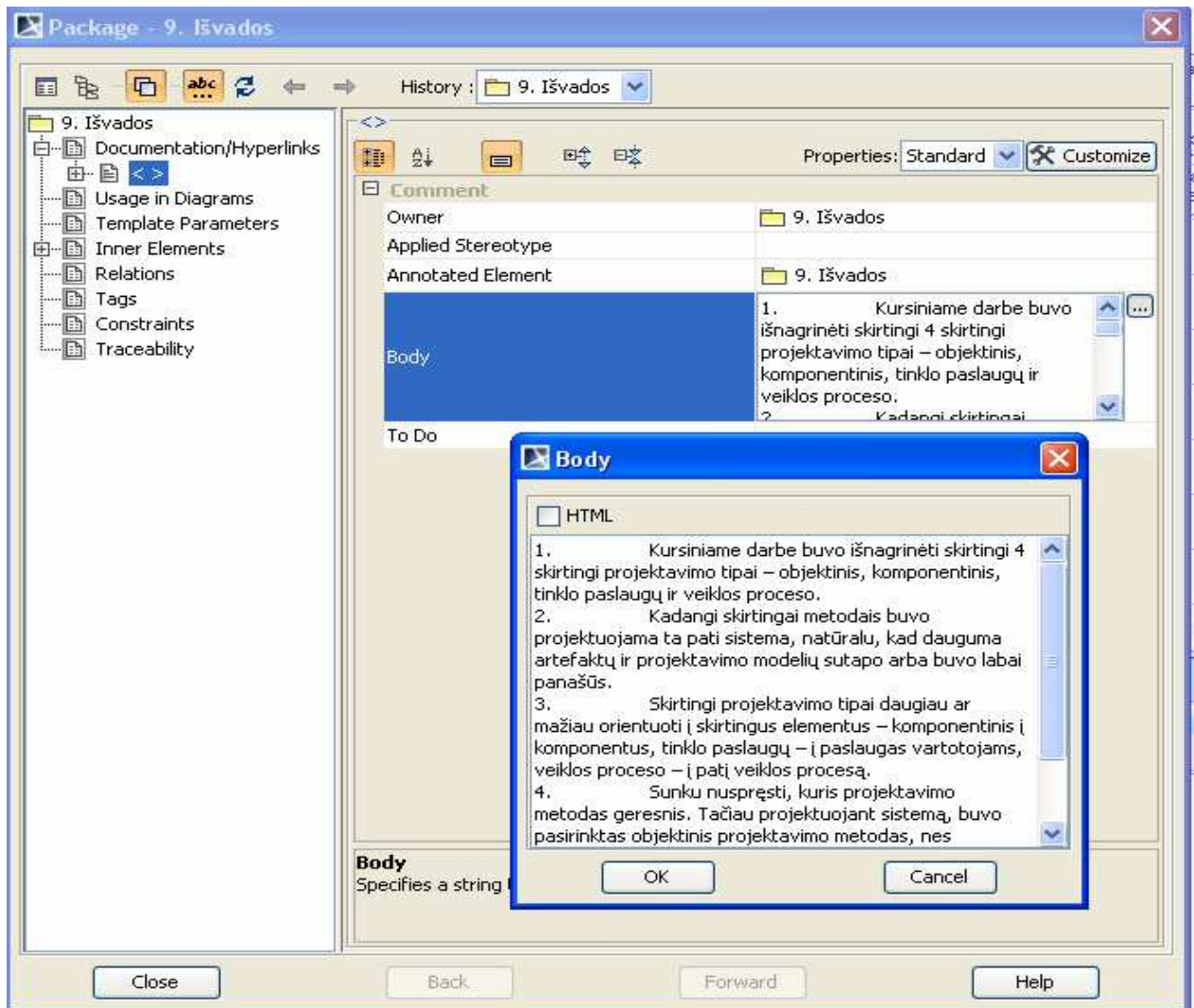
#end
#end
```

Taip galima sukurti tiek katalogų lygių, kiek reikia pagal situaciją.

### 5.5.2 Šablonas projektams su kintančia struktūra

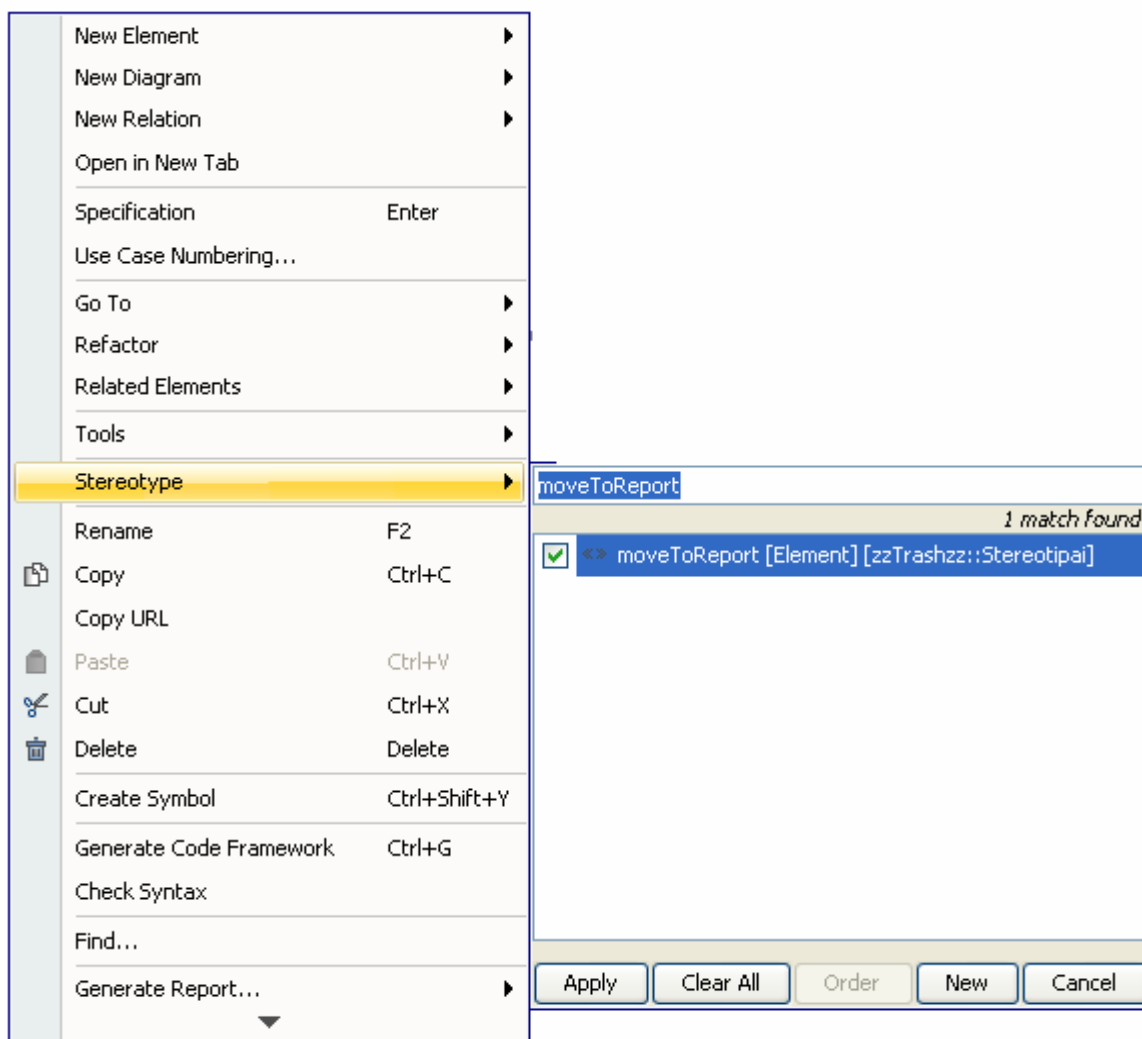
Tais atvejais, kai projekto struktūra nėra pastovi, pageidautina, kad dokumento šablonas nebūtų susietas su konkrečia projekto struktūra – tai yra, kad būtų galima generuoti dokumentaciją nekoreguojant šablono. Dėl šios priežasties buvo sukurtas šablonas, skirtas projektams su kintančia struktūra.

Dokumento struktūra sudaroma pagal projekto katalogų struktūrą – t.y. katalogų pavadinimai atitinka dokumento skyrių pavadinimus. Todėl tam, kad būtų galima sugeneruoti visą dokumento struktūrą, gali reikėti sukurti tuščius katalogus, kurie bus naudojami tik tam, kad atvaizduotų kokią nors dokumento dalį, tarkime, įvadą ar išvadas. Tokių dokumento dalių tekstas turėtų būti įvedamas kaip atitinkamo katalogo komentaras (5.20 paveikslas).



**5.20 pav.** Projekto katalogo komentarų pildymas

Tikėtina, kad projektavimo įrankyje reikės panaudoti katalogus, kurie bus pagalbinaiai ir kurių nereikės įtraukti į dokumentą kaip atskiros jo dalies. Todėl tiems katalogams, kurių pavadinimai sudaro dokumento struktūrą, reikia priskirti naujai susikurtą stereotipą „moveToReport“ (5.21 paveikslas), pagal kurį nusprendžiama, ar katalogo pavadinimą įtraukti kaip dokumento skyriaus pavadinimą, ar neįtraukti.



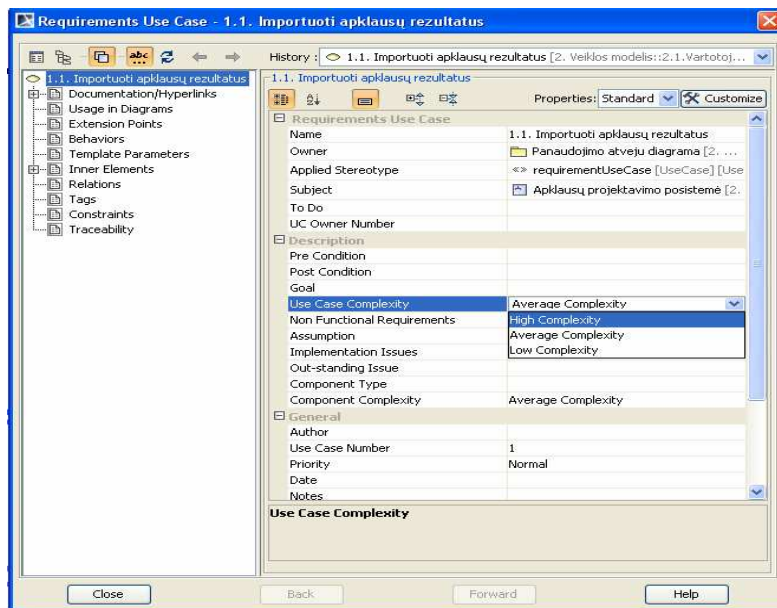
**5.21 pav.** Stereotipo „moveToReport“ priskyrimas katalogui, kurio pavadinimas atitinka dokumento skyriaus pavadinimą

Šio šablono naudojimas paprastesnis, nes nereikalaujama šablono programinio kodo koregavimo įgūdžių. Tačiau, naudojant šį šabloną, lieka mažiau galimybių koreguoti dokumento struktūrą – nėra galimybės į dokumentą įtraukti papildomų skyrių, nesukuriant atitinkamų katalogų projektavimo įrankyje. Be to, dokumento dalių pavadinimai tiesiogiai susiję su projekto katalogų pavadinimais.

## 5.6. Projekto dokumentavimo ir testavimo darbų apimties skaičiavimo realizacija

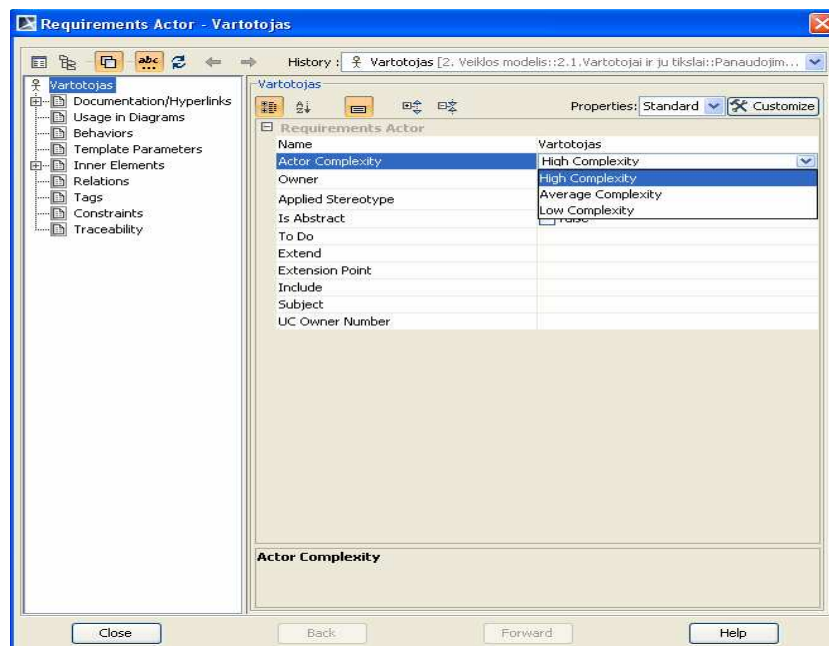
Programavimo ir testavimo apimčių skaičiavimas vyksta įvertinant panaudojimo atvejams ir aktoriams priskirtų svorinių verčių reikšmes bei įvertinant techninius bei aplinkos faktorius [13][14].

Panaudojimo atvejų svarinės reikšmės priskiriamos konkretaus panaudojimo atvejo *Specification* lange nurodžius sudėtingumo lygį *Use Case Complexity: Low Complexity, Average Complexity* arba *High Complexity*:



5.22 pav. Panaudojimo atvejo svarinės vertės nustatymas

Aktorius sudėtingumo lygis nurodomas identiškai:



5.23 pav. Aktoriaus svarinės vertės nustatymas

Šios svarinės reikšmės naudojamos nustatinėjant neapskaičiuojamų panaudojimo atvejų taškų UUCP (*unadjusted use case points*) reikšmę.

Kiekvieno panaudojimo atvejo sudėtingumo lygio svarinė vertė apskaičiuojama taip:

Priskirta sudėtingumo reikšmė	Svorinė vertė
<i>Low Complexity</i>	5
<i>Medium Complexity</i>	10
<i>High Complexity</i>	15

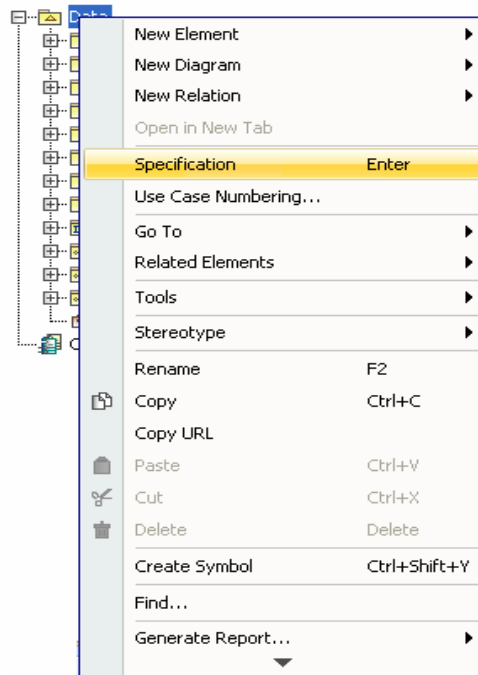
Aktoriaus sudėtingumo lygio svorinė vertė:

Priskirta sudėtingumo reikšmė	Svorinė vertė
<i>Low Complexity</i>	1
<i>Medium Complexity</i>	2
<i>High Complexity</i>	3

Tarkime, kad  $U_i$  yra vieno panaudojimo atvejo sudėtingumo svorinė vertė, o  $A_j$  – vieno aktoriaus sudėtingumo svorinė reikšmė. Tuomet UUCP apskaičiuojama taip:

$$UUCP = \sum U_i + \sum A_j, \text{ kur } i = 1..n, j = 1..m.$$

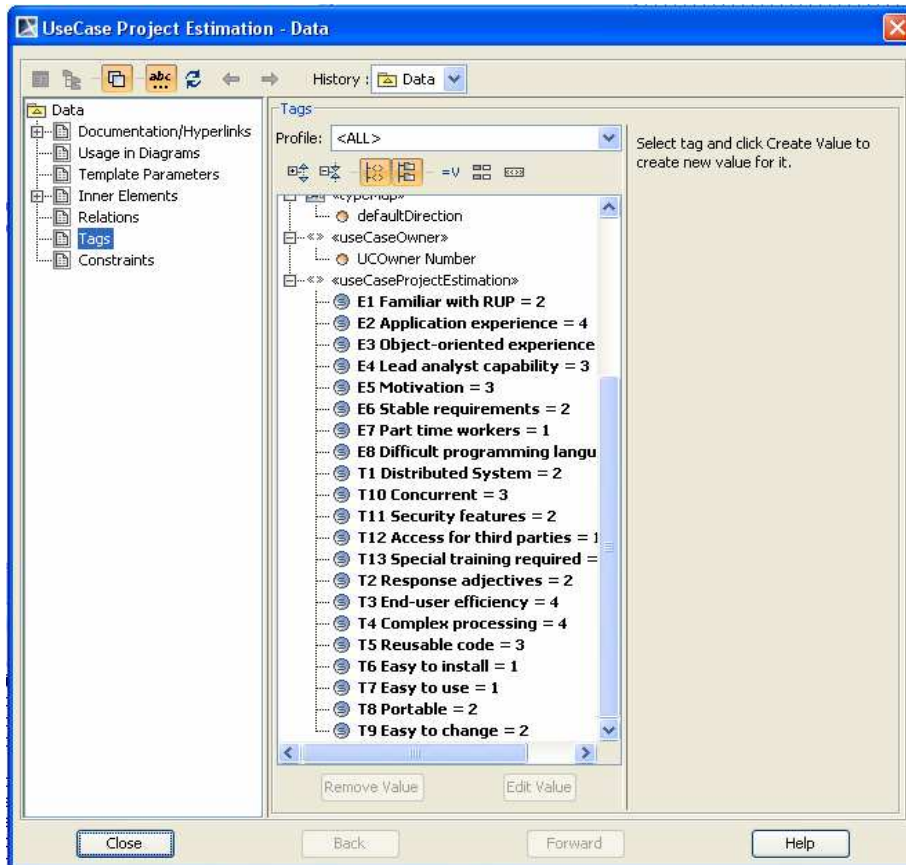
Skaičiuojant programavimo apimtis, vertinamų projekto techninių ir aplinkos faktorių reikšmės priskiriamos paspaudus dešinį pelės klavišą ant šakninio projekto katalogo *Data* bei pasirinkus *Specification*:



5.24 pav. Projekto bendrosios informacijos redagavimas

Kitame lange reikia pasirinkti *Tags*:

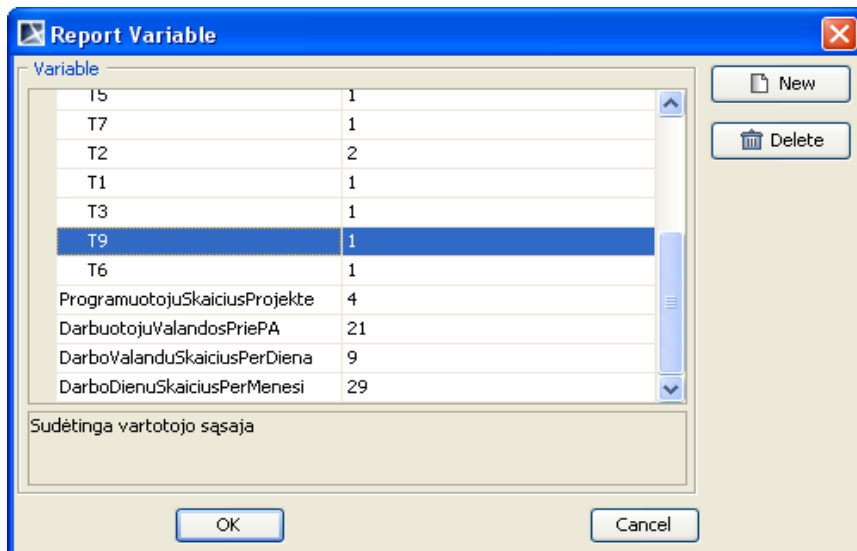




5.25 pav. Projekto techninių ir aplinkos faktorių reikšmių redagavimas

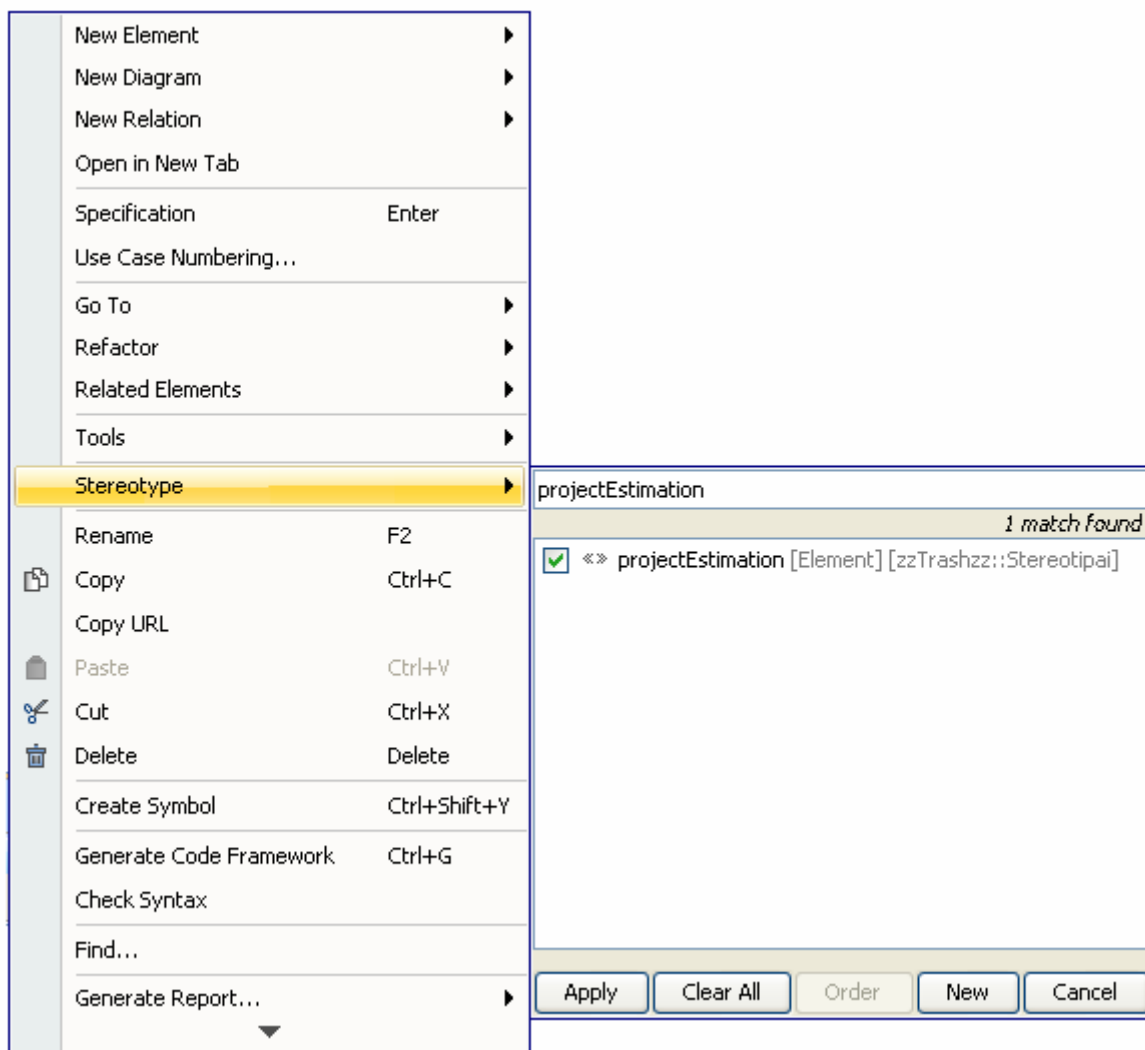
Aplinkos faktoriai žymimi E1-E8, techniniai faktoriai T1-T13.

Skaičiuojant testavimo apimtį, techniniai faktoriai skiriasi, o aplinkos faktoriai yra bendri. Testavimo apimtį skaičiavimo techniniai faktoriai nurodomi *Report Wizard* įrankyje pasirinkus sukurtą šabloną, paspaudus *Next* bei paspaudus *Variable*. Faktorių reikšmės galima keisti išskleidus kintamąjį „TestavimoVertinimas“ ir koreguojant kintamųjų T1-T9 reikšmes. Be to, šiame lange koreguojami papildomų kintamųjų, tokių kaip programuotojų skaičius, numatomos darbuotojų valandos prie panaudojimo atvejo, darbo valandų skaičius per dieną ir darbo dienų skaičius per mėnesį, reikšmės, kurios taip pat įtakoja programavimo ir testavimo apimtį.



**5.26 pav.** Projekto bendrosios informacijos redagavimas

Todėl, kad projekte gali būti keletas panaudojimo atvejų diagramų, todėl būtina nurodyti, kuri iš jų naudojama projekto darbų įvertinimui. Tai atliekama sukuriant naują stereotipą „projectEstimation“ ir ji priskiriant katalogui, kuriame saugomi panaudojimo atvejai, kurių svarinės vertės naudojamos įverčiui skaičiuoti (5.27 paveikslas).



**5.27 pav.** Naujo stereotipo „projectEstimation“ priskyrimas katalogui

## 5.7. Projekto elementų ryšių pilnumo nustatymo realizacija

Projekto elementų ryšių pilnumo nustatymas įvertinamas analizuojant, ar, pereinant nuo skirtingų projekto kūrimo etapų, t.y. nuo reikalavimų prie analizės, nuo analizės prie realizavimo, visi projekto elementai perkeliami į kitą etapą ir ar pilnai realizuoti. Tikrinami tokie ryšiai:

- ar panaudojimo atvejai realizuoti projekto klasėmis;
- ar panaudojimo atvejai iš reikalavimų identifikavimo realizuoti *Collaboration* elementais;
- ar *Collaboration* elementai susieti *Dependency* ryšiu su klasėmis iš reikalavimų analizės;
- ar *Collaboration* elementai realizuoti klasėmis;
- ar komponentai realizuoti artefaktais;
- ar komponentai realizuoti klasėmis.

Tarkime, reikalavimų identifikavimo metu buvo sudarytas toks veiklos modelis:



5.28 pav. Reikalavimų identifikavimo metu sudaryto veiklos modelio pavyzdys

Pereinant nuo reikalavimų identifikavimo prie analizės, sudarytas veiklos modelis turi būti perkeltas į analizės dalį, t.y. panaudojimo atvejai turi būti realizuoti *Collaboration* tipo elementais. Tarkime, per klaidą projektuotojas neperkėlė panaudojimo atvejo „6.2. Prisijungti prie sistemos“ iš reikalavimų identifikavimo į analizę ir turime tokią situaciją, kad tik du panaudojimo atvejai iš trijų buvo realizuoti *Collaboration* tipo elementais:



5.29 pav. Perėjimas nuo reikalavimų identifikavimo prie analizės

Tuomet, sugeneravus dokumentą, jei bus generuojama detalus dokumentas ir projektuotojas nuspręs patikrinti elementų ryšių pilnumą, tuomet bus sugeneruota tokia dokumento dalis:

Projekto elementų ryšių pilnumo tikrinimas	
Panaudojimo atvejis, nerealizuotas <i>Collaboration</i> elementu:	
Elemento pavadinimas	Tėvinis elementas
🕒 6.2. Prisijungti prie anketos	1.1. Panaudojimo atveju modelis

5.30 pav. Dokumento fragmentas, kuris sugeneruojamas aptikus trūkstamą ryšį tarp elementų

## 6. Eksperimentinis Projektų dokumentavimo sistemos tyrimas

### 6.1. Eksperimento apibrėžimas

Eksperimento tikslas – išanalizuoti Projektų dokumentavimo sistemą, siekiant nustatyti realią Projektų dokumentavimo sistemos teikiamą naudą projektuotojo požiūriu IS projektų rengimo procese. Naudą planuojama įvertinti palyginant įprastą dokumentacijos ruošimo būdą (kai dokumentacija kuriama rankiniu būdu teksto redagavimo priemonėmis) su dokumentacijos ruošimu, panaudojant Projektų dokumentavimo sistemos metodiką.

### 6.2. Eksperimento planas

Eksperimentą numatoma vykdyti trimis etapais. Etapai aprašyti 6.1 lentelėje.

#### 6.1 lentelė. Eksperimento vykdymo etapai

Etapas	Aprašymas
1. Vartotojų apklausa	Numatomiems metodikos naudotojams išdalinamos anketos, kuriose prašoma įvertinti projekto kūrimo ir dokumentavimo darbų apimtį.
2. Metodikos pritaikymas vartotojų projektams	Apklaustųjų vartotojų projektams, panaudojant sukurta metodiką, sugeneruojama dokumentacija, užfiksuojant laiką, kuris sugaištamas šiai dokumentacijai paruošti.
3. Dokumentacijos ruošimo būdų palyginimas	Įvedamos faktinės projekto vykdymo laiko vertės.

**Eksperimento objektas** – Projektų dokumentavimo sistema.

**Eksperimento subjektas** yra asmuo, atsakingas už projekto dokumentacijos rengimą. **Eksperimento atveju** – tai bakalaurnio darbo vykdytojai.

**Nepriklausomi kintamieji** – eksperimente naudojami septyni bakalauro darbų projektai, sukurti *MagicDraw UML* įrankiu.

**Priklausomi kintamieji** – tai laikas, kurį sugaišo magistrinių ir bakalauro darbų autoriai kurdami ir dokumentuodami savo darbus, bei laikas, kuris buvo sugaištas ruošiant dokumentaciją atitinkamam projektui, panaudojant Projektų dokumentavimo sistemos metodiką.

### 6.3. Eksperimento vykdymas

Eksperimento metu dalyviams buvo pateikta anketa, kurioje prašoma nurodyti, kiek projekto atlikta, kiek laiko buvo planuota sukurti visą projektą, kiek laiko sugaišta projekto kūrimui, per kiek laiko buvo planuota sukurti projekto dokumentaciją, kiek laiko sugaišta projekto dokumentacijos rengimui, koku būdu

ruošiama dokumentacija (rankiniu ar generuojant) ir ar dalyvis domėjosi galimybe generuoti dokumentaciją tiesiai iš projektavimo įrankio (nes būtent šia savybe pasižymi Projektų dokumentavimo sistema).

Taip pat eksperimento dalyvių buvo paprašyta pateikti savo projektus, sukurtus *MagicDraw UML* įrankiu, kad būtų galima paruošti dokumentaciją naudojant Projektų dokumentavimo sistemos metodiką ir įvertinti, ar metodikoje suformuluoti darbo principai tikrai efektyvūs.

Sugeneravus dokumentaciją, galima palyginti skirtingus dokumentacijos ruošimo būdus, įvertinti, kiek sumažėja laiko sąnaudos projekto dokumentacijai ruošti, todėl paspartėja viso projekto kūrimas.

#### **6.4. Eksperimento interpretavimas ir pateikimas**

Eksperimente buvo naudojami 8 bakalaurinių darbų projektai (6.2 lentelė.). Toliau eksperimente projekto numeris atitinka 6.2 lentelėje esantį autorių. Tik 2 iš 8 eksperimento dalyvių domėjosi galimybe generuoti dokumentaciją tiesiai iš projektavimo įrankio. Visų projektų dokumentacija buvo ruošiama rankiniu būdu.

**6.2 lentelė.** Eksperimente naudotų projektų autoriai

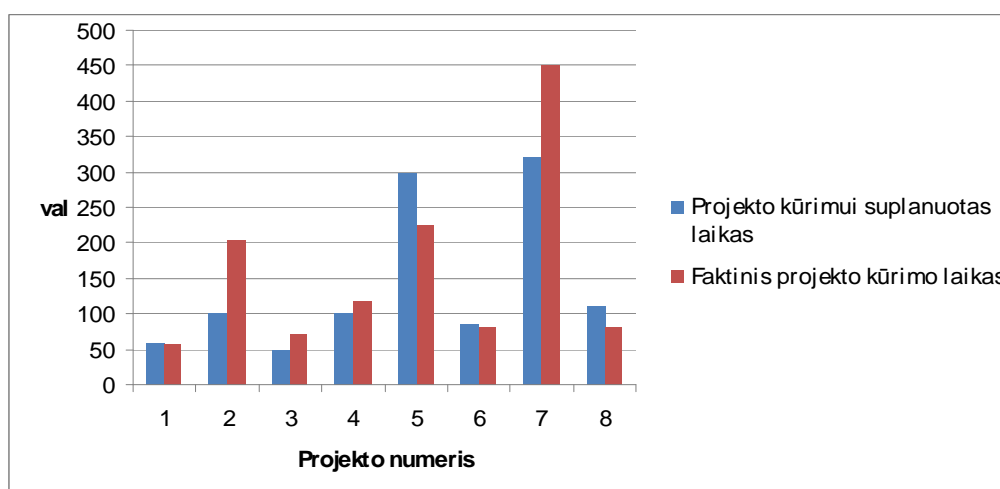
	<b>Autorius</b>
1.	Aistė Pilvinytė
2.	Aurimas Gudas
3.	Edvardas Pranckevičius
4.	Paulius Dunauskas
5.	Paulius Gaidamavičius
6.	Tomas Skroblas
7.	Vilmantas Grinius
8.	Kasparas Liutkevičius

Kaip matyti 6.3 lentelėje, net 4 iš 8 projektų tiek kūrimo, tiek dokumentacijos rengimo darbai užtruks ilgiau nei planuota. Jei pavyktų paspartinti dokumentacijos rengimo procesą, būtų galima tikėtis, kad šiuos projektus būtų galima sukurti ir paruošti jų dokumentaciją laiku.

### 6.3 lentelė. Eksperimento dalyvių projektų darbų įvertinimai

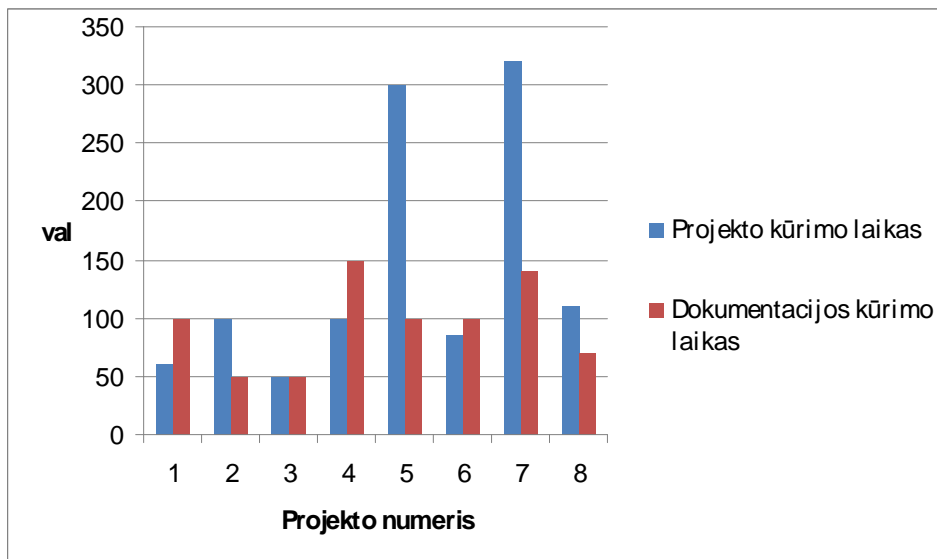
		1	2	3	4	5	6	7	8
1.	Kiek projekto darbų atlikta (%)?	85	65	80	70	50	85	50	85
2.	Per kiek laiko (valandomis) buvo planuota sukurti visą bakalauro darbo projektą?	60	100	50	100	300	85	320	110
3.	Kiek faktiškai laiko (valandomis) iki dabar buvo sugaišta sistemos projekto kūrimui?	50	150	60	90	150	70	300	70
4.	Apskaičiuotas faktinis projekto laikas.	57,5	202,5	72	117	225	80,5	450	80,5
5.	Per kiek laiko (valandomis) buvo planuota sukurti visas bakalauro darbo projekto ataskaitas?	100	50	50	150	100	100	140	70
6.	Kiek faktiškai laiko (valandomis) iki dabar sugaišote bakalauro darbo ataskaitų kūrimui?	70	80	60	90	90	88	120	50

6.1 paveiksle grafiškai pateikiama projekto kūrimui suplanuoto laiko ir apskaičiuoto faktinio laiko, kurio reikės projektui sukurti, palyginimas. Pauliaus Gaidamavičiaus ir Vilmanto Griniaus projektų kūrimui suplanuota daugiausiai laiko, tačiau, įvertinus atliktų darbų kiekį, Paulius turėtų spėti sukurti projektą greičiau, nei buvo planuota, tuo tarpu Vilmantas, jei ir toliau dirbs panašiu našumu, viršys planuotą laiką daugiau nei 100 valandų.



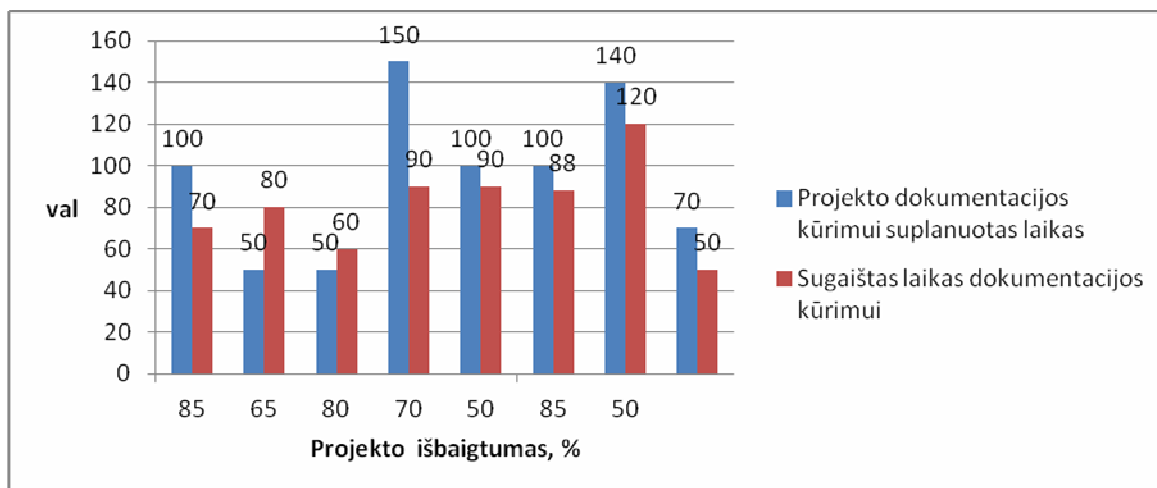
6.1 pav. Eksperimente naudojamų projektų kūrimui suplanuoto ir faktinio laikų palyginimas

6.2 paveiksle palyginta projekto kūrimo ir dokumentavimo darbų trukmės palyginimas - 3 projektų dokumentacijos kūrimo darbų planuojama trukmė yra didesnė nei paties projekto kūrimo darbų trukmė, todėl galima daryti išvadą, kad tam tikruose projektuose dokumentacijos ruošimas yra vienas iš pagrindinių veiksnių, lemiančių viso projekto trukmę.



**6.2 pav.** Projektų kūrimo ir dokumentavimo darbų trukmės palyginimas

Įvertinus dokumentacijos kūrimui sugaištą laiką su laiku, kurį buvo planuojama paskirti, dviejų projektų darbai užtruks ilgiau, nei buvo planuota. Papildomai įvertinus projekto išbaigtumą (6.3 paveikslas), akivaizdu, kad dar dviejų projektų dokumentacijos ruošimo darbai užtruks ilgiau, nei buvo planuota.



**6.3 pav.** Projekto dokumentacijos darbų planuojamos trukmės ir sugaišto laiko palyginimas

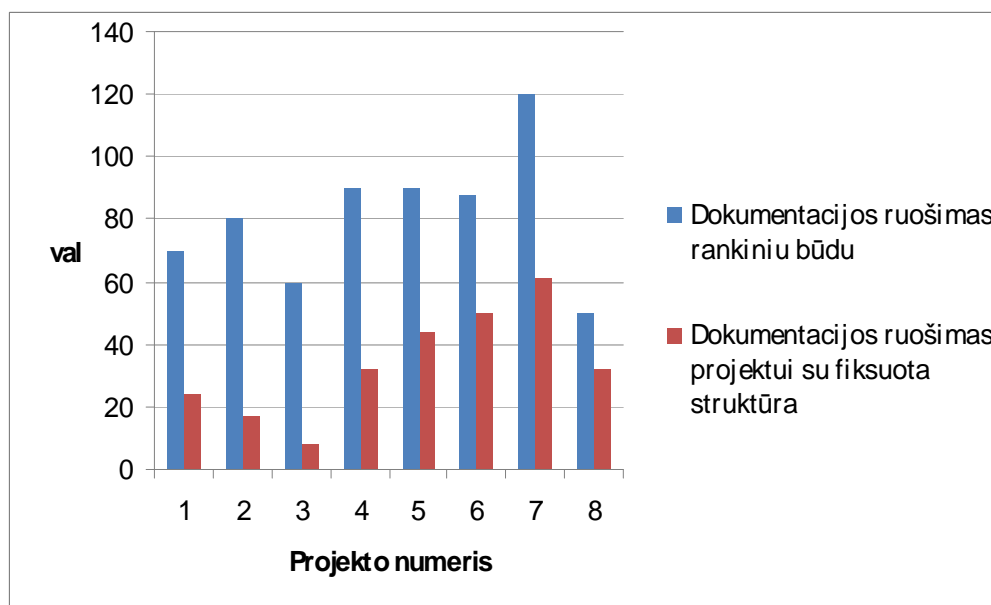


Laikas, sugaištas ruošiant kiekvieno projekto dokumentaciją pagal Projektų dokumentavimo sistemą, matomas 6.4 lentelėje.

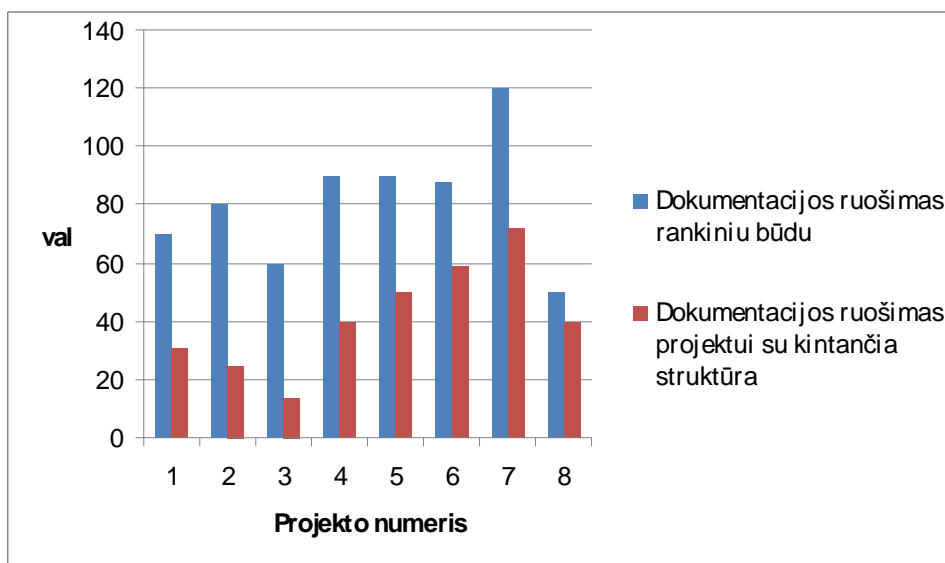
**6.4 lentelė.** Eksperimento dalyvių projekto darbų įvertinimai

	Laikas, sugaištas generuojant dokumentaciją, panaudojant šabloną projektams su fiksuota struktūra	Laikas, sugaištas generuojant dokumentaciją, panaudojant šabloną projektams su kintančia struktūra
1.	24	31
2.	17	25
3.	8	14
4.	32	40
5.	44	50
6.	50	59
7.	61	72
8.	32	40

Panaudojus Projektų dokumentavimo sistemą, vieno projekto dokumentaciją, naudojant šabloną projektams su fiksuota struktūra, pavyko paruošti vidutiniškai 60%, o naudojant šabloną projektams su kintančia struktūra, vidutiniškai 50% greičiau. 6.4 paveiksle matyti grafinis kiekvieno projekto dokumentacijos ruošimo ranka ir panaudojant Projektų dokumentacijos sistemos šabloną projektams su fiksuota struktūra darbų trukmės palyginimas, o 6.5 paveiksle panaudojant šabloną projektams su kintančia struktūra.



**6.4 pav.** Dokumentacijos ruošimo rankiniu būdu ir generuojant panaudojus šabloną projektams su fiksuota struktūra palyginimas



**6.5 pav.** Dokumentacijos ruošimo rankiniu būdu ir generuojant panaudojus šabloną projektams su kintančia struktūra palyginimas

Teoriškai sugeneruoti dokumentaciją panaudojant šabloną projektams su kintančia struktūra, turėtų būti greičiau, nei panaudojant šabloną projektams su fiksuota struktūra. Iš tikrųjų, taip ir būtų, jei būtų projektuojama laikantis Projektų dokumentavimo sistemos nurodymų. Tačiau, kadangi eksperimente buvo naudojami projektai, kurie nebuvo skirti dokumentacijai generuoti, todėl reikėjo atlikti nemažai pakeitimų tam, kad projektą būtų galima pritaikyti dokumentacijai generuoti pagal Projektų dokumentavimo sistemą. Tam, kad būtų sugeneruota dokumentacija pagal šabloną projektams su fiksuota struktūra, reikėjo šablone pakoreguoti tas vietas, kuriose buvo nuorodos į pavyzdinės sistemos katalogus. Tuo tarpu generuojant dokumentaciją pagal šabloną projektams su kintančia struktūra, pakeitimai užtruko ilgiau, nes reikėjo perkurti viso projekto katalogų struktūrą tam, kad būtų suformuota dokumento struktūra. Be to, daug laiko užtruko užpildyti elementų komentarus, kurie įtraukiami į sugeneruotą dokumentą. Taip pat reikėjo užpildyti aplinkos ir techninių faktorių bei panaudojimo atvejų ir aktorių sudėtingumo vertes, kurios naudojamos skaičiuojant programavimo ir testavimo darbų apimtį.

Eksperimento metu nebuvo vertinama kiekvieno eksperimento dalyvio sugebėjimas pritaikyti Projektų dokumentavimo sistemą. Kadangi dokumentaciją generavo Projektų dokumentavimo sistemos autorius, todėl laikas, kuris buvo reikalingas pritaikyti šabloną konkrečiam projektui, yra labai trumpas, palyginus su dokumentacijos ruošimu ranka. Be abejonės, paprastam vartotojui pritaikyti šabloną

savo reikmėms būtų reikėję žymiai daugiau laiko. Tačiau, eksperimento atveju, galima įvertinti, kad, turint pakankamai įgūdžių, dokumentaciją galima paruošti dvigubai ar dar greičiau.

### **6.5. Eksperimento išvados**

1. Apklausus eksperimento dalyvius, paaiškėjo, kad daugiau nei pusės projektų tiek projekto kūrimo, tiek dokumentavimo darbai užtruks ilgiau, nei buvo planuota.
2. Įvertinant dokumentacijos ruošimo darbų svarbą, net 3 projektų iš 7 dokumentacijos ruošimo darbams suplanuota daugiau laiko nei paties projekto kūrimui.
3. Sugeneravus dokumentaciją pagal Projektų dokumentavimo sistemos metodiką, tiek panaudojus šabloną projektams su fiksuota struktūra, tiek projektams su kintančia struktūra, dokumentaciją pavyko paruošti žymiai greičiau.
4. Pagrindinė problema, su kuriai buvo susidurta generuojant dokumentaciją, buvo projekto katalogo struktūros pritaikymas šablonams. Todėl, kuo tvarkingiau ir sistemiškiau bus kuriamas projektas, tuo lengviau bus pritaikyti šabloną.
5. Kadangi eksperimente buvo naudojami projektai, kurie nebuvo skirti dokumentacijos generavimui, todėl tam, kad būtų sugeneruotas išsamus dokumentas, reikėjo užpildyti elementų komentarus, nurodyti aplinkos ir techninių faktorių, aktorių ir panaudojimo atvejų svorines vertes programavimo ir testavimo darbų apimčių skaičiavimui.
6. Dėl to, kad eksperimente naudoti projektai nebuvo projektuojami laikanti RUP nurodymų, todėl šiems projektams nebuvo įmanoma pilnai pritaikyti projekto elementų ryšių pilnumo nustatymo.

## 7. Išvados

1. Atlikus kaskadinio, persidengimų, evoliucinio ciklo, laipsniškojo kūrimo ir kitų projektavimo metodų analizę, nustatyta, kad projekto kūrimo strategijos neapima dokumentavimo kaip atskiros projektinės veiklos, dėl to šis procesas nėra aiškiai apibrėžtas.
2. Siekiant užpildyti šią spragą ir remiantis unifikuotu kūrimo procesu (RUP), sudaryta projektavimo metodika ir suprojektuota dokumentavimo sistema, kuri realizuota UML CASE įrankyje *MagicDraw UML*, naudojant VTL kalbą šablonams realizuoti ir *MagicDraw UML* ataskaitų vedlį dokumentams generuoti.
3. Taikant šią sistemą projektui dokumentuoti, galima naudoti iš ansto parengtus ataskaitų šablonus, kurių vienas skirtas dokumentavimui pagal tipinę, kitas – pagal laisvai nustatytą projekto struktūrą. Šiuos šablonus galima prisitaikyti savo poreikiams arba kurti naujus.
4. Panašaus pobūdžio šablonai suteikia galimybę sugeneruoti tik atskiras dokumento dalis, pavyzdžiui, „Panaudojimo atvejų dokumentas“, „Klasių diagramų dokumentas“ ir pan. Taikant metodikoje siūlomus šablonus, galima sugeneruoti pilną projekto dokumentaciją.
5. Metodikos naujumas yra tas, kad ji leidžia suskaičiuoti preliminarią darbų trukmę ir įvertinti projekto stadijų rezultatų integralumą: ar egzistuoja *Realization* tipo ryšiai tarp panaudojimo atvejų ir jų realizacijų kooperacijomis (*Collaboration*), tarp panaudojimo atvejų realizacijų ir analizės (*Analysis Class*) bei projekto (*Design Class*) klasių, tarp projekto klasių ir komponentų bei tarp komponentų ir artefaktų.
6. Metodika buvo išbandyta su pavyzdine „Apklausų projektavimo ir vykdymo sistema“ su pagrindinėmis projektavimo diagramomis ir projektavimo principais bei suformuluotais reikalavimais.
7. Metodikai įvertinti atliktas eksperimentas su 8 projektais parodė, kad, įsisavinus sukurtą metodiką, paruošti dokumentaciją pagal Projektų dokumentavimo sistemą galima žymiai sparčiau, nei įprastiniu rankiniu būdu.
8. Darbo tema parašytas straipsnis ir pristatytas “Mag&Dok IT2011” konferencijoje.

## 8. Literatūra

- [1] Pulkkinen, M., Hirvonen, A. „EA Planning, Development and Management Process for Agile Enterprise Development“, Proceedings of the 38th Hawaii International Conference on System Sciences, 2005, p. 2-3.
- [2] Priestley, M. „A unified process for software and documentation development“, Professional Communication Conference, Proceedings of 2000 Joint IEEE International and 18th Annual Conference on Computer Documentation, 2000.
- [3] The Institute of Electrical and Electronics Engineers, IEEE Standard for Software Test Documentation, IEEE 829-1998, 1998, p. 1-10.
- [4] The Institute of Electrical and Electronics Engineers, Systems and software engineering - Software life cycle processes, ISO/IEC 12207 standartas, 2008, p. 9-84 .
- [5] Robertson, J., S. „Volere Requirements Specification Template“, Edition 13, 2007, p. 1.
- [6] Rational Unified Process. Prieiga per Internetą: <http://www.rational.com/rup/>
- [7] Ambler, S.W., Introduction to the Enterprise Unified Process (EUP). A Ronin International, Inc. White Paper, 2002, p. 5-12.
- [8] Basic Enterprise Architecture Methodology BEAM . Prieiga per Internetą: <http://www.one-world-is.com/beam/>
- [9] Patel, J., Lee, R., Kim, H-K. „Architectural View in Software Development Life-Cycle Practices, 2007, p. 2-4.
- [10] Langari, Z., Pidduck, A. B. „Quality, cleanroom and formal methods,“ 2005, p. 3-4.
- [11] Wang, J., Lin, Y-I., „An overlapping process model to assess schedule risk for new product development“, 2007, p.7-9.
- [12] Mantere, T, Alander T., „Evolutionary software engineering“, a review, 2004, p. 3-4.
- [13] Banerjee G., „Use Case Points – An Estimation Approach“, 2001.
- [14] Nageswaran S., „Test Effort Estimation Using Use Case Points“, Quality Week 2001, San Francisco, Kalifornija, JAV, 2001.
- [15] Savulis R. Projektų dokumentavimas MagicDraw UML įrankio pagrindu // Mag&Dok IT2011: XVI tarpuniversitetinės magistrantų ir doktorantų konferencijos pranešimų medžiaga [Kaunas, 2011 m. balandžio 22 d.]. Kaunas, 2011, p.101-104.

## 9. Priedai

### 9.1. Priedas. Apklausos anketa

Prašau padėti atlikti tyrimą tema dėl projektų dokumentavimo efektyvumo.  
Dėkojame už Jūsų atsakymus!

Pagarbiai,

ISI magistrantas Raimondas Savulis

1.	Jūsų pavardė, vardas, e. paštas		
2.	Anketos pildymo data		
3.	Kiek procentų bakalauro darbo projekto padarėte?		
4.	Per kiek laiko (valandomis) buvo planuota sukurti visą bakalauro darbo projektą?		
5.	Kiek faktiškai laiko (valandomis) iki dabar buvo sugaišta sistemos projekto kūrimui?		
6.	Per kiek laiko (valandomis) buvo planuota sukurti visas bakalauro darbo projekto ataskaitas?		
7.	Kiek faktiškai laiko (valandomis) iki dabar sugaišote bakalauro darbo ataskaitų kūrimui?		
8.	Kokiu būdu buvo ruošiama dokumentacija? (Pažymėkite + )	a) rankiniu	
		b) generavote	
9.	(Jei 5 klausime pasirinkote b variantą, šį klausimą praleiskite) Ar kada nors domėjotės galimybe generuoti dokumentaciją tiesiai iš projektavimo įrankio? (Pažymėkite + )	taip	
		ne	

## 9.2. Priedas. Mokslinėje konferencijoje „Mag&Dok IT2011“

### pristatytas straipsnis

# PROJEKTŲ DOKUMENTAVIMAS MAGICDRAW UML ĮRANKIO PAGRINDU

**Raimondas Savulis**

*Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų g. 50,  
Kaunas, Lietuva, raimondas.savulis@gmail.com*

**Santrauka.** Programinės įrangos kūrimo projektų dokumentacijos kūrimas dažnai vykdomas neefektyviai – dokumentai kuriami lėtai, esama dokumentavimo metodika būna perteklinė, nėra vieningos dokumentacijos kūrimo strategijos. Vienas iš būdų paspartinti dokumentų ruošimo procesą – panaudoti šiame darbe suformuluotus nurodymus bei principus. Taikant sukurtus dokumentų šablonus, dokumentaciją galima generuoti tiesiai iš projektavimo įrankio. Šiuose šablonuose realizuotas papildomas funkcionalumas – projekto kūrimo ir testavimo darbų apimčių skaičiavimas bei elementų ryšių vientisumo nustatymas, o tai svarbu vertinant projekto kokybę ir planuojant kūrimo darbus. Sukūrus projektą, dokumentacija automatiškai sugeneruojama panaudojant projektavimo įrankio, pavyzdžiui, *MagicDraw UML*, teikiamas dokumentų generavimo priemones, todėl nereikia atskirai kurti dokumentacijos teksto redagavimo priemonėmis. Sukurtąjį dokumento šabloną galima pakartotinai panaudoti kituose projektuose.

**Raktiniai žodžiai:** Projektų dokumentavimas, dokumento šablonas, dokumentacijos kokybės kriterijai, projekto kūrimo darbų įvertinimas, elementų ryšių vientisumo nustatymas

## 1. Įvadas

Galutinio vartotojo informacija, arba dokumentacija, pateikiama kartu su kompiuterizuota informacine sistema (KIS) kaip neatsiejama jos dalis. Iš tikrųjų KIS negalima pristatyti, kol neišbaigta jos dokumentacija. Tačiau dauguma KIS kūrimo procesų nelaiko rimtomis veiklų, kurios sudaro vartotojo informacijos kūrimo procesą. Dėl to vartotojo informacija labai dažnai sukuriama paskubomis projekto pabaigoje, dėl ko kyla reali grėsmė, kad dokumentacija bus neišbaigta, netiksli, todėl paprasčiausiai nenaudinga.

Tam, kad būtų suformuoti nurodymai bei identifikuotas programinės įrangos (PI) dokumentacijos rengimo procesas, tyrimo sritimi pasirinkta informacinių sistemų projektavimo ir dokumentavimo metodai, o tyrimo objektas – dokumentacijos rengimo procesas. Yra daug straipsnių ir atliktų darbų, kuriuose analizuojami projektavimo metodai, tačiau darbų apie dokumentavimo metodus ar dokumentacijos rengimo procesą sunku surasti – yra nurodymai *Rational Unified Process (RUP)* dokumentacijai [4], *Agile* procesų dokumentacija [5], ISO/IEC 12207 standarte [6] apibrėžti procesai ir dokumentų tipai, tačiau jie ne itin naudingi, kai juos reikia pritaikyti konkrečiam projektui. Todėl šiame darbe bus stengiamasi atskleisti bei identifikuoti dokumentavimo proceso gaires, kurios būtų naudingos konkrečiam projektui.

Ši straipsnį, be įvado ir literatūros sąrašo, sudaro keturios dalys. Antroje dalyje pateikiama esamų projektavimo procesų, įrankių ir su jais susijusių dokumentų analizė, trečioje – pateiktas dokumentacijos rengimo procesas. Ketvirtoje ir penktoje dalyse aprašomos įdomesnės sukurtos dokumentavimo įrankio funkcijos – kūrimo ir testavimo darbų apimties nustatymas ir projekto etapų artefaktų ryšių vientisumo tikrinimas, o šeštoje – darbo išvados.

## 2. Projektavimo procesų, įrankių ir su jais susijusių dokumentų analizė

Kadangi specialių mokslinių tyrimų ataskaitų kūrimo klausimu nepavyko aptikti, buvo išanalizuoti esami PĮ projektų kūrimo metodai. Analizės metu įsitikinta, kad dokumentavimo procesas nėra pakankamai reglamentuotas. Tradicinės projektų kūrimo strategijos, pavyzdžiui, kaskadinis modelis[1], persidengimų [2], evoliucinis[3], laipsniškasis ir kt. net neįtraukia dokumentacijos kūrimo proceso į bendrą projekto kūrimo ciklą, todėl gali būti naudingi tik kaip strategijos principai dokumentacijai ruošti. Kiti metodai, pavyzdžiui, *Rational Unified Process (RUP)* [4] ar lankstieji (*Agile*) [5], pateikia gaires dokumentacijos kūrimui, tačiau *RUP* atveju dokumentacijos kūrimo šablonai yra pertekliniai ir sunkiai pritaikomi konkrečiam projektui; tuo tarpu *Agile* dokumentacija yra nepakankama. Egzistuojantys standartai, pavyzdžiui, *ISO/IEC 12207* [6], pateikia tik bendrus nurodymus, kokius procesus ir jų aspektus dokumentuoti, tačiau išsamesnių nurodymų dokumento sudėčiai, jau nekalbant apie kokius nors dokumentų šablonus, juose nėra.

Vienas iš galimų būdų dokumentavimo procesui paspartinti yra dokumentacijos generavimas projektavimo įrankiu taikant dokumentų šablonus. Nors ir egzistuoja nemažai projektavimo įrankių, pavyzdžiui, *Microsoft Office Visio*, *Visual Paradigm*, *Altova Umodel*, *MagicDraw UML*, suteikiančių galimybę generuoti dokumentaciją ir pateikiančių naudotojo instrukcijas, trūksta teorinio pagrindo dokumentacijos generavimui ir šablonų taikymui integruoti į bendrą projekto kūrimo procesą.

## 3. Dokumentacijos rengimo procesas

1 paveiksle vaizduojama apibendrinta dokumentacijos kūrimo schema rodo procesus, kuriuos reikia atlikti, norint sukurti sistemos dokumentaciją.



1 pav. Dokumentacijos kūrimo procesas

Visų pirma reikia išsiaiškinti patį projektavimo procesą bei jo etapus, kad būtų galima suprasti, kokios srities dokumentų reikia, norint dokumentuoti kuriamą sistemą. Išskirti kontroliniai taškai yra tie taškai, kuriuose atsiranda poreikis dokumentuoti, t. y. kai reikia tikrinti, derinti, tvirtinti ar pristatyti projekto artefaktus.

Apibrėžiant kontrolinių taškų artefaktus, išskiriami galimi ataskaitų tipai bei jų prioritetai, nustatomi svarbiausi artefaktai. Galimi ataskaitų tipai: „Veiklos analizės stadijos ataskaita“, „Reikalavimų specifikavimo ataskaita“, „Architektūros projekto ataskaita“, „Detalaus komponento projekto ataskaita“, „DB schemas projekto ataskaita“. Atskiru atveju tai gali būti „Magistro darbo projekto ataskaita“.

Vėliau suformuojama kontrolinio taško ataskaitos eskizas bei struktūra. Turint šią informaciją, galima formuoti ataskaitos šabloną, naudojant *VTL (Velocity Template Language)* kalbą – kalbą, kuria kuriami ataskaitų šablonai generavimui *MagicDraw UML* įrankiu. Sukūrus ataskaitos šabloną, atliekamas ataskaitos



generavimas. Praktiniam dokumentavimo metodikos taikymui iliustruoti buvo suprojektuota pavyzdinė „Apklausų projektavimo ir vykdymo“ sistema, kuriai buvo sukurtas dokumento šablonas bei sugeneruota viso projekto dokumentacija.

Nė vienas iš analizuotų metodų nenagrinėja dokumentų generavimo kaip alternatyvos dokumentacijos kūrimui. Tačiau būtent panaudojus dokumentų generavimą, galima paspartinti dokumentavimo procesą – sukūrus šabloną vienam projektui, jį galima panaudoti generuojant dokumentaciją kitiems projektams. Tad, nors iš pirmo žvilgsnio gali atrodyti, kad pritaikyti siūlomą dokumentacijos rengimo procesą ir jos pagrindinį artefaktą – šabloną – daug laiko sąnaudų reikalaujantis būdas, tačiau investicijos galėtų atsiperkti su kaupu.

#### 4. Kūrimo ir testavimo darbų apimties nustatymas

Be įprasto dokumentų generavimo funkcionalumo, sukurtame dokumentavimo šablone realizuotas projekto kūrimo ir testavimo darbų apimčių, išreiškiamų darbo dienomis bei mėnesiais, skaičiavimas.

Projekto kūrimo ir testavimo darbų skaičiavimas vykdomas įvertinant projekto techninius ir aplinkos faktorius, panaudojimo atvejų svorines vertes [7], [8] ir projekto komandos sudėtį.

Pažymėkime  $TFactor$  – bendra techninių faktorių vertė. Tuomet  $TFactor$  reikšmė bus apskaičiuojama pagal 1 formulę, kai  $i$  kinta nuo 1 iki 13 programavimo darbų skaičiavimo atveju, arba  $i$  kinta nuo 1 iki 9 testavimo darbų apimties skaičiavimo atveju.

$$TFactor = \sum (T_i * W_i), \text{ kur } i=1..8 \text{ arba } i=1..9 \quad (1)$$

Bendra aplinkos faktorių reikšmė  $EFactor$  apskaičiuojama pagal 2 formulę, kur  $i$  kinta nuo 1 iki 8.

$$EFactor = \sum (E_i * W_i), \text{ kur } i=1..8 \quad (2)$$

Techninio sudėtingumo vertė  $TCF$  gaunama pagal 3 formulę (programavimo darbų skaičiavimo atveju), arba pagal 4 formulę (testavimo darbų skaičiavimo atveju).

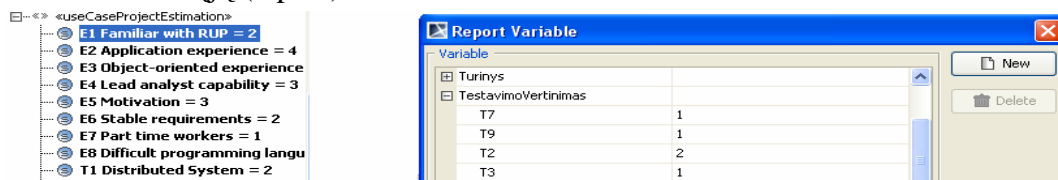
$$TCF = 0,6 + (0,01 * TFactor) \quad (3)$$

$$TCF = 0,65 + (0,01 * TFactor) \quad (4)$$

Aplinkos faktoriaus vertė  $ECF$  gaunama pagal 5 formulę.

$$ECF = 1,4 + (-0,03 * EFactor) \quad (5)$$

Projekto kūrimo darbų apimties skaičiavimui aktualių techninių ir aplinkos faktorių svorinės reikšmės nurodomos prie projekto savybių, o testavimo darbų apimties skaičiavimo techninių faktorių svorinės vertės nurodomos prie sukurtojo šablono kintamųjų (2 pav.).



2 pav. Aplinkos ir techninių faktorių svorinių reikšmių priskyrimas

Panaudojimo atvejų ir aktorių svorinės reikšmės nurodomos prie atitinkamo elemento ypatybių (3 pav.), kur kiekvienas sudėtingumo lygis turi atitinkamą skaitinį įvertį: „Low Complexity“ – 5, „Average Complexity“ – 10, „High Complexity“ – 15 panaudojimo atvejams; „Low Complexity“ – 1, „Average Complexity“ – 2, „High Complexity“ – 3 aktoriams.

Use Case Complexity	Average Complexity	Actor Complexity	High Complexity
Non Functional Requirements	High Complexity	Owner	High Complexity
Assumption	Average Complexity	Applied Stereotype	Average Complexity
Implementation Issues	Low Complexity	Is Abstract	Low Complexity

3 pav. Panaudojimo atvejų ir aktorių svorinių reikšmių priskyrimas

Tarkime, kad  $U_i$  yra vieno panaudojimo atvejo sudėtingumo svorinė vertė, o  $A_j$  – vieno aktoriaus sudėtingumo svorinė reikšmė. Tuomet  $UUCP$  apskaičiuojama pagal 6 formulę, kur  $i = 1..n, j = 1..m$ .

$$UUCP = \sum U_i + \sum A_j \quad (6)$$

Suradus  $UUCP$ ,  $TCF$  ir  $ECF$ , galima nustatyti apskaičiuojamų panaudojimo atvejų taškų reikšmę  $UCP$  (*adjusted Use Case Points*) (7 formulė).

$$UCP = UUCP * TCF * ECF \quad (7)$$

Tarkime, kad  $H$  yra pagal ankstesnių projektų patirtį nustatytas valandų skaičius, kuris sugaišamas vienam panaudojimo atvejui sukurti/ištestuoti. Tuomet viso projekto kūrimo/testavimo darbų apimtis žmogaus darbo valandomis apskaičiuojama  $UCP$  reikšmę padauginus iš  $H$ , t.y.  $UCP * H$ .

Tarkime, projekto kūrime dalyvaujam  $d$  programuotojų/testuotojų ir jie dirba  $dh$  darbo valandų per dieną, o mėnesyje yra 20 darbo dienų. Tuomet, pagal projekto komandos sudėtį, projekto kūrimas/testavimas užtruks  $UCP * H \div d \div dh$  dienų arba  $UCP * H \div d \div dh \div 20$  mėnesių.

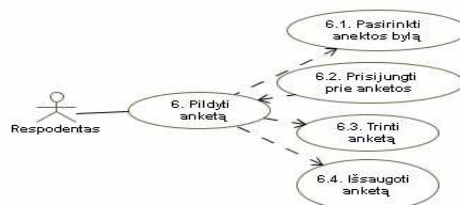
## 5. Projekto elementų ryšių vientisumo nustatymas

Projekto elementų ryšių vientisumo nustatymas įvertinamas analizuojant, ar pereinant prie vėlesnių projekto kūrimo etapų, t.y. nuo reikalavimų prie analizės, nuo analizės prie realizavimo, visi projekto elementai perkeliami į kitą etapą ir ar pilnai realizuojami ryšiai tarp skirtingų projekto kūrimo etapų. Tarkime, projektavimui buvo pritaikyta *Rational Unified Process (RUP)* metodika bei stereotipai ir projektavimo metu buvo sukurti šie realizavimo (*Realization*) ryšiai:

- tarp panaudojimo atvejų ir jų realizacijų kooperacijomis (*Collaboration*);
- tarp panaudojimo atvejų realizacijų ir analizės (*Analysis class*) bei projekto klasių (*Design Class*);
- tarp projekto klasių ir komponentų;
- tarp komponentų ir artefaktų.

Tarkime, reikalavimų identifikavimo metu buvo sudarytas toks panaudojimo atvejų modelis (4 pav.). Pereinant nuo reikalavimų identifikavimo prie analizės, sudarytas veiklos modelis turi būti perkeltas į analizės dalį, t.y. panaudojimo atvejai turi būti susieti *Realization* ryšiu su *Collaboration* tipo elementais.

Sakysime, projektuotojas per klaidą neperkėlė panaudojimo atvejo „6.2. Prisijungti prie sistemos“ iš reikalavimų identifikavimo į analizę ir turime tokią situaciją, kad tik du panaudojimo atvejai iš trijų buvo susieti *Realization* ryšiu su *Collaboration* tipo elementais, o panaudojimo atvejai „6.3. Trinti anketą“ ir „6.4. Išsaugoti anketą“ nebuvo susieti *Realization* ryšiu su projekto klasėmis.



4 pav. Reikalavimų identifikavimo metu sudaryto panaudojimo atvejų modelio pavyzdys

Jei projektuotojas norės ataskaitoje pateikti projekto elementų ryšių vientisumo užtikrinimą, bus sugeneruota tokia dokumento dalis:

### Projekto elementų ryšių pilnumo tikrinimas

Panaudojimo atvejai, nesusieti *Realization* ryšiu su klasėmis:

Elemento pavadinimas	Tėvinis elementas
◊ 6.3. Trinti anketą	PA modelis
◊ 6.4. Išsaugoti anketą	PA modelis

Panaudojimo atvejai, nesusieti *Realization* ryšiu su *Collaboration* elementais:

Elemento pavadinimas	Tėvinis elementas
◊ 6.2. Prisijungti prie anketos	PA modelis

5 pav. Dokumento fragmentas, kuris sugeneruojamas aptikus trūkstamą ryšį tarp elementų

## 6. Išvados

- Atlikus projektavimo procesų, įrankių ir su jais susijusių dokumentų analizę, pastebėta, kad dokumentavimo procesas nėra pakankamai reglamentuotas, nėra vieningos strategijos dokumentams rengti, todėl dokumentacija dažnai ruošiamą neefektyviai, paskubomis, dėl ko nukenčia jos kokybė.
- Šiame darbe sudaryta metodika ir sukurti *MagicDraw UML* įrankio ataskaitų generavimo šablonai leidžia sugeneruoti projektų dokumentaciją pagal iš anksto nustatytą arba projektuotojo norimą paketų struktūrą. Norint dar geriau pritaikyti savo poreikiams, šablonus galima toliau redaguoti.
- Sukurti šablonai turi papildomą funkcionalumą, kuriuo nepasižymi esami ataskaitų generavimo įrankiai: projektavimo ir testavimo darbų apimties skaičiavimą ir projekto elementų vientisumo tikrinimą. Elementų vientisumo tikrinimas leidžia objektyviai įvertinti, ar projekto dokumentacijoje nėra tam tikrų klaidų, o darbų apimties skaičiavimas palengvina projekto vykdymo užduočių planavimą. Modifikuojant projektą, ataskaitas galima greitai perdaryti iš naujo.
- Sukurtiems ataskaitų šablonams įvertinti numatytas eksperimentas, kuriame bus atliekama *MagicDraw UML* įrankio vartotojų apklausa ir prašoma įvertinti sukurtų priemonių funkcionalumą, patogumą ir kitas vartotojui svarbias savybes.
- Pagrindinis sprendimo trūkumas yra tas, kad didžioji sukurtojo šablono funkcionalumo dalis susijusi su panaudojimo atvejų modeliu, todėl jo viso negalima pritaikyti projektams, kuriuose šis modelis nenaudojamas. Be to, šablonas realizuotas taikant *Velocity Template Language* kalbą, todėl paprastam vartotojui gali būti sudėtinga redaguoti sukurtą šabloną dėl specifinės šios kalbos sintaksės.

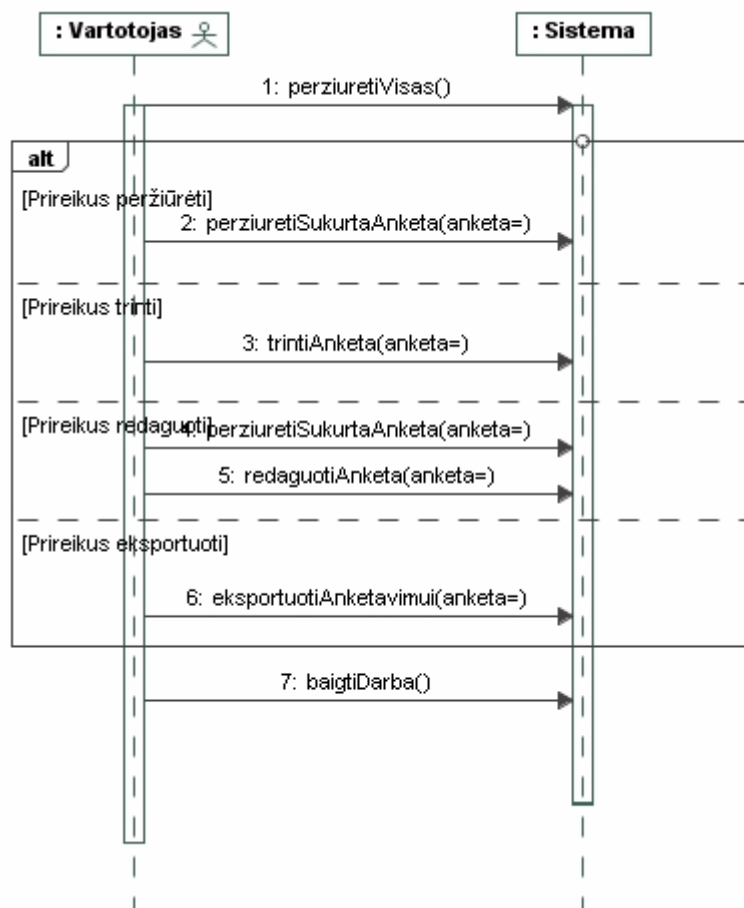
### Literatūros sąrašas

- [1] Patel J., Lee R., Kim, H-K. Architectural View in Software Development Life-Cycle Practices. 2007, p. 2-4.
- [2] Wang J., Lin Y-I. An overlapping process model to assess schedule risk for new product development. 2007, p. 7-9
- [3] Mantere T, Alander T. Evolutionary software engineering. 2004, p. 3-4.
- [4] Priestley M. A unified process for software and documentation development. *Professional Communication Conference, Proceedings of 2000 Joint IEEE International and 18th Annual Conference on Computer Documentation*, 2000.
- [5] Pulkkinen M., Hirvonen A. EA Planning, Development and Management Process for Agile Enterprise Development. "Proceedings of the 38th Hawaii International Conference on System Sciences", 2005, p. 2-3.
- [6] The Institute of Electrical and Electronics Engineers. IEEE Standard for Software Test Documentation, IEEE 829-1998. 1998, p. 1-10.
- [7] Banerjee G. Use Case Points – An Estimation Approach. 2001.

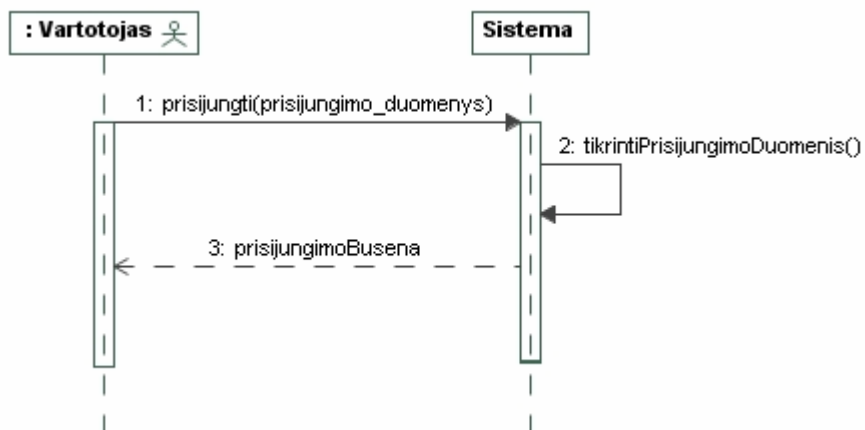
### Project Documentation Based On MagicDaw UML Tool

Creation of documentation usually is inefficient - documents are being created slowly, the methodology of documentation is usually redundant, there is no common strategy for this. The paper proposes a way to boost the process of creating documentation by using documentation templates that give an opportunity to generate documentation directly from a CASE tool. The created template implements additional functionality - the estimation of a size of programming and testing tasks of a project and tracing the dependencies between project's elements. The implemented template of a document may be applied in more than one project.

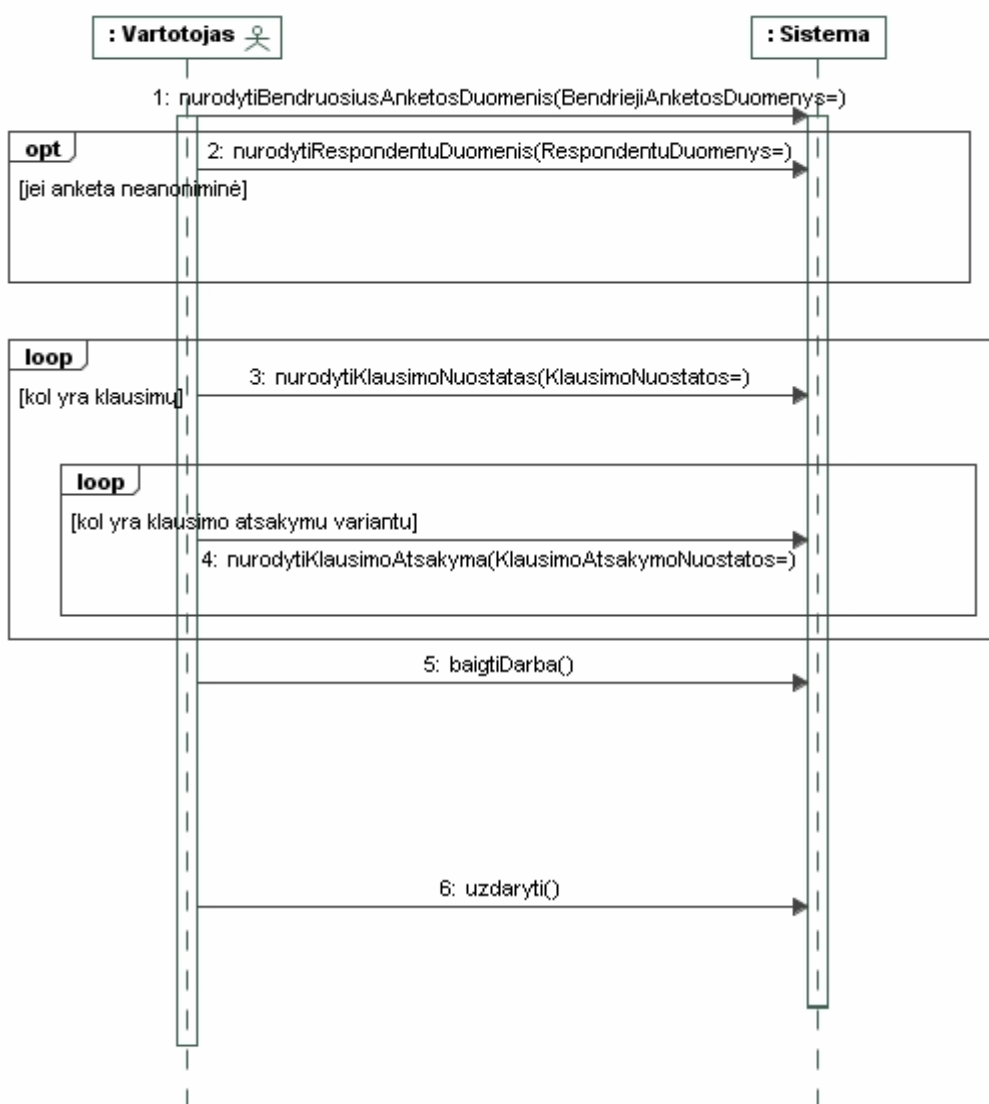
### 9.3. Priedas. Pavyzdinės apklausų informacinės sistemos sekų ir veiklos diagramos



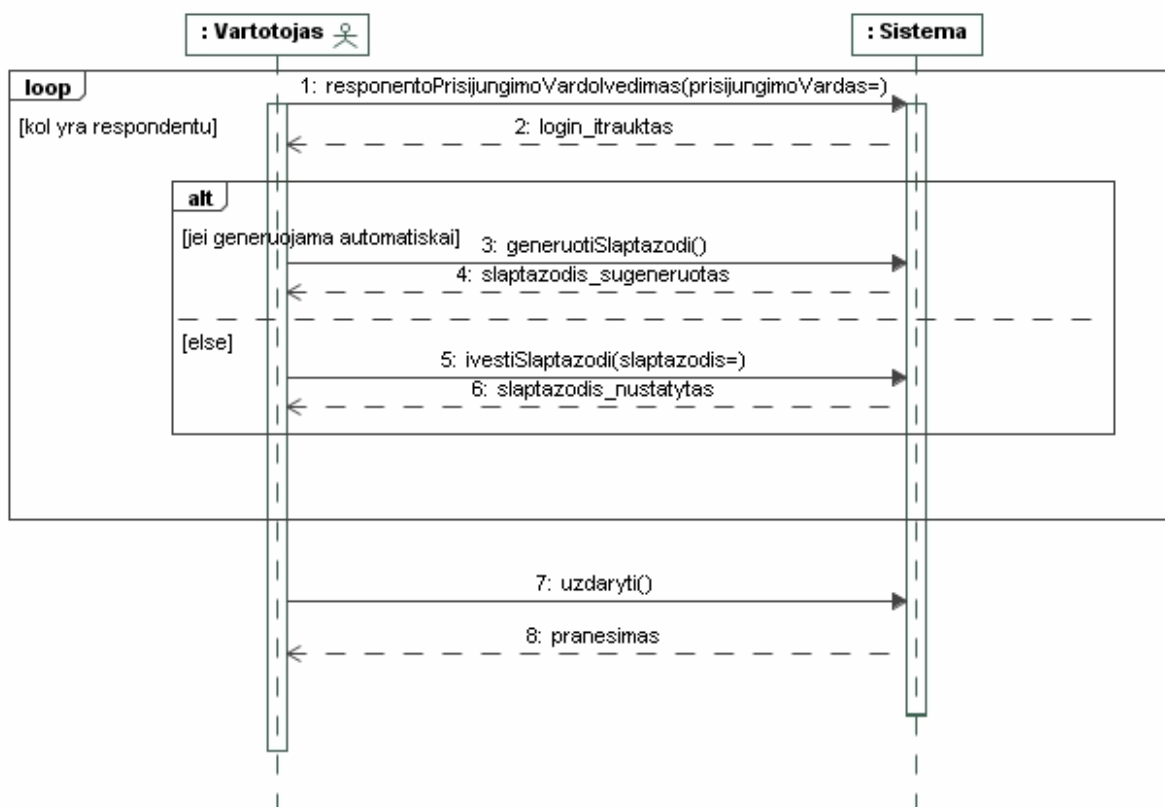
9.1 pav. Peržiūrėti apklausos anketas“ sekų diagrama



9.2 pav. „Prisijungti prie sistemos“ sekų diagrama



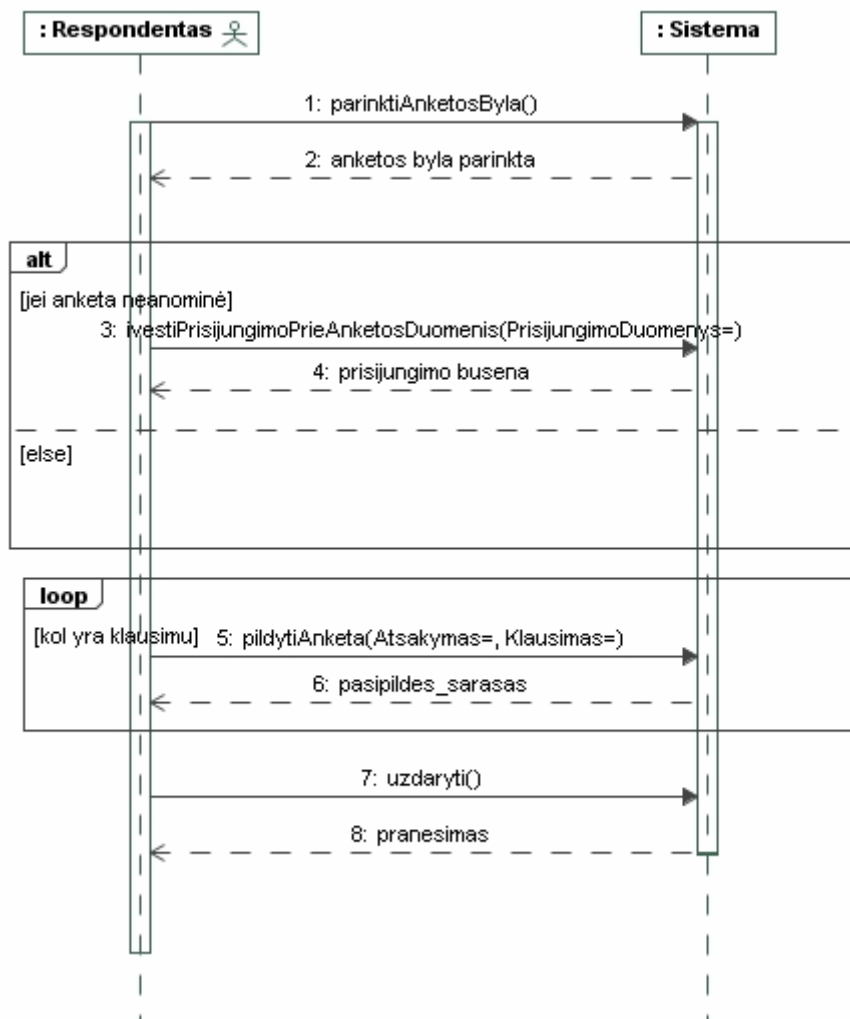
9.3 pav. „Sukurti naują anketą“ sekų diagrama



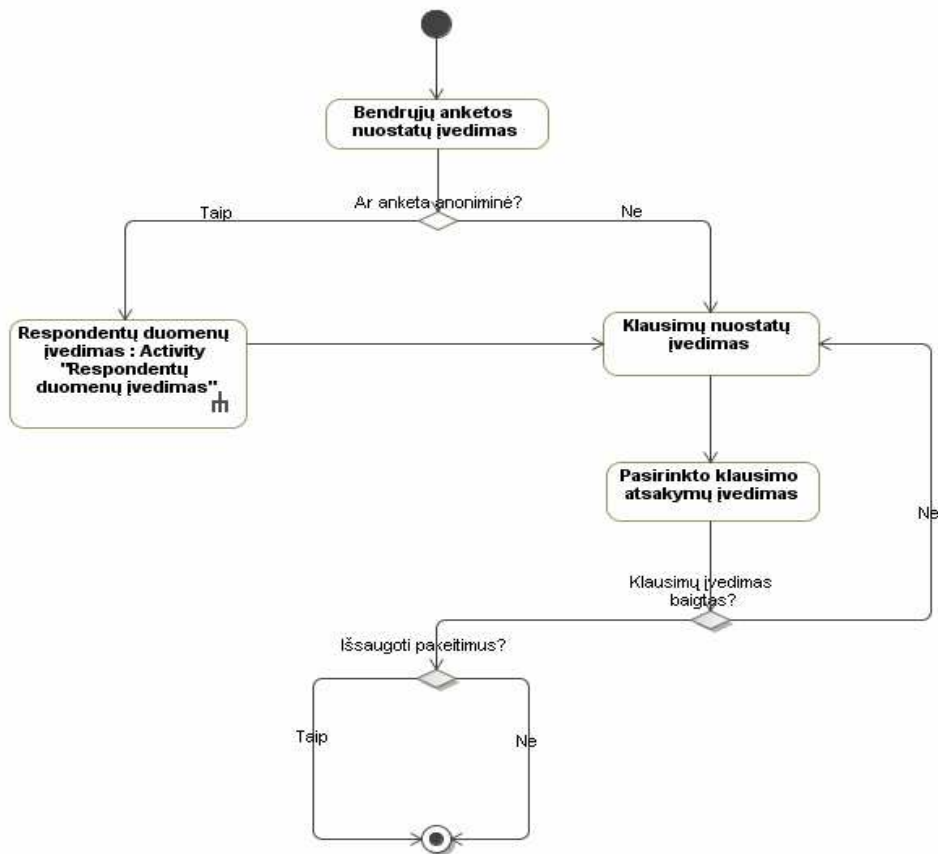
9.4 pav. „Respondentų įvedimas“ sekų diagrama



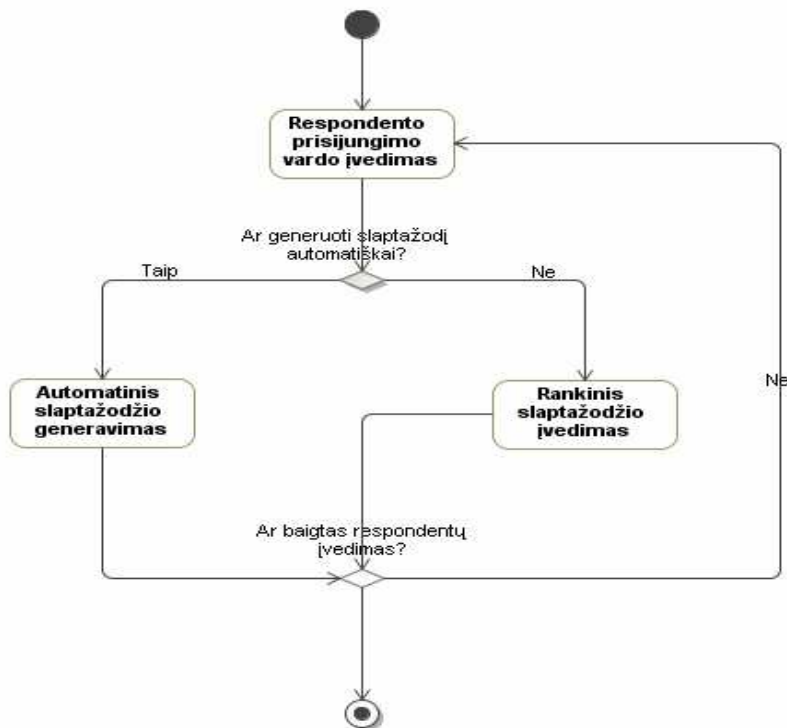
9.5 pav. „Trinti anketa“ sekų diagrama



9.6 pav. „Pildyti anketa“ sekų diagrama

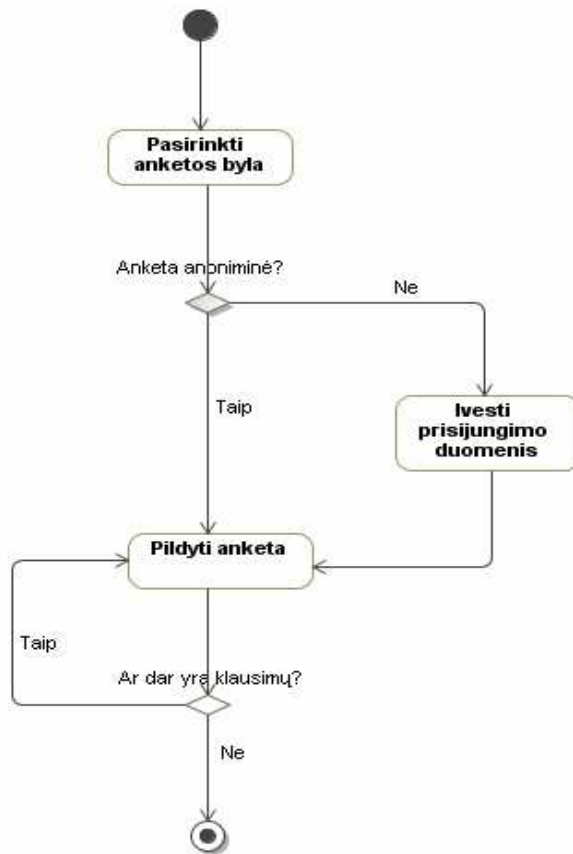


9.7 pav. „Sukurti naują anketą“ veiklos diagrama

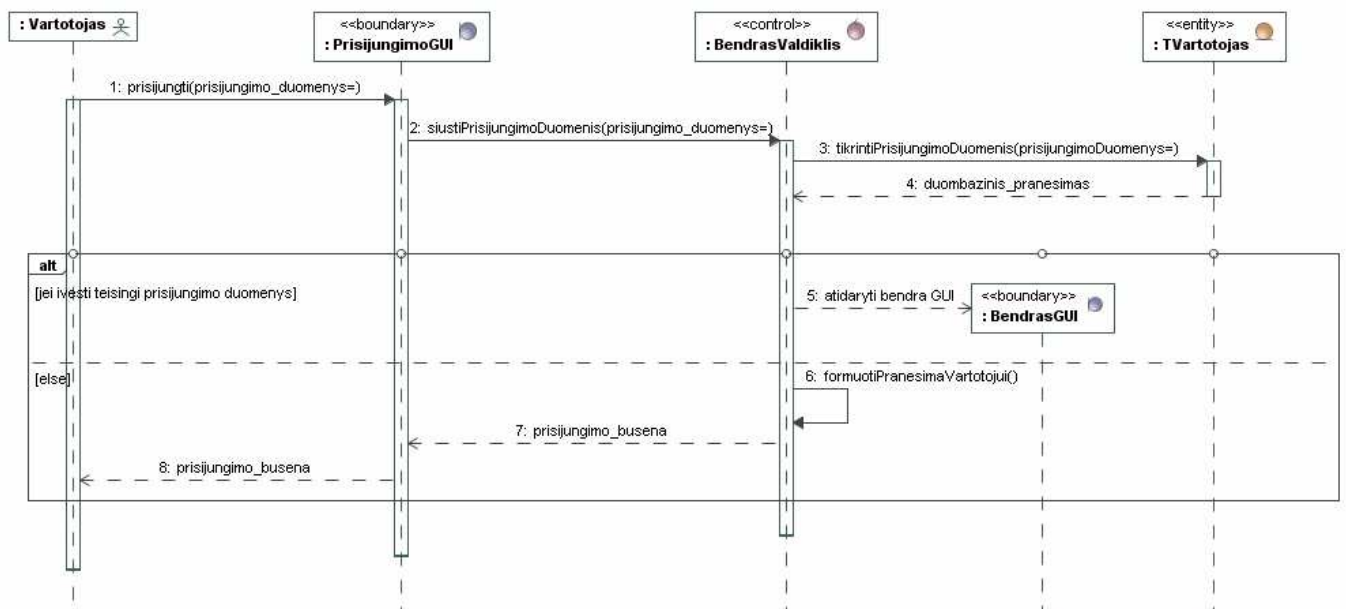


9.8 pav. „Respondentų įvedimas“ veiklos diagrama

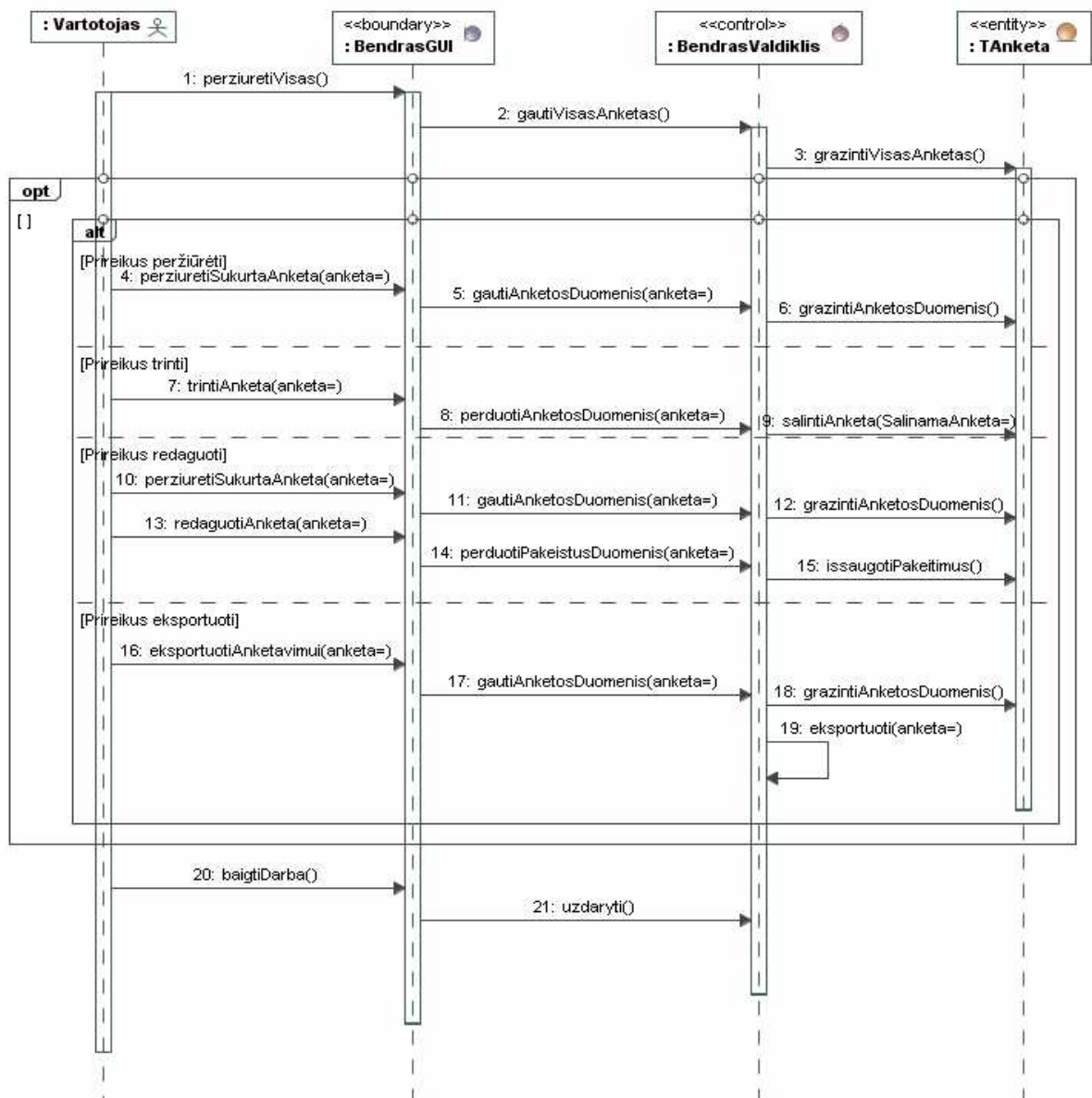




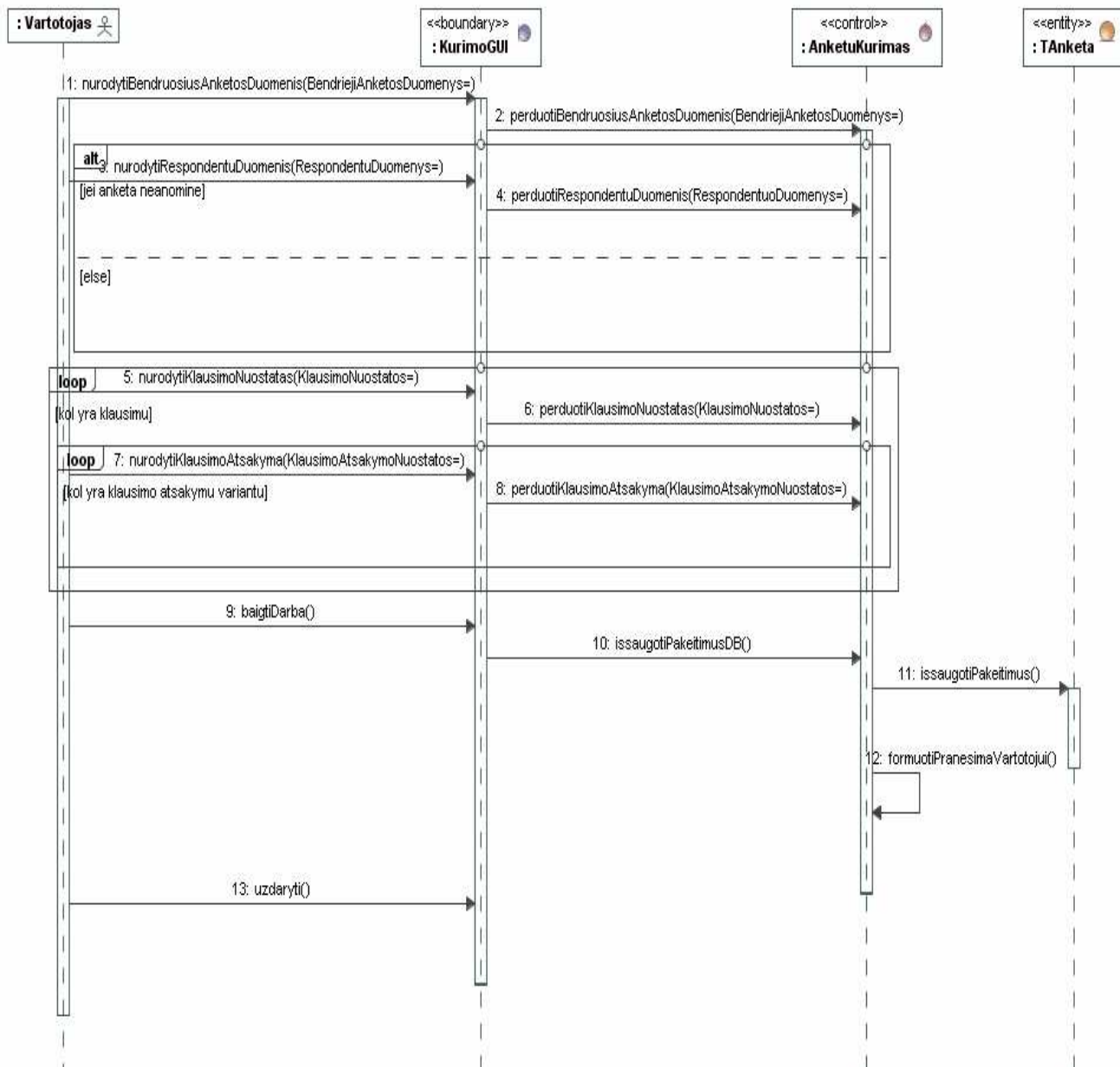
9.9 pav. „Pildyti anketa“ veiklos diagrama



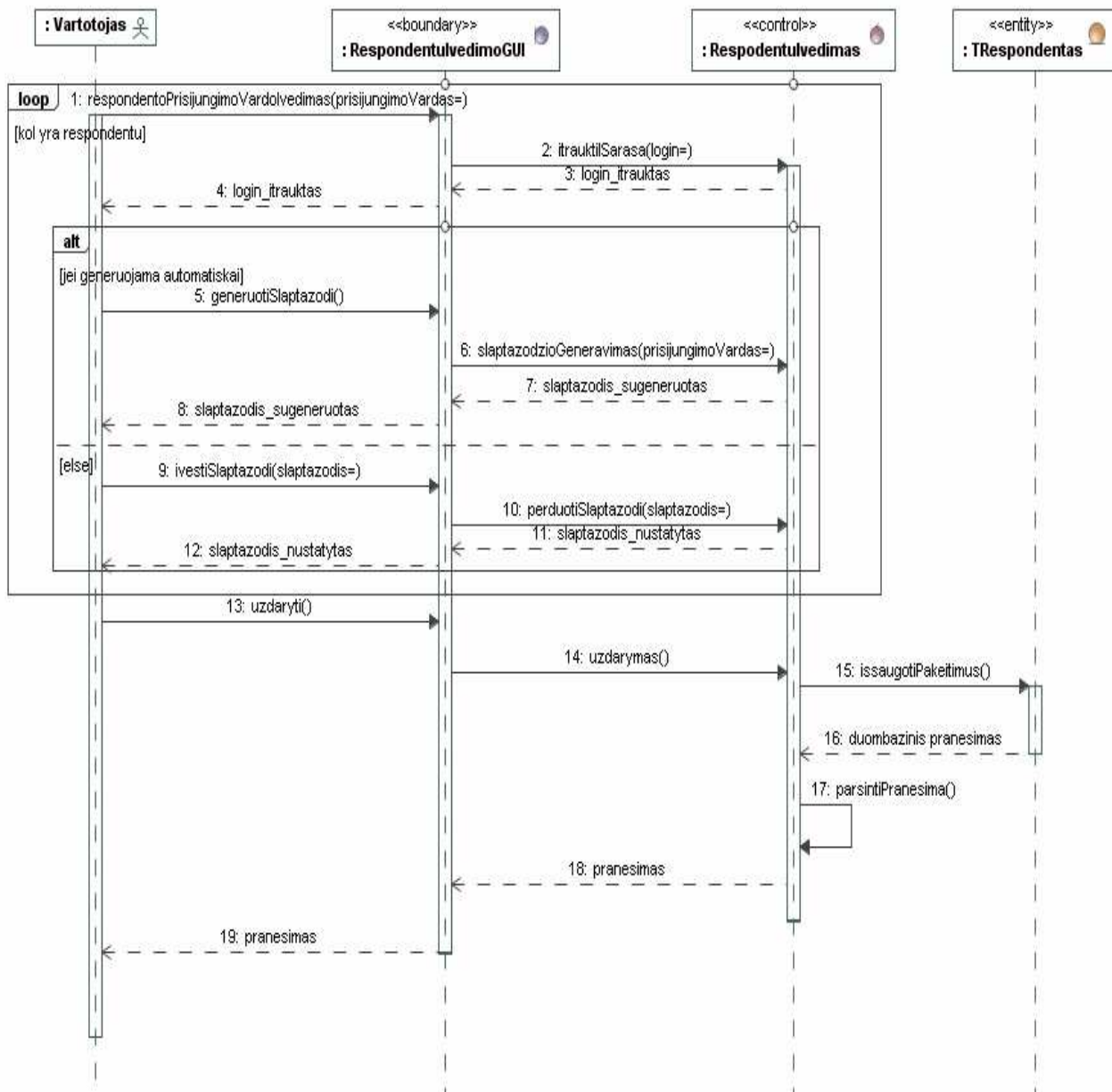
9.10 pav. „Prisijungti prie sistemos“ projekto lygio sekų diagrama



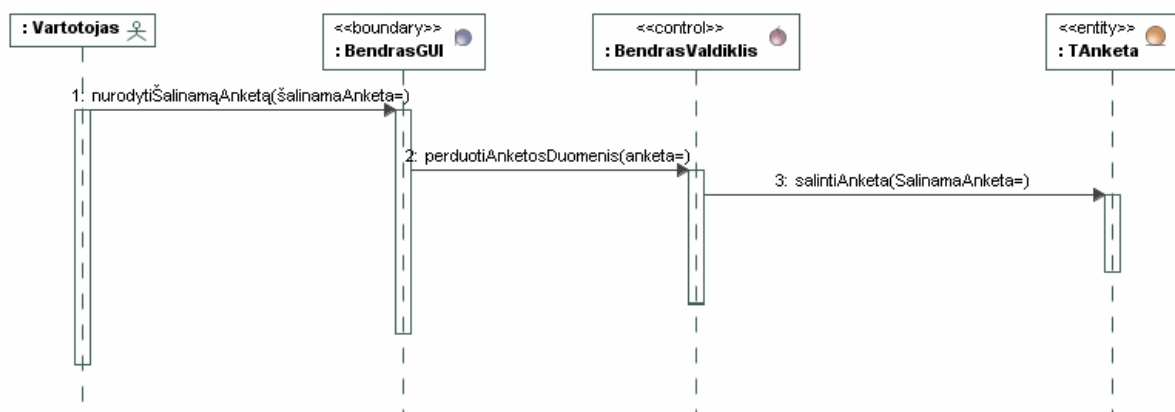
9.11 pav. „Peržiūrėti apklausos anketas“ projekto lygio sekų diagrama



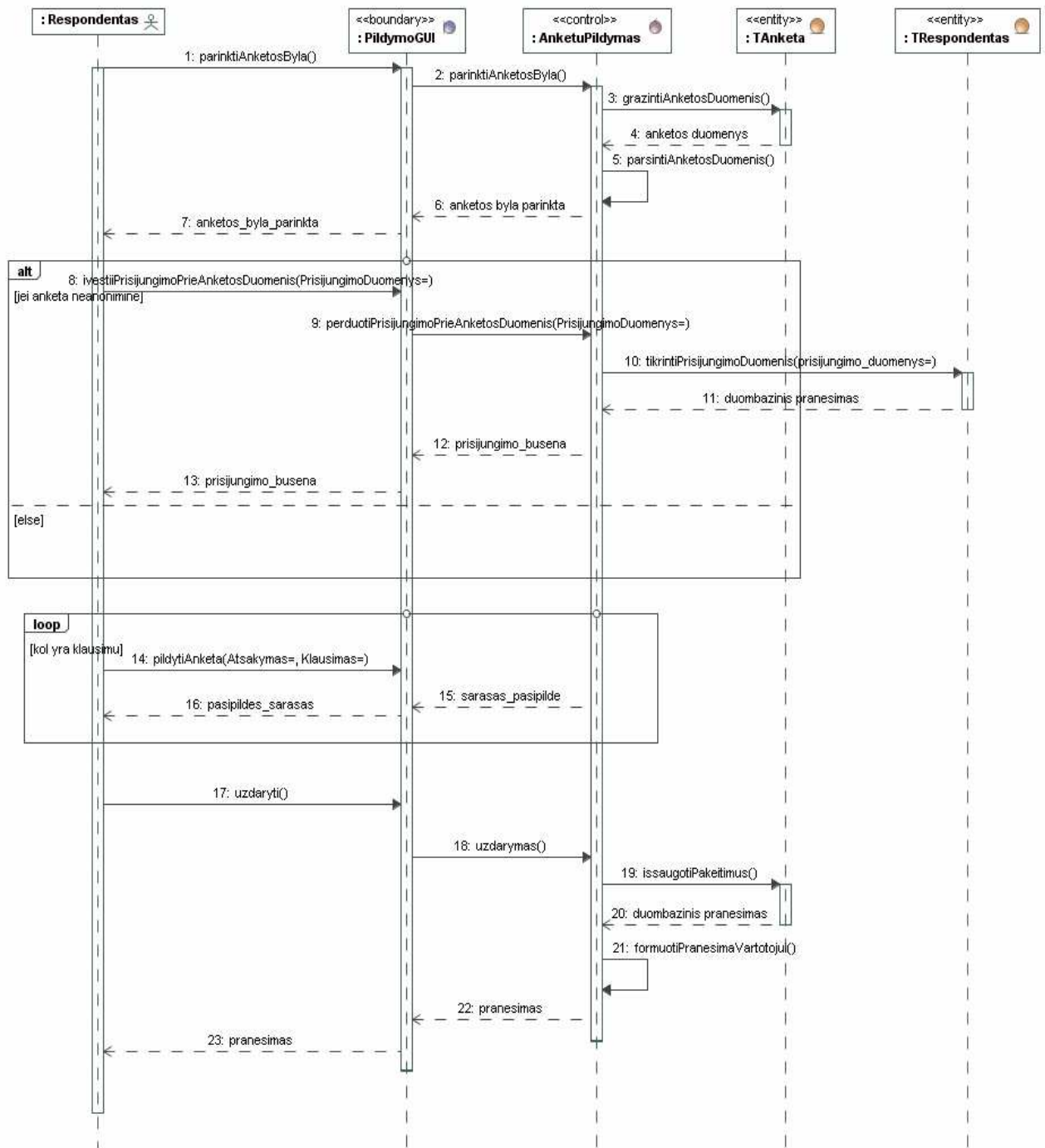
9.12 pav. „Sukurti naują anketą“ projekto lygio sekų diagrama



9.13 pav. „Respondentų įvedimas“ projekto lygio sekų diagrama



9.14 pav. „Trinti anketą“ projekto lygio sekų diagrama



9.15 pav. „Pildyti anketą“ projekto lygio sekų diagrama