

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Jūratė Čekanauskaitė, Jolanta Čekanauskaitė

**Reikalavimų specifikacijos integralumo užtikrinimo
metodas**

Magistro darbas

Darbo vadovas

prof. Rimantas Butleris

KAUNAS, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Jūratė Čekanauskaitė, Jolanta Čekanauskaitė

**Reikalavimų specifikacijos integralumo užtikrinimo
metodas**

Magistro darbas

Recenzentas

doc. dr. R. Misevičienė

2011-05-27

Darbo vadovas

prof. Rimantas Butleris

2011-05-27

Atliko

IFM - 9/4 gr. Studentės

Jūratė Čekanauskaitė, Jolanta Čekanauskaitė

2011-05-27

KAUNAS, 2011

Turinys

1. ĮVADAS	11
2. REIKALAVIMŲ, REIKALAVIMŲ INTEGRALUMO METODŲ, SPECIFIKAVIMO ŠABLONŲ IR REIKALAVIMŲ VALDYMO ĮRANKIŲ ANALIZĖ	14
2.1. ANALIZĖS TIKSLAS	14
2.2. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA	14
2.3. REIKALAVIMŲ INŽINERIJA	14
2.4. AORE INŽINERIJA	15
2.5. REIKALAVIMŲ ANALIZĖ.....	16
2.5.1. <i>Besikertančių reikalavimų analizė</i>	16
2.5.2. <i>Besikertančių reikalavimų identifikavimas</i>	17
2.5.3. <i>Besikertančių reikalavimų išgavimo būdai ir strategijos</i>	20
2.5.4. <i>Reikalavimų kokybės charakteristikos</i>	22
2.6. KOKYBIŠKOS REIKALAVIMŲ SPECIFIKACIJOS SAVYBĖS IR VERTINIMO KRITERIJAI	30
2.7. REIKALAVIMŲ KLASIFIKACIJA REMIANTIS ISO 9126 IR FURPS+ STANDARTAIS	32
2.7.1. <i>ISO 9126 - tarptautinis PJ kokybės vertinimo standartas</i>	32
2.7.2. <i>FURPS+ modelis</i>	34
2.8. REIKALAVIMŲ SPECIFIKAVIMO ŠABLONŲ ANALIZĖ	35
2.8.1. <i>SRS šablono analizė</i>	35
2.8.2. <i>URS šablonas</i>	37
2.8.3. <i>Volere šablono analizė</i>	38
2.8.4. <i>FSD reikalavimų specifikavimo šablonas</i>	40
2.8.5. <i>Reikalavimų specifikavimo šablonų palyginimas</i>	41
2.9. REIKALAVIMŲ SPECIFIKAVIMO ĮRANKIŲ ANALIZĖ	42
2.9.1. <i>RequisitePro įrankis</i>	42
2.9.2. <i>DOORS įrankis</i>	43
2.9.3. <i>CaliberRM įrankis</i>	44
2.9.4. <i>TopTeam™ Analyst įrankis</i>	45
2.9.5. <i>Reikalavimų specifikavimo įrankių palyginimas</i>	46
2.10. FR IR NFR REIKALAVIMŲ INTEGRALUMO METODŲ IR PRINCIPŲ ANALIZĖ.....	48
2.10.1. <i>FR ir NFR integralumas, naudojant panaudos atvejų ir tikslais grindžiamą metodą</i>	48
2.10.2. <i>NFR integravimas į konceptualų modelį</i>	50
2.10.3. <i>Esamų sprendimų analizė</i>	52
2.11. REIKALAVIMŲ SPECIFIKACIJOS VARTOTOJŲ ANALIZĖ	54
2.11.1. <i>Vartotojų aibė, tipai ir savybės</i>	54
2.11.2. <i>Vartotojų tikslai ir problemos</i>	54
2.12. ANALIZĖS IŠVADOS.....	55
3. REIKALAVIMŲ SPECIFIKAVIMO INTEGRALUMĄ UŽTIKRINANTIS MODELIS	56
3.1. REIKALAVIMŲ INTEGRALUMĄ UŽTIKRINANČIO METODO PROCESŲ MODELIS.....	56
3.2. REIKALAVIMŲ INTEGRALUMĄ UŽTIKRINANTIS PROCESŲ MODELIS	57
3.3. REIKALAVIMŲ IDENTIFIKAVIMO PROCESAS	57
3.4. PANAUDOJIMO ATVEJŲ APRAŠYMO VEIKLOS MODELIS	62
3.5. REIKALAVIMŲ SPECIFIKAVIMO PROCESAS	64
3.6. REIKALAVIMŲ KOMPONAVIMO PROCESAS.....	67
3.7. REIKALAVIMŲ INTEGRALUMĄ UŽTIKRINANTIS MODELIS.....	72
3.8. REIKALAVIMŲ INTEGRALUMĄ UŽTIKRINANČIO METODO TAISYKLĖS	73
3.9. REIKALAVIMŲ INTEGRALUMO METODIKOS APIBENDRINIMAS	76
4. REQUISITE PRO IR MAGIC DRAW ĮRANKIŲ PRITAIKYMAS METODUI	78
4.1. ŠABLONO KŪRIMAS REQUISITEPRO ĮRANKIU.....	78
4.2. REIKALAVIMŲ INTEGRALUMO METODO ATVAIZDAVIMAS UML	82

4.2.1.	<i>Reikalavimų integralumo profilio kūrimo procesas</i>	82
4.2.2.	<i>Stereotipų modulio kūrimas</i>	82
4.3.	METODO REALIZACIJAI PASIRINKTŲ ĮRANKIŲ REQUISITE PRO IR MAGICDRAW IŠPLĖTIMO APIBENDRINIMAS	89
5.	METODO REALIZACIJA PASIRINKTOMIS PRIEMONĖMIS	91
5.1.	PASIRINKTOS DALYKINĖS SRITIES APRAŠAS	91
5.1.1.	<i>Reikalavimų identifikavimo etapas</i>	92
5.2.	REIKALAVIMŲ SPECIFIKAVIMO ETAPAS	97
5.3.	REIKALAVIMŲ KOMPONAVIMO ETAPAS	104
5.4.	REIKALAVIMŲ RINKINIŲ ATVAIZDAVIMAS UML	108
5.5.	REALIZACIJOS IŠVADOS	111
6.	REIKALAVIMŲ INTEGRALUMO METODIKOS EKSPERIMENTINIS TYRIMAS	112
6.1.	EKSPERIMENTO REZULTATŲ APIBENDRINIMAS	113
7.	IŠVADOS	114
8.	LITERATŪRA	116
9.	PRIEDAI	120
9.1.	1 PRIEDAS. ATM SISTEMOS REIKALAVIMŲ IR PA DETALIZAVIMAS	120
9.2.	2 PRIEDAS. DARBŲ PASISKIRSTYMO LENTELĖ	137
9.3.	3 PRIEDAS. STRAIPSNIS	138

Paveikslų sąrašas

1.	PAV. REIKALAVIMŲ INŽINERIJOS PROCESAI [35]	15
2.	PAV. ISO/IEC 9126 VIDAUS IR IŠORĖS KOKYBĖS MODELIS [43].....	34
3.	PAV. FURPS+ MODELIO REIKALAVIMŲ KLASIFIKAVIMO SCHEMA [11]	35
4.	PAV. VOLERE ŠABLONAS FUNKCINIAMS REIKALAVIMAMS APRAŠYTI [39].....	39
5.	PAV. UŽPILDYTOS REIKALAVIMŲ FORMOS PAVYZDYS [39]	40
6.	PAV. CALIBERM ĮRANKIO REIKALAVIMŲ VALDYMO LANGAS	ERROR! BOOKMARK NOT DEFINED.
7.	PAV. REIKALAVIMŲ HIERARCHIJOS LANGAS TOPTEAM™ ANALYST ĮRANKYJE.....	ERROR! BOOKMARK NOT DEFINED.
8.	PAV. NFR ASOCIACIJOS TAŠKAI PANAUDOS ATVEJŲ DIAGRAMOJE[36]	49
9.	PAV. VEIKLOS DIAGRAMA APIBŪDINANTI INTEGRALUMO PROCESĄ [36]	50
10.	PAV. SADT DIAGRAMA [8].....	51
11.	PAV. REIKALAVIMŲ INTEGRALUMO PROCESO MODELIS	56
12.	PAV. PAGRINDINIŲ UŽDUOČIŲ VEIKLOS DIAGRAMA	57
13.	PAV. REIKALAVIMŲ IDENTIFIKAVIMO PROCESAS	58
14.	PAV. PANAUDOJIMO ATVEJŲ APRAŠYMO VEIKLOS MODELIS NAUDOJANT C. LARMAN'Ų ŠABLONĄ.....	64
15.	PAV. REIKALAVIMŲ SPECIFIKAVIMO PROCESAS	65
16.	PAV. REIKALAVIMŲ KOMPONAVIMO PROCESAS.....	68
17.	PAV. BESIKERTANČIO REIKALAVIMO PAVYZDYS	69
18.	PAV. REIKALAVIMŲ INTEGRALUMĄ UŽTIKRINANTIS MODELIS	73
19.	PAV. SUKURTI REIKALAVIMŲ TIPAI REQUISITEPRO ĮRANKYJE	78
20.	PAV. PANAUDOS ATVEJO ATRIBUTAI IR REIŠMĖS	79
21.	PAV. FUNKCINIŲ REIKALAVIMŲ ATRIBUTAI IR REIŠMĖS.....	80
22.	PAV. NEFUNKCINIO REIKALAVIMO „SAUGUMAS“ ATRIBUTAI IR REIŠMĖS	81
23.	PAV. NEFUNKCINIO REIKALAVIMO „VYKDYMO SAVYBIŲ“ ATRIBUTAI IR REIŠMĖS	81
24.	PAV. NUSTATYTA KONTRIBUCIJOS TIPAS REIKALAVIMUI „VYKDYMO SAVYBĖ“	81
25.	PAV. METODO UML PANAUDOS ATVEJO MODELIS.....	83
26.	PAV. PROFILIO STEREOTIPŲ NUSTATYMO MODELIS	83
27.	PAV. RYŠIO TIPO NUSTATYMAI.....	84
28.	PAV. STEREOTIPO „CUSTOMIZATION“ PRISKYRIMAS.....	85
29.	PAV. ELEMENTO NUSTATYMAI PAGAL ŽYMES	86
30.	PAV. PAKETO EKSPORTAVIMAS	86
31.	PAV. DIAGRAMOS KŪRIMO LANGAS.....	87
32.	PAV. PROFILIO PARINKIMAS	87
33.	PAV. ĮRANKIŲ JUOSTOS SUDARYMAS	87
34.	PAV. NAUJO MYGTUKO PRIDĖJIMAS.....	88
35.	PAV. MYGTUKO KŪRIMO LANGAS	88
36.	PAV. SIMBOLIŲ SAVYBIŲ NUSTATYMO LANGAS.....	89
37.	PAV. GREITŲ MANIPULIACIJŲ NUSTATYMO LANGAS	89
38.	PAV. SISTEMOS VARTOTOJAI IR FUNKCIJŲ TARPUSAVIO PRIKLAUSOMYBĖS	93
39.	PANAUDOS ATVEJŲ SPECIFIKAVIMAS REQUISITEPRO APLINKOJE	94
40.	PAV. ATM SISTEMOS PANAUDOS ATVEJŲ SUSIETUMO MATRICĄ.....	96
41.	PAV. PANAUDOS ATVEJŲ ATRIBUTŲ MATRICA	97
42.	PAV. REIKALAVIMO „SAUGUMAS“ SUSKAIÐYMAS Į POTIPIUS.....	97
43.	PAV. FUNKCINIO REIKALAVIMO „TIKRINTI PIN KODĄ“ PRIKLAUSOMYBĖS.....	98
44.	PAV. NEFUNKCINIO PANAUDOJAMUMO REIKALAVIMO TIPO POVEIKIS (TEIGIAMAS IR NEIGIAMAS)	99
45.	PAV. NEFUNKCINIŲ REIKALAVIMŲ ATRIBUTŲ MATRICA FILTRUOTA PAGAL KONTRIBUCIJŲ ĮTAKĄ	99
46.	PAV. FUNKCINIŲ REIKALAVIMŲ IR PANAUDOS ATVEJŲ SAJAJŲ MATRICA	102
47.	PAV. REIKALAVIMŲ SAJAJŲ INTO MEDIS	103
48.	PAV. REIKALAVIMŲ OUT MEDIS.....	104
49.	PAV. NEFUNKCINIAI BESIKERTANTYS IR BŪTINI REIKALAVIMAI	106
50.	PAV. FUNKCINIS BESIKERTANTIS IR BŪTINAS REIKALAVIMAS	106
51.	PAV. PANAUDOS ATVEJO DIAGRAMA „SUVESTI PIN KODĄ“ REIKALAVIMUI	110
52.	PAV. PANAUDOS ATVEJO DIAGRAMA „TIKRINTI KORTELE“ REIKALAVIMUI	111

Lentelių sąrašas

1.	LENTELĖ. REIKALAVIMŲ KOKYBĖS CHARAKTERISTIKOS[40].....	22
2.	LENTELĖ. SRS ŠABLONO CHARAKTERISTIKOS [25].....	37
3.	LENTELĖ. ŠABLONŲ PALYGINIMAS	42
4.	LENTELĖ. REIKALAVIMŲ SPECIFIKAVIMO ĮRANKIŲ PALYGINIMAS	46
5.	LENTELĖ. MOKSLININKŲ DARBAI SUSIJĘ SU REIKALAVIMŲ INTEGRAVIMO METODAIS.....	53
6.	LENTELĖ. VARTOTOJŲ SIEKIAMI TIKSLAI IR PROBLEMAS	54
7.	LENTELĖ. REIKALAVIMŲ SPECIFIKAVIMO ŠABLONAS.....	60
8.	LENTELĖ. ŠABLONAS PANAUDOS ATVEJAMS SPECIFIKUOTI [23]	64
9.	LENTELĖ. REIKALAVIMŲ SVARBA SUINTERESUOTOSIOMS ŠALIMS [28].....	66
10.	LENTELĖ. SĄVEIKOS TAŠKŲ IDENTIFIKAVIMAS	68
11.	LENTELĖ. SUINTERESUOTŲJŲ ŠALIŲ SUSIEJIMAS SU SĄVEIKOS TAŠKAIS.....	68
12.	LENTELĖ. STEREOTIPŲ APIBŪDINIMO LENTELĖ	83
13.	LENTELĖ. PA „ĮDĖTI KORTELĘ“ SPECIFIKAVIMAS	94
14.	LENTELĖ. PA „SUVESTI PIN KODĄ“ SPECIFIKAVIMAS.....	95
15.	LENTELĖ. PA „TIKRINTI KORTELĘ“ SPECIFIKAVIMAS.....	95
16.	LENTELĖ. FUNKCINIO REIKALAVIMO „PRISIJUNGTI PRIE SISTEMOS“ DETALIZAVIMAS	100
17.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „UŽDUOČIŲ VYKDYMO GREITIS“ DETALIZAVIMAS	100
18.	LENTELĖ. FUNKCINIO REIKALAVIMO „KORTELĖS TIKRINIMAS“ DETALIZAVIMAS	101
19.	LENTELĖ. SĄVEIKOS TAŠKŲ LENTELĖ	105
20.	LENTELĖ. GAIRĖS, PA MODELIAVIMUI PAGAL SUDARYTĄ REIKALAVIMŲ ŠABLONĄ	109
21.	LENTELĖ. METODŲ PALYGINIMO LENTELĖ	113
1.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	120
2.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	121
3.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	122
4.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	123
5.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	124
6.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	124
7.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	125
8.	LENTELĖ. FUNKCINIO REIKALAVIMO DETALIZAVIMAS	126
9.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „REIKALAVIMAI PRIEIGAI“ DETALIZAVIMAS.....	127
10.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „KONFIDENCIALUMAS“ DETALIZAVIMAS	127
11.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „VIENTISUMO“ DETALIZAVIMAS	128
12.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „UŽDUOČIŲ VYKDYMO GREITIS“ DETALIZAVIMAS	129
13.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „REIKALAVIMAS TIKSLUMUI“ DETALIZAVIMAS	130
14.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „REIKALAVIMAS TIKSLUMUI“ DETALIZAVIMAS	131
15.	LENTELĖ. NEFUNKCINIO REIKALAVIMO „NAUDOJIMOSI PAPRASTUMAS“ DETALIZAVIMAS	131
16.	LENTELĖ. PA „PASIRINKTI OPERACIJĄ“ SPECIFIKAVIMAS.....	132
17.	LENTELĖ. PA „NETEISINGAS PIN“ SPECIFIKAVIMAS	133
18.	LENTELĖ. PA „IŠIMTI PINIGUS“ SPECIFIKAVIMAS.....	133
19.	LENTELĖ. PA „PERVESTI PINIGUS“ SPECIFIKAVIMAS	134
20.	LENTELĖ. PA „PATIKRINTI BALANSĄ“ SPECIFIKAVIMAS.....	135
21.	LENTELĖ. PA „PASIRINKTI SĄSKAITĄ“ SPECIFIKAVIMAS.....	135
22.	LENTELĖ. PA „SPAUSDINTI ČEKĮ“ SPECIFIKAVIMAS	135

Summary

A method for integrity support requirements specification

The aim of this work is to create a requirements specification integrity method. For this purpose a process model was developed, which consists of three main tasks: requirements identification, specification and composition. The first two tasks help to decrease the dispersal of crosscutting requirements in requirements specification, establish the required requirements and dependencies. The goal of the composition task is to compose the requirements to give the developer a view of the whole system and to identify and manage conflicts between requirements. Integrity of requirements is ensured by keeping all requirements in a single form.

The paper analyzes the identification of crosscutting requirements and gathering strategies. Templates of requirements specification, requirements of standards classification and requirements management tools were analyzed. Methods and techniques of functional and non-functional requirements for the integrity were analyzed.

The unique template for requirements specification was created, which includes functional and non-functional requirements, functional and non-functional crosscutting requirements. The template for requirements management was made in RequisitePro environment.

The MagicDraw profile package was created, which provides the whole system view for a developer, representing requirements sets for use case diagrams.

Requirements integrity method was tested through the ATM system requirements specification.

Keywords: functional and non-functional requirements, required requirements, crosscutting requirements, requirements specification, requirements specification template, RequisitePro tool, UML

Terminų ir santrumpų žodynas

NFR – nefunkcinis reikalavimas

FR – funkcinis reikalavimas

UML (angl. *Unified Modeling language*) – unifikuota modeliavimo kalba

PĮ – programinė įranga

IS – informacinė sistema

IT – informacinės technologijos

SRS (angl. *Software Requirements Specification*) – Programinės įrangos reikalavimų specifikacija

FURPS+ - reikalavimų modelis, apibrėžiantis vartotojo reikalavimus sistemai

URS (angl. *User Requirements Specification*) – vartotojo reikalavimų specifikacija

Rational RequisitePro - programinė įranga leidžianti kurti reikalavimus, naudojant panaudos atvejus.

DOORS (angl. *Dynamic Object Oriented Requirements System*) – dinaminis objektais orientuota reikalavimų sistema.

LEL (angl. *Language Extended Lexicon*) – išplėstas kalbos leksikonas

ER modelis (angl. *Entity Relation*) – Esybių ryšių modelis

OO (angl. *Object oriented*) modelis – objektais orientuotas modelis

SADT (*Structured Analysis and Design Technique*) – struktūros analizė ir diegimo technika.

Globalinis NFR (*Global*) - NFR yra susijęs su visa sistema.

Dalinis NFR (*Partial*) - šiuo atveju NFR yra siejamas su kai kuriais funkciniais reikalavimais.

AORE (angl. *Aspect-Oriented Requirements Engineering (AORE)*) - Aspektais grindžiama reikalavimų inžinerija.

AORA (angl. *Aspect-Oriented Requirements Analysis*) – Aspektais grindžiama reikalavimų analizė.

1. ĮVADAS

Informacinės technologijos šiuolaikiniame pasaulyje daro didelę įtaką, visi procesai tampa kompiuterizuojami. Kompiuterizacijos procesui didelę įtaką turi tinkamos programinės įrangos kūrimas bei naudojimas.

Reikalavimų inžinerija susideda iš veiklų, kuriomis grindžiama ką plėtoti ir kam to reikia. Rašant projektų specifikacijas, siekiama valdyti pokyčius šiuose aprašymuose ir užtikrinti nuoseklius ir tinkamus reikalavimus. Reikalavimų inžinerijos tikslas yra valdyti reikalavimus, tam kad reikalavimų inžinierius galėtų kurti ir plėtoti užsakovo poreikius tenkinančią programinę įrangą [34, 39]. Sėkmingo projekto pagrindas - aiškiai suvokiami reikalavimai. Tokie reikalavimai aprašomi ankstyvoje produkto kūrimo proceso stadijoje ir jie atsispindi visame projekto gyvavimo cikle.

Nepakankamas vartotojo įtraukimas, neišsamūs reikalavimai ir besikeičiantys reikalavimai yra pagrindinė nesėkmių priežastis programinės įrangos projektų kūrime, atsižvelgiant į užsakovo poreikius, laiką ir biudžeto viršijimą [12, 29, 34]. Reikalavimų inžinerijoje stengiamasi išspręsti šią problemą. Todėl šiame darbe siekiama sudaryti reikalavimų specifikacijos integralumą užtikrinantį metodą, remiantis AORA (angl. *Aspect-Oriented Requirements Analysis*) metodu ir Volere šablonu. Siekiant užtikrinti reikalavimų specifikacijos integralumą būtina sudaryti sąsajas tarp reikalavimų, identifikuoti ir specifikuoti besikertančius reikalavimus bei sudaryti reikalavimų rinkinius, kuriuos būtų galima ištraukti įvairiais pjūviais. Tik tikslus reikalavimų specifikavimo proceso valdymas leis sėkmingai projektuoti, realizuoti ir įdiegti informacines sistemas kompiuterizuojamose srityse.

Kuriamas metodas turi koncentruotis į besikertančius reikalavimus bei į reikalavimų rinkinių sudarymą. Besikertančių reikalavimų nustatymas, reikalavimų specifikavime padeda išvengti dviprasmybių bei reikalavimų pasikartojimų. Sudaryti reikalavimų rinkiniai, padeda spręsti konfliktus ankstyvoje sistemos kūrimo stadijoje. Tam, kad aprašyti tokį metodą, atliekama panašių metodų, šablonų analizė ir pasirenkamas konkretus šablonas, kurį būtų galima išplėsti.

Tyrimo tikslas – sukurti reikalavimų integralumą užtikrinantį metodą, sudarant reikalavimų rinkinius reikalavimų specifikacijoje, taip užtikrinant reikalavimų vientisumą.

Metodas sudaromas remiantis AORA (angl. *Aspect-Oriented Requirements Analysis*) metodu ir pasirinkto specifikavimo šablono pagrindu.

Tyrimo uždaviniai:

- Atlikti reikalavimų valdymo įrankių RequisitePro, Doors, CalimberRM, TopTeamTM analizę ir pasirinkti tinkamą įrankį metodo realizavimui;
- Reikalavimų specifikacijos savybių ir vertinimo kriterijų analizę;
- Atlikti reikalavimų specifikavimo šablonų analizę, pasirinkti ar papildyti šabloną, kuris išpildytų reikalavimų specifikavimo integralumo užtikrinimo metodą;
- Atlikti reikalavimų klasifikavimo FURPS ir ISO/IEC 9126 standartų analizę;
- Reikalavimų integralumo metodų analizę;
- Išbandyti metodo taikymą pasirinktai informacinei sistemai;

Šį darbą sudaro įvadas, šeši pagrindiniai skyriai, išvados, literatūros sąrašas ir trys priedai.

Antrame darbo skyriuje nagrinėjama aspektais grindžiama reikalavimų analizė. Aptariamos reikalavimų gavybos strategijos, bei reikalavimų specifikacijos kriterijai. Atlikta besikertančių reikalavimų ir jų išgavimo strategijų analizė. Išanalizuoti reikalavimų specifikavimo šablonai, bei reikalavimų valdymo įrankiai. Atlikta FR ir NFR reikalavimų integralumo metodų analizė, analizuojami moksliniai darbai susiję su darbo tema.

Trečiame skyriuje pateikta modelio architektūra, bei specifikavimo procesų modeliai. Integralumo modelis projektuojamas bei sudaromas remiantis analitinės dalies medžiaga. Modelis atvaizduotas hierarchine notacija, kurioje kiekviena veikla turi hierarchinę tvarką.

Ketvirtas darbo skyrius aprašo pasirinktų priemonių išplėtimą. Reikalavimų valdymui sudarytas šablonas RequisitePro įrankiu. Sukurtas šablonas, atsižvelgiant į reikalavimų tipus ir jų atributus naudojamus metodikoje. Sukurtas UML profilis, siekiant atvaizduoti reikalingus metodikos atributus ir ryšius.

Penktame skyriuje pateikiamas metodo realizavimas pasirinktomis ir sukurtomis priemonėmis. Reikalavimų valdymui ir specifikavimui naudojamas sukurtas šablonas RequisitePro aplinkoje. Ryšiams tarp reikalavimų atvaizduoti naudojamos atributų ir susietumo matricos, susietumo medžiai. Filtrų pagalba filtruojame informaciją pagal reikalavimų charakteristikas. Reikalavimų rinkiniai ir ryšiai tarp skirtingų reikalavimų tipų atvaizduojami UML panaudos atvejų diagramomis.

Šeštame skyriuje pateikiami eksperimento rezultatai iš penktame skyriuje atlikto metodo pritaikymo dalykinei sričiai.

Darbo prieduose pateikiama: pirmame priede pateikiami reikalavimų ir panaudos atvejų detalizavimas; antrame priede pateiktas darbų pasiskirstymas, trečiame priede pateiktas darbo tema parašytas straipsnis.

2. REIKALAVIMŲ, REIKALAVIMŲ INTEGRALUMO METODŲ, SPECIFIKAVIMO ŠABLONŲ IR REIKALAVIMŲ VALDYMO ĮRANKIŲ ANALIZĖ

2.1. Analizės tikslas

Reikalavimų, reikalavimų integravimo metodų, specifikavimo šablonų ir reikalavimų valdymo įrankių analizės tikslas:

- Atlikti reikalavimų ir reikalavimų gavybos strategijų analizę;
- Atlikti besikertančių reikalavimų identifikavimo ir išgavimo strategijų analizę;
- Išanalizuoti esamus reikalavimų integralumo metodus;
- Atlikti reikalavimų specifikavimo šablonų, klasifikavimo standartų bei reikalavimų valdymo įrankių analizę;

2.2. Tyrimo sritis, objektas ir problema

Tyrimo sritis - apima reikalavimų valdymo įrankių, šablonų analizę bei reikalavimų specifikacijos integralumo metodus. Tiriama metodai, jų pritaikymas praktikoje.

Tyrimo objektas – reikalavimų integralumą užtikrinantys metodai ir priemonės.

Pagrindinė problema - funkcinių ir nefunkcinių reikalavimų gausa, bei reikalavimų išsamumo stoka. Neišsamūs reikalavimai dažniausiai atsiranda neišskiriant jų į atskirus modulius ar dėl neišsamaus ir netikslaus reikalavimo specifikavimo, dėl tokių reikalavimų atsiranda grėsmė galutiniam projektui. Stipriai susiję sistemos reikalavimai turi būti realizuojami vienoje sistemos dalyje. Nesusiję, arba tokie reikalavimai, kurie gali būti atskirti, turi būti realizuoti skirtingose sistemos dalyse. Todėl būtina specifikuoti reikalavimus sudarant rinkinius, kurie nusakytų ir apibrėžtų visapusiškai tik vieną sistemos funkciją. Siekiant užtikrinti reikalavimų išsamumą būtinas integralumo metodas ir tinkamas reikalavimų surinkimo ir specifikavimo procesas. Tokia integrali specifikacija užtikrintų ne tik specifikacijos kokybę bet ir palengvintų projektuotojams projektuoti ir plėtoti sistemą.

2.3. Reikalavimų inžinerija

Reikalavimų inžinerija – tai procesai, naudojami sistemos reikalavimų išgavimui, analizei ir atestavimui [21]. Reikalavimų inžinerija yra vienas svarbiausių žingsnių sistemos kūrimo

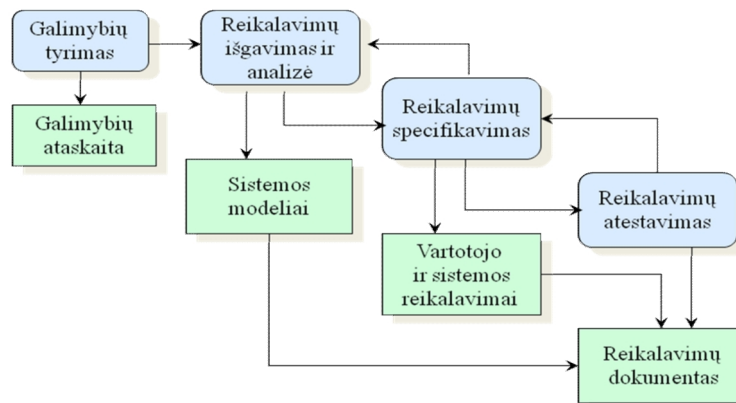
procesu. Reikalavimų išgavimo procesas leidžia išsiaiškinti ir surinkti vartotojo keliamus reikalavimus reikalingus sistemai. Pirmiausia, analizuojami sistemos dalyviai [21].

Yra trys svarbiausios dalyvių kategorijos: kūrėjas, vartotojas ir klientas. Į reikalavimų išgavimo procesą įtraukiami ir kiti dalyviai, pavyzdžiui, advokatai, standartai, organizacijos ir kt..

Reikalavimų inžinerija susiduria su tokiomis problemomis:

1. Klientai nesupranta, kas įmanoma techniškai;
2. Reikalavimų pakitimas, atsižvelgiant į sistemos galimybes;
3. Klientas ne pilnai išdėsto, ko jis norėtų;
4. Vėliau iškilę reikalavimai gali prieštarauti su jau įgyvendintais reikalavimais;
5. Prieš projektuojant sistemą sunku apibrėžti visus reikalavimus ir jų korektiškumą.

Reikalavimų inžinerijos procesai pateikiami pagal Sommerville [21,34] 1 paveiksle :



1. pav. Reikalavimų inžinerijos procesai [34]

2.4. AORE inžinerija

Aspektais grindžiama reikalavimų inžinerija (*Aspect-Oriented Requirements Engineering (AORE)*) užpildo spragą tradiciniuose reikalavimų inžinerijos metoduose, naudojant sistemingas priemones identifikuojant, modeliuojant, atvaizduojant ir komponuojant besikertančius reikalavimus tiek funkcinis, tiek nefunkcinis. Aspektais grindžiama reikalavimų analizė (AORA) tai metodas, kuris besikertančius ir nesikertančius reikalavimus iškelia į reikalavimų lygį [34, 39].

Aspektais grindžiama reikalavimų inžinerija (*Aspect-oriented requirement Engineering*) apima besikertančių savybių identifikavimą ir specifikavimą reikalavimų lygmenyje [28]. Pavyzdžiui, tokias savybes turi saugumo, mobilumo, prieinamumo ir realaus laiko apribojimai.

Besikertančias savybes turi - reikalavimai, panaudojimo atvejai, kurie plačiai apima ir įtakoje kitus reikalavimus ar architektūrinius komponentus. Su besikertančiais reikalavimais susiduriama didelės apimties specifikacijos dokumentuose. Dokumentai, tokie, kaip interviu, užrašai yra labai netikslūs, pilni prieštaravimų ir juose trūksta svarbios informacijos. Be to, besikertantys reikalavimai yra „išsibarstę“ visame dokumente, todėl jų identifikavimas labai sunkus. Blogiausia tai, kad kartais panašūs reikalavimai pasirodo skirtingose dokumento dalyse, tik aprašomi kitais žodžiais [28].

2.5. Reikalavimų analizė

Reikalavimas – tai būtinas sistemos atributas. Reikalavimus priimta išreikšti sakiniiais: identifikuojančiais sistemos gebėjimus, charakteristikas ar kokybės faktorius, teikiančius naudą galutiniams sistemos vartotojams [41].

Programinės įrangos pramonėje priklausomai nuo požiūrio į reikalavimą, apibrėžimas gali būti skirtingas. Pavyzdžiui, iš galutinio sistemos vartotojo požiūrio, reikalavimą galima apibūdinti kaip aukšto lygio produkto koncepciją programuotojui. Tuo tarpu programuotojas reikalavimą gali apibūdinti kaip detalų vartotojo sąsajos aprašymą [39].

IEEE terminologijos žodyne reikalavimas apibrėžiamas kaip:

- Sąlyga arba gebėjimas reikalingas vartotojui tam tikrai problemai išspręsti arba tam tikram tikslui pasiekti;
- Sąlyga, kurią turi įvykdyti arba gebėjimas, kurį turi turėti sistema arba sistemos komponentė, kuri patenkintų formaliai aprašytą specifikaciją, kontraktą ar standartą.
- Dokumentuota sąlygos arba gebėjimo pateikimas. Šie apibrėžimai apima tiek vartotojo, tiek programuotojo požiūrius į reikalavimą.

Dar vieną reikalavimo apibrėžimą pateikia Sommerville'is ir Sawyer'is [34]:

- Reikalavimai yra specifikacija to, kas turi būti suprogramuota. Tai aprašymas kaip sistema turi elgtis (gebėjimas) arba nesielti (apribojimas).

2.5.1. Besikertančių reikalavimų analizė

Gerai parašytai reikalavimų specifikacijai yra būdinga tai, kad kiekvienas reikalavimas nėra sumaišytas su kitais reikalavimais, bet atitinka tiksliai vieną reikalavimą. Pavyzdžiui, S. Robertson ir J. Robertson [33] rekomenduoja kiekvieną reikalavimą aprašyti vienu sakiniu, išvengiant sudėtinių sakinių, nes tokie sakiniai gali turėti daugiau nei vieną reikalavimą.

Be to, geras atsekamumas yra kritinis kokybės faktorius, kurį užtikrina tik gerai parašyta reikalavimų specifikacija [10]. Tai svarbi sąlyga prieš sudarant tikslius ryšius, kurie paprastai yra tarp skirtingų reikalavimų. Pavyzdžiui, tokius ryšius apima tarpusavio priklausomybės, detalizavimo, daliniai / pilnumo ryšiai (pvz., [30]).

Specialus ryšio tipas tarp reikalavimų gali nusakyti besikertantį reikalavimą. Reikalavimas A kerta reikalavimą B, jei programinės įrangos dekompozicijoje buvo nustatyta, kad B negali būti įvykdytas neįtraukiant reikalavimo A. Taigi projektavimo elementas ar kodas patenkina reikalavimą B jei išpildomas ir reikalavimas A. Reikalavimai, kurie kerta kitus reikalavimus vadinami besikertančiais reikalavimais (šiuo atveju reikalavimas A). Be to, išsireiškimas „besikertančio įtaka“ yra naudojama kaip sinonimas tarp dviejų reikalavimo ryšių, kuris yra nustatytas vieno besikertančio reikalavimo kitam. Jei pasakyta, kad reikalavimas A turi įtakos reikalavimui B, jis papildomai žymi ryšio kryptį, kad A kerta B.

2.5.2. Besikertančių reikalavimų identifikavimas

E. Baniassad pasiūlė metodą kaip surinkti besikertančius reikalavimus parengtuose reikalavimų dokumentuose [3]. Šis metodas pradedamas tiriant dekompoziciją reikalavimų dokumentuose, nes dekompozicijos supratimas yra svarbiausia sąlyga nustatant besikertančius reikalavimus. Šis metodas susideda iš keturių etapų – besikertančių reikalavimų identifikavimas reikalavimuose, surenkami besikertantys reikalavimai kaip reikalavimų aspektai, aprašomi kompozicijos taisyklėmis ir aspektų analizė. Besikertančių reikalavimų identifikavo etape, reikalavimų kūrėjas analizuoja reikalavimų dokumentus, tokius kaip programinės įrangos specifikacijas, ieškant besikertančių reikalavimų. Aspektų terminai paprastai aprašo kokybės atributus sistemai, tokius kaip saugumas ir efektyvumas, kurie yra išsibarstę ir pakartotinai aprašyti keliose vietose visame reikalavimų dokumente, yra dažniausiai laikomi besikertančiais reikalavimais. Besikertančių reikalavimų etape E. Baniassad [3] siūlo, kad reikalavimų kūrėjai pertvarkytų reikalavimų dokumentą tam tikra tvarka, siekiant, kad kiekvienas reikalavimas, įskaitant kiekvieną besikertantį reikalavimą, būtų fiksuojami atskiruose reikalavimų artefaktuose, pavyzdžiui, skyriuose ar pastraipose, taip sumažinant pasikartojimą. Įtaka vieno besikertančio reikalavimo kitam reikalavimui, pavyzdžiui, vienas reikalavimas apriboja kito reikalavimo vykdymą, aiškiai apibūdina kompozicijos taisyklės trečiame etape. Kompozicijos taisyklės gali būti apibrėžtos analogiškai kaip ir „pointcut“ specifikacijoje AOP. Paskutiniame analizės etape,

identifikuoti besikertantys reikalavimai yra vertinami, siekiant įvertinti jų taikymo įtaka, įvertinant jų galimus konfliktus tarpusavyje ir pateikti pagrįstą kompromisą sprendimui.

Du tipai atvaizduojant besikertančius reikalavimus pateikti E. Baniassad [3] – besikertantys reikalavimai, kurie specifikuojami išsibarsčiusiu būdu ir besikertantys reikalavimai, kurie specifikuojami nekartojant jų. Reikalavimų pavyzdys, kuris paimtas iš Baniassad [3]:

1. Palūkanų mokėjimas tam tikru procentu nuo kiekvienos sąskaitos, įsitikinkite, kad operacija yra pilnai įvykdyta ir saugomos audito istorijoje.
2. Leisti vartotojams išimti pinigus iš jų sąskaitos, įsitikinkite, kad operacija yra pilnai įvykdyta ir saugomos audito istorijoje.

Du besikertantys reikalavimai yra atvaizduoti šiame pavyzdyje – „kad operacija yra pilnai įvykdyta“ ir „laikoma audito istorijoje“. Šie du besikertantys reikalavimai yra specifikuojami išsibarsčiusiu būdu.

Neišsibarsčiusiu būdu, šiame pavyzdyje, kuris taip pat pateiktas E. Baniassad [3], išgaunamas taip:

1. *Palūkanų mokėjimas tam tikru procentu nuo kiekvienos sąskaitos.*
2. *Leisti vartotojams išimti pinigus iš jų sąskaitos.*
3. *Visos operacijos yra pilnai įvykdytos.*
4. *Visos operacijos saugomos audito istorijoje.*

Šiame pavyzdyje, „operacija yra pilnai įvykdyta“ ir „saugoma audito istorijoje“ specifikuojami moduliavimo būdu – jie specifikuojami atskirame skyriuje. Taip sumažinamas reikalavimų išsibarstymas reikalavimų dokumentuose, tačiau šie du besikertantys reikalavimai daro įtraukimo poveikį. „Visos operacijos“ negali būti nustatytos, kol dokumentas neužbaigtas. Be to, sąsaja tarp šių dviejų reikalavimų (*operacijų įvykdymas ir auditas*) ir kitų dviejų pagrindinių reikalavimų (*palūkanų mokėjimas ir pinigų išėmimas*) tampa įtraukiami taip pat, todėl konfliktų atskleidimas ir kompromisų sąlygų aptarimas tarp jų yra labai sunkus. Tai galima pamatyti anksčiau minėtame pavyzdyje „operacijos yra pilnai įvykdytos“ ir „saugomos audito istorijoje“ prieštarauja vienas kitam, jei jie konfliktuoja, jų konfliktai nežinomi, kol visos operacijos neidentifikuotos.

Greitas ir tikslus neišsibarsčiusių besikertančių reikalavimų specifikavimas, išlaikant besikertančius reikalavimus tuo pačiu metu, E. Baniassad [3] pasiūlė AO sprendimą, kuris registruoja besikertančius reikalavimus intuityviu ir moduliavimo būdu [3]. Šiuo sprendimu,

kiekvienas besikertantis reikalavimas yra specifikuojamas skirtinguose skyriuose – vienas reikalavimams ir kitas reikalavimų poveikiams. Naudojamas pavyzdys iš E. Baniassad [3]:

1. *Palūkanų mokėjimas tam tikru procentu nuo kiekvienos sąskaitos.*
2. *Leisti vartotojams išimti pinigus iš jų sąskaitos.*
3. *Visos operacijos yra pilnai įvykdytos.*

Operacijų sąrašas apima (1) ir (2).

4. *Visos operacijos saugomos audito istorijoje.*

Operacijų sąrašas apima (1) ir (2).

Tekstas pastorintu šriftu yra besikertančių reikalavimų poveikių specifikacija. Pastebėta, kad be sumažinto išsibarstymo, taip pat nustatoma įtaka besikertančių reikalavimų vienas kitam, kurie yra specifikuojami pateiktu metodu. Iš ankstesnių pavyzdžių, pastebėta, kad E. Baniassad [3] pasiūlė reikalavimų dokumento struktūrą, taigi besikertantys reikalavimai yra surenkami atskirai ir jų numatomi poveikiai yra specifikuojami.

Pateiktas Baniassad metodas [3] daro prielaidą, geram reikalavimų organizavimui dokumentuose, kuriuose reikalavimai yra atskirai specifikuojami savo dalyse reikalavimų artefaktuose, pavyzdžiui, atskiruose skyriuose, pastraipose. Tačiau, toks reikalavimų dokumentas dažnai nepasiekiamas. Ši prielaida apribojama tam tikru taikymu. Be to, nagrinėjant reikalavimų artefaktus, dominuojančių dekompozicijos supratimui, besikertančių reikalavimų ieškojimas ir reikalavimų artefaktų restruktūrizavimas yra sunkus, daug pastangų reikalaujantis ir į klaidas linkęs darbas.

Atsižvelgiant į nefunkcinius reikalavimus, funkciniai reikalavimai gali tapti besikertančiais reikalavimais taip pat [3]. Pavyzdžiui, sistema, kuri susideda iš kelių paslaugų mazgų ir kiekvienas paslaugų mazgas teikia tam tikras paslaugas, pavyzdžiui, skambučio priėmimo paslaugas tik tiems, kurie priklauso tam tikrai grupei. Šioje sistemoje, funkcinis reikalavimas – patikrinti kuriai grupei skambinanties asmuo priklauso – yra besikertantis reikalavimas, kuris apima visus paslaugų mazgus. Panašiai, kaip ir banko sistema, audito istorijoje saugomos visos vartotojo operacijos [3], taip pat yra funkcinis besikertantis reikalavimas. Egzistuojantiems AORE metodams taip pat trūksta priemonių išgaunant funkcinis besikertančius reikalavimus.

2.5.3. Besikertančių reikalavimų išgavimo būdai ir strategijos

L. Rosenhainer [22] pasiūlė du būdus, kurie gali būti naudojami identifikuojant besikertančius reikalavimus reikalavimų lygyje, programinės įrangos reikalavimų specifikacijoje. Vienas iš būdų ieškoti besikertančių reikalavimų, reikalavimų tikrinimo metu rankiniu būdu. Šiuo būdu, nefunkciniai reikalavimai pasirenkami kaip pradžios taškai. Rankinis paieškos būdas vyksta per visą reikalavimų specifikaciją, siekiant rasti reikalavimus, kurie yra apriboti reikalavimo. Jeigu jis rastas, šis reikalavimas pažymimas kaip besikertantis reikalavimas. Todėl būtų paprasta peržiūrėti specifikaciją konkrečiu aspektu, identifikuojant besikertančius reikalavimus ir jų įtakas mintyse. Rekomenduojama pradėti nuo konkrečių nefunkcinių (tikriausiai ir besikertančių) reikalavimų, tarkim - atsako laiko ir tada ieškoma tekste fakto, kuriuose paminėtas sistemos atsakas. Jeigu tokio fakto nebuvo rasta tikrintojas pažymi „nėra“. Kai patikrinimas baigtas surinkti faktai turi būti išanalizuoti, ar jie atspindi reikalavimų apribojimus, pagal atsako laiko reikalavimą. Jei taip, besikertančio reikalavimo poveikis buvo rastas ir turi būti dokumentuotas. Jeigu bent vieno besikertančio reikalavimo įtaka buvo rasta, tada atsako laiko reikalavimas pažymimas kaip besikertantis reikalavimas.

Kitas būdas naudojant informacijos paieškos programą, ieškant besikertančių reikalavimų pusiau automatinio būdu. Šiuo būdu, informacijos paieškos programa naudojama ieškant visame dokumente iš anksto nustatytų sakiniu ar terminu, kurie kaip manoma gali būti besikertantys reikalavimai. Nors paieška gali būti automatizuota dalinai, tačiau išnagrinėti kiekvieną besikertančio reikalavimo faktą, atliekamas rankiniu būdu. Programa, pavyzdžiui, gali rasti faktus, kurie gali būti paveikti reikalavimų, analitikas nutaria ar tai besikertantis reikalavimas. Panašiu būdu, kaip aptarta ankščiau, analitikas privalo išnagrinėti kiekvieną faktą ir dokumentą, ar besikertančio reikalavimo įtaka rasta. Registruojant bent vieną įtaką reikalavimas charakterizuojamas kaip besikertantis. Taigi abiem būdams reikia intensyvaus rankinio darbo.

Šie pasiūlyti būdai turi defektų. Pirma, jie pagrįsti prielaida, kad besikertantys reikalavimai kilę iš nefunkcinių reikalavimų, kurie gali būti nustatomi rankiniu būdu pasirenkant juos kaip pradžios tašką arba taikomi išankstiniai sakinių ar terminų rinkiniai. Net naudojant kartu abu būdus, jie neras visų besikertančių reikalavimų, reikalavimų specifikacijoje. Tai iš dalies lemia, kad tiek nefunkciniai tiek funkciniai reikalavimai gali būti besikertantys reikalavimai [3]. Be to, atžvelgiant į besikertančius reikalavimus jie gali būti išreiškiami skirtingai, skirtinguose reikalavimų specifikacijos vietose. Neįmanoma sukurti išankstinių sakinių ar terminų rinkinių, kurie apimtų visas besikertančių reikalavimų posakių formas, kurios gali

pasirodyti. Abu būdai reikalingi nustatant visus galimus aspektus, kurie gali būti besikertantys. Jeigu besikertantis reikalavimas negali tapti pradžios tašku, tai jis negali būti identifikuojamas kaip besikertantis reikalavimas, naudojant vieną iš šių būdų. Be to, kadangi abu būdai įtraukia daug rankinio darbo, jų efektyvumas yra probleminis. Tarkim, naudojamas pirmas būdas. Jeigu turite surinktus 5 nefunkcinius reikalavimus kaip pradžios taškus, jūs turėsite eiti per reikalavimų specifikaciją penkis kartus – po kartą nuo pradžios taško. Realus pasaulio reikalavimų specifikacijoje dažnai yra kur kas daugiau negu 5 nefunkciniai reikalavimai. Šis būdas akivaizdžiai reikalauja daug darbo ir laiko, Rosenhainer [22] nepateikia jokių įrodymų apie šių metodų efektyvumą.

Be pasiūlytų dviejų metodų, kurie aptarti aukščiau, keturi etapai kuriais identifikuojami besikertantys reikalavimai yra įmanomas nustatant pagal Rosenhainer [22]. Tai yra:

1. Besikertančių reikalavimų identifikavimas reikalavimų modeliavimo metu. Kai reikalavimai jau išgauti juos reikia modeliuoti. Modeliavimas yra analizavimo veikla, kurios tikslas - susisteminti reikalavimus, identifikuojant ir nustatant priklausomybes ir kitus ryšius tarp reikalavimų, pašalinant neatitikimus bei nustatant dviprasmybes ir neaiškumus. Taip pat galima identifikuoti ir modeliuoti besikertančius reikalavimus ir jų įtaką šiame etape [3].
2. Besikertančių reikalavimų identifikavimas tuo pat metu kai rašoma PĮ reikalavimų specifikacija. Specifikacijos rašymo metu seka reikalavimų modeliavimas. Tačiau, dažnai nutinka, kad specifikacija parašoma iškart po reikalavimų išgavimo taip išvengiant modeliavimo etapo. Šiuo atveju, rašant specifikaciją taip pat yra analitinis procesas. Akivaizdu, kad jis turėtų taip pat apimti ir besikertančių reikalavimų identifikavimą ir dokumentavimą [3].
3. Besikertančių reikalavimų identifikavimas iš egzistuojančių PĮ reikalavimų specifikacijos. L. Rosenhainer [22] pabrėžia, kad šiame etape yra daugybė egzistuojančių PĮ reikalavimų specifikacijų, kurios sudarytos nenagrinėjant besikertančių reikalavimų, ir aspektais grindžiama reikalavimų inžinerijos metodų (1) ir (2) etapuose, nebus naudojamos artimiausiu metu. Tačiau, abejotina, kad besikertančių reikalavimų gavyba iš egzistuojančių dokumentų gali būti produktyvus atsižvelgiant į sąnaudas ir naudą, ypač kai besikertančių reikalavimų gavyba iš esamų reikalavimų specifikacijų, kurios yra sudarytos nesuprantant besikertančių reikalavimų. Jeigu šios specifikacijos sudarytos be aspektais grindžiamo mąstymo, o tai tikėtina nėra palankus aspektų gavybai, pavyzdžiui,

aprašant reikalavimą naudojant skirtingus terminus ir frazes skirtingose vietose, todėl aspektų išgavimas yra sunkesnis, nei jis galėtų būti.

4. Besikertančių reikalavimų identifikavimas, jų aptikimas vystymosi procese. Žinoma, geriausiai identifikuoti ir dokumentuoti besikertančius reikalavimus ir jų poveikį, kai jie atsiranda, tai yra reikalavimų analizės metu, faktas tas, kad egzistuoja daugybė parašytų reikalavimų specifikacijų, kuriose neįtraukiami besikertantys reikalavimai. Tačiau, vėliau tokie reikalavimai gali tapti pageidaujami kūrėjų, kurie įtraukiami ir identifikuojami dokumentuose [3].

2.5.4. Reikalavimų kokybės charakteristikos

Norint užtikrinti reikalavimų kokybę reikia sudaryti kontrolinius sąrašus ir anketas. Beveik kiekvienoje specifikacijoje galima rasti daug reikalavimų su trūkumais, tai yra blogai nustatyti reikalavimai. Reikalavimai yra neapibrėžti, neišsamūs, nenuoseklūs, neteisingi, neįmanomi, netinkami naudoti arba nepatikrinami (netestuojami).

Kokybiški reikalavimai turi atitikti tokias charakteristikas, 1 lentelė.

1. lentelė. Reikalavimų kokybės charakteristikos[39]

Kokybės charakteristikos	Savybės
Darnumas <i>(Cohesiveness)</i>	<p>Kiekvienas reikalavimas turi būti darnus, sanglaudos tipas priklauso nuo skirtingų specifikuojamų reikalavimų tipų.</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas nurodo tik vieną aplinkybę? • Ar visos reikalavimų dalys vientisos? • Ar visi reikalavimų duomenys įtraukiami į tą pačią duomenų abstrakciją? • Ar visos funkcinių reikalavimų dalys, įtraukiamos tos pačios funkcijų abstrakcijos? • Ar visos sąsajos reikalavimų dalys įtraukiamos tos pačios sąsajos. • Ar visos kokybės reikalavimų dalys įtraukiamos to paties kokybės faktoriaus.
Pilnumas/Išsamumas	Reikalavimų specifikacija turi būti išsami, joje pateikiami visi

Kokybės charakteristikos	Savybės
(Completeness)	<p>svarbūs reikalavimai ir papildoma medžiaga (pvz., kaip jos šablone, turinyje ar formos standarte), taip ir kiekvienas reikalavimas turi būti pilnas/išsamus. Tai dažna problema, nes šios srities ekspertas, kuris nustato reikalavimus dažnai tam tikrą suprantamą informaciją praleidžia, net jei dėl jo kyla neaiškumų kitam užsakovo reikalavimui.</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas išsamus, be trūkstamos informacijos? • Ar kiekvienas reikalavimas turi visą reikiamą informaciją? Pavyzdžiui, ar reikalavimai įtraukia visas susijusias sąlygas, tokias kaip, susijusias būsenas, komponentus? • Ar kiekvienam reikalavimui reikalingas tolimesnis klasifikavimas ir plėtojimas? • Ar kiekvienam reikalavimui pateikta pakankamai informacijos, kad būtų išvengiama dviprasmybių? • <i>Jeigu reikalavimas neturi prieštaravimų, tada jis nurodomas kaip pilnas ir išbaigtas</i> • Ar kiekvienas reikalavimas faktiškai yra tik vienas reikalavimas, o ne sudėtinis reikalavimas?
Nuoseklumas (Consistency)	<p>Dėl reikalavimų nenuoseklumo rinkiniuose jų neįmanoma įgyvendinti, todėl kiekvienas reikalavimas turi būti nuoseklus:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas išoriškai atitinka dokumento šaltinius, tokius kaip aukšto lygio tikslus (<i>higher-level goals</i>) ir reikalavimus? • Ar kiekvienas reikalavimas išoriškai atitinka su visais kitais keliamais reikalavimais to paties tipo arba reikalavimais specifikacijoje? Pavyzdžiui, du reikalavimai turi neprieštarauti nors jie aprašyti tomis pačiomis sąvokomis

Kokybės charakteristikos	Savybės
	<p>tik skirtingais žodžiais.</p> <ul style="list-style-type: none"> • Ar suderinamos kiekvieno reikalavimo dalys nuosekliai?
Teisingumas (Correctness)	<p>Netaisyklingi reikalavimai gali atsirasti dėl architektūros, diegimo ir įgyvendinimo atitinkamų rezultatų. Taigi kiekvienas reikalavimas turi būti teisingas:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas semantiškai teisingas? <p>Ar kiekvienas reikalavimas patenkina visus ar dalį realius poreikius, susijusius su suinteresuotais asmenimis?</p> <p>Ar kiekvienas reikalavimas aiškiai išdėsto veiklos tikslus ir tikslus dokumente?</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas aiškiai išdėsto aukšto lygio reikalavimus? • Ar visi reikalavimų asociacijos numeriai yra teisingi? • Ar kiekvieno reikalavimo sintaksė yra teisinga? • Ar kiekvienas reikalavimas naudoja tinkamą standartinį formatą? • Ar visi žodžiai reikalavimams parašyti be klaidų?
Vartoseną (Currency)	<p>Dažnai reikalavimų specifikacija nėra atnaujinama, kai reikalavimai pakeičiami. Taip pat nėra atnaujinama projektavime, todėl keičiasi pagrindiniai reikalavimai. Taigi kiekvienas reikalavimas neturėtų tapti nebevertinamu:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas reikalingas? • Ar kiekvienas reikalavimas nebuvo nebevertotinas (pvz., dėl besikeičiančių verslo tikslų)?
Užsakovo ir vartotojo poreikiai (Customer/User Orientation)	<p>Labai dažnai, reikalavimai (ypač gauti reikalavimai) yra pateikti kūrėjų, kurie naudoja techninį žargoną, kurių nesupranta užsakovai. Tačiau individualūs reikalavimai turėtų būti orientuoti į kliento ir vartotojo poreikius, jie turi būti suprantami ir validuojami:</p>

Kokybės charakteristikos	Savybės
	<ul style="list-style-type: none"> • Ar kiekvienas reikalavimas suformuluotas atsižvelgiant į kliento ir vartotojo organizacijos kalbą? • Ar kiekvienas reikalavimas nenaudoja techninio žargono?
Išorinių poreikių nustatymus (<i>External Observability</i>)	<p>Reikalavimai neturėtų pernelyg nurodyti vidaus architektūros ir dizaino komponentų. Taigi kiekvienas reikalavimas turėtų nurodyti elgseną arba charakteristikas, kurios išoriškai pastebimos:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas aprašo elgesį ar požymį išoriškai pastebimą? • Ar kiekvienas reikalavimas aprašomas, išvengiant vidaus architektūros, projektavimo arba testavimo sprendimų? • Jeigu reikalavimas nurodo vieną ar daugiau vidaus architektūros, projektavimo ar testavimo sprendimų, tai reikalavimas identifikuojamas kaip apribojimas, o ne kaip grynas reikalavimas?
Tinkamumo (<i>Feasibility</i>)	<p>Reikalavimai neturi prasmės, jei kūrėjai negali jų įgyvendinti. Taigi atskiriems reikalavimams turi būti įmanoma pateikti visus atitinkamus apribojimus:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas įgyvendinamas atsižvelgiant į esamą techninę ir programinę technologiją? • Ar kiekvienas reikalavimas įgyvendinamas atsižvelgiant į biudžetą? • Ar kiekvienas reikalavimas įgyvendinamas atsižvelgiant į tvarkaraštį? • Ar kiekvienas reikalavimas įgyvendinamas atsižvelgiant į darbuotojus (pvz., darbuotojų skaičiumi, žiniomis ir patirtimi)? • Ar kiekvienas reikalavimas įgyvendinamas atsižvelgiant į

Kokybės charakteristikos	Savybės
	fizinius, <ul style="list-style-type: none"> • cheminius apribojimus, ir t.t.?
Dviprasmiškumas <i>(Lack of Ambiguity)</i>	Kiekvienas reikalavimų komponentas turi būti nedviprasmiškas. <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas aiškus ir tikslus? • Ar kiekvienas reikalavimas glaustas (t.y. be nereikalingos ir neturinčios reikšmės informacijos)? • Ar kiekvienas reikalavimas turi tik vienos prasmės interpretaciją? • Ar kiekvienas reikalavimas suprantamas numatomai auditorijai? • Ar kiekvieno reikalavimo reikšmė suprantama taip pat, kas jį rašė ir kas jį skaitys? • Ar kiekvienas reikalavimas naudoja konkrečias reikšmes? • Ar kiekvienas reikalavimas vengia būdingų dviprasmiškų žodžių? • Ar kiekvienas reikalavimas apibūdinamas kiekybiniu būdu, jei tai tik įmanoma? • Ar kiekvienas reikalavimas apima visus būtinus teiginius: <ul style="list-style-type: none"> ✓ Invariantus? ✓ Prieš sąlygas? ✓ Po sąlygas? • Ar kiekvienas reikalavimas apima tik santrumpas ir techninius terminus, kurie vienareikšmiškai apibrėžti terminų žodyne ar reikalavimų specifikacijoje? • Ar kiekvienas reikalavimas apima tik aiškias nuorodas į kitus dokumentus? • Ar kiekviena diagrama susijusi su reikalavimu turi įrašą, kuris nurodo jo piktogramas?

Kokybės charakteristikos	Savybės
Būtinumas <i>(Mandatory)</i>	<p>Reikalavimai turi būti prioretizuoti, kiekvienas reikalavimas pagal savo pobūdį yra būtinas (pvz., reikalaujama):</p> <ul style="list-style-type: none"> • Ar kiekvieno reikalavimo komponentai būtini? • Ar kiekvienas reikalavimas tikrai būtinas (t.y. ar tikrai reikalavimas turi būti įgyvendintas)? • Ar kiekvienas reikalavimas tenkina užsakovo poreikius? • Ar kiekvienas reikalavimas neturi nereikalingų suvaržymų (pvz., architektūros, dizaino, įgyvendinimo, testavimo ir kitų technologinių sprendimų)? • Ar kiekvienas reikalavimas aprašo "kas", o ne "kaip"? • Ar kiekvienas reikalavimas išsiskiria iš: "Gerai turėti" punkto ir norėti? • Apribojimai?
Duomenys <i>(Metadata)</i>	<p>Kiekvienas reikalavimas turi turėti duomenis (pvz., atributus ar anotacijas), kurie charakterizuoja juos. Duomenys gali apimti (tačiau ne tik) pripažinimo kriterijus, paskirstymo, prielaidų, identifikavimo, prioritetų, loginio pagrindo, grafiko, būsenos, ir sekimo informacijas.</p> <ol style="list-style-type: none"> 1. Pripažinimo kriterijai: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas turi susijusius priėmimo kriterijus? • Ar kriterijai aiškūs ir objektyvūs? 2. Paskirstymo: <ul style="list-style-type: none"> • Ar kiekvienas sistemos reikalavimas skirtas PĮ architektūros elementui pilnai tinkamas? • Ar kiekvienas sistemos reikalavimas skirtas sistemos architektūriniam elementui pilnai tinkamas? 3. Prielaidos:

Kokybės charakteristikos	Savybės
	<ul style="list-style-type: none"> • Ar visos svarbios prielaidos susijusios su kiekvienu reikalavimu tinkamai dokumentuotos? 4. Identifikacija: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas turi savo unikalų identifikatorių, kuris gali būti naudojamas sekimo tikslais? • Ar kiekvienas reikalavimas nėra perteklinis su kitu reikalavimu tame pačiame abstrakcijos lygmenyje? 5. Prioritetų: <ul style="list-style-type: none"> • Ar reikalavimų prioritetai, pagrįsti taip: <ul style="list-style-type: none"> • Kritiniai reikalavimai klientui, organizacijai? • Planavimas architektūriniu požiūriu? • Projekto rizikos mažinimo? 6. Loginis: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas turi pagrindimą? 7. Tvarkaraštis: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas numatytas įgyvendinimui pagal konkrečias gaires? • Ar planas grindžiamas reikalavimų prioritetais? 8. Būsenos: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas turi susijusias būsenas (pvz., identifikavimo, analizavimo, patvirtinimo ir sustingimo)? • Ar būsenos atnaujinamos reikalavimų gyvavimo ciklo metu? 9. Sekamumas: <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas atsekamas jo šaltinio tikslais, dokumentu? • Ar kiekvienas reikalavimas apima pirminę ir atgalinę sekamumo informaciją?

Kokybės charakteristikos	Savybės
	<ul style="list-style-type: none"> • Ar kiekvienas sistemos reikalavimas įtraukia sekamumą sistemos tikslais? • Ar kiekvienas sistemos reikalavimas apima duomenų sekamumą ? • Ar kiekvienas programinės įrangos reikalavimas apima sekamumą savo sistemos komponentams ir sistemos reikalavimams? • Ar kiekvienas programinės įrangos reikalavimas apima duomenų, techninės įrangos, personalo ir programinės įrangos komponentus.
Aktualumas (Relevance)	<p>Kai identifikuoti ir nurodyti "reikalavimai" faktiškai atsiduria už taikymo srities. Taigi, svarbu užtikrinti, kad kiekvienas reikalavimas tinkamas:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas specifikuotas atitinkamai veiklos taikymo sričiai? • Ar kiekvienas komponentų reikalavimas išvengia nurodyto elgesio ir charakteristikų susijusių su vartotojais? • Ar kiekvienas komponentų reikalavimas išvengia nurodyto elgesio ir charakteristikų susijusių su sistemos išore (išskyrus privalomos sąsajos)?
Naudojimas (Usability)	<p>Reikalavimai turi daug vartotojų (pvz., vadovo, pirkėjo atstovų, prekybos atstovai, vartotojų atstovai, architektai, kūrėjai, testuotojai, pagalbinis personalas) ir juos naudoja įvairiais tikslais. Taigi, kiekvienas reikalavimas turėtų būti tinkamas naudoti skirtingų grupių vartotojams:</p> <ul style="list-style-type: none"> • Ar kiekvienas reikalavimas suprantamas ir naudojamas pagal užsakovo ir vartotojo įgaliojimus? • Ar kiekvienas reikalavimas suprantamas vadovui?

Kokybės charakteristikos	Savybės
	<ul style="list-style-type: none"> • Ar kiekvienas architektūrinis reikalavimas suprantamas architektui? • Ar kiekvienas reikalavimas, suprantamas projektuotojui ir programoms, kurios privalo juos įgyvendinti? • Ar kiekvienas reikalavimas naudojamas testuotojų, kuris tikrina ar reikalavimas tinkamas?
Validavimas (<i>Validatability</i>)	<p>Kiekvienas reikalavimas turi atitikti suinteresuotų šalių poreikius. Reikalavimai turi būti patvirtinti:</p> <ul style="list-style-type: none"> • Ar įmanoma užtikrinti, kad kiekvienas reikalavimas atitinka užsakovo norus ir poreikius? • Ar įmanoma užtikrinti, kad kiekvienas reikalavimas atitinka vartotojo norus ir poreikius? • Ar įmanoma užtikrinti, kad kiekvienas reikalavimas yra iš tikrųjų toks ko rinkodaros atstovai norėjo?
Verifikavimas (<i>Verifiability</i>)	<p>Reikalavimai turi šaltinius/ištakas ir yra svarbu, kad reikalavimai būtų suderinti su jais. Be to, reikalavimai turi būti suderinti su standartais, gairėmis, šablonais, kurie yra naudojami jų sudarymui. Taigi, kiekvienas reikalavimas turi būti patikrintas:</p> <ul style="list-style-type: none"> • Ar kiekvieną reikalavimą galima patikrinti nuo jo ištakų? • Ar reikalavimą galima patikrinti pagal atitinkamus standartus, gaires ir šablonus?

2.6. Kokybiškos reikalavimų specifikacijos savybės ir vertinimo kriterijai

Visų pirma reikalavimų specifikacija yra gerai parašyta, jeigu ji tenkina projektuotojų poreikius. Tai reiškia, kad dokumento savybės lemia tikslai, kuriems jis yra naudojamas. Vieni savybių reikia tam, kad dokumentą būtų patogu skaityti, kitų tam, kad būtų nesunku patikrinti, ar projektuotojai tikrai įgyvendino visus reikalavimus, dar kitų tam, kad dokumentą būtų nesunku keisti. Nėra kokio nors visuotinai pripažinto reikalavimų specifikacijos formato standarto, tačiau yra įprasta numeruoti visus reikalavimus [31]:

Tinkamumas (*appropriateness*): reikalavimų specifikacija yra konceptuali, jei visi joje pateikti reikalavimai yra abstraktūs, t.y. joje nėra liečiami sistemos projektavimo ar įgyvendinimo klausimai. Nekonceptuali specifikacija per daug suvaržytų projektuotojus ir programuotojus, trukdytų jiems parinkti geriausią sistemos įgyvendinimo būdą. Kita vertus, dažnai yra labai sunku išvengti kai kurių projektavimo ar realizavimo ribojimų [31].

Koncepcinis skaidrumas: apima specifikacijos paprastumą, aiškumą ir suprantamumą. Ši savybė siejasi su specifikacijos konceptualumu. Jei reikalavimai saugomi duomenų bazėje ir tvarkomi programiniu būdu, koncepcinis skaidrumas dažniausiai yra paaukojamas, siekiant padidinti reikalavimų apdorojimo našumą [31].

Konkretumas: specifikacija yra konkreti, jei įmanoma patikrinti visų joje suformuluotų reikalavimų įgyvendinimo laipsnį. Tai svarbi specifikacijos savybė, nes šis dokumentas yra užsakovo ir vykdytojo sandorio dalis ir jos pagrindu yra sprendžiama ar sandoris buvo įvykdytas [31].

Geras struktūrizavimas: joje griežtai išlaikytas turinių atskyrimo principas.

Tikslumas: visi joje suformuluoti reikalavimai yra tikslūs.

Išsamumas: specifikacijoje aprašyti visi reikalingi sistemos funkcionalumai ir visi pateikti reikalavimai yra išsamūs. Tai reiškia, kad specifikacijoje yra viskas, kas joje turėtų būti. Sunku patikrinti ar specifikacija tikrai yra išsami. Nagrinėjant tai, kas yra aprašyta, labai sunku pastebėti, ko gi ten trūksta. Geriausias būdas išsamumui patikrinti yra sukurti sistemos prototipą [31].

Išsamioje specifikacijoje turi būti aprašyta, kaip sistema reaguoja į kiekvieną iš galimų įvesčių tipų, kiekvienoje galimoje situacijoje. Reikia nagrinėti ne tik kaip reaguoja sistema į leistinas (teisingas) įvestis, bet ir kaip ji reaguoja į klaidingas įvestis [31].

Išsamioje specifikacijoje turi būti visos reikalaujamos to dokumento dalys, visi puslapiai, visi paveikslėliai ir visos lentelės turi būti sunumeruoti, paveikslėliai ir lentelės turi turėti pavadinimus, turi būti pateiktos tvarkingos nuorodos į visus naudojamus išorinius informacijos šaltinius. Tai formalus dokumento išsamumas, kurį galima vadinti ir dokumento tvarkingumu. Kokias dalis turi turėti dokumentas, nustato vidiniai organizacijos (arba projekto) standartai. Kartais analizės metu visų reikalavimų specifikuoti nepavyksta, kai kurių reikalavimų specifikavimą tenka atidėti vėlesniam laikui. Tokie reikalavimai pažymimi žyme. Išsamioje specifikacijoje kiekvienam žyme pažymėtam reikalavimui turi būti nurodyta [31]:

- Kodėl reikalavimo formulavimas yra atidėtas;

- Kas turi būti atlikta, kad reikalavimą būtų galima suformuluoti;
- Iki kada reikalavimas turi būti suformuluotas;
- Kas atsakingas už tai, kad reikalavimas būtų suformuluotas nurodytu laiku.

Vienareikšmiškumas: joje neturi būti jokių dviprasmybių. Tai labai svarbi specifikacijos savybė, nes šis dokumentas yra užsakovo ir vykdytojo sandorio dalis ir jos pagrindu yra sprendžiama ar sandoris buvo įvykdytas [31].

Trasuojamumas: reikalavimai yra lokalizuojami, specifikacijas galima tarpusavyje susieti kryžminėmis nuorodomis [31].

Darnumas: visi suformuluoti reikalavimai yra integruojami, jokių prieštaravimų dokumente nėra. Nedarna gali atsirasti dėl įvairių priežasčių, pavyzdžiui, dėl [31]:

- Terminų konflikto: skirtingose dokumento vietose tas pats daiktas yra vadinamas skirtingai;
- Savybių konflikto: skirtingose reikalavimuose yra reikalaujama, kad sistema turėtų kokias nors nesuderinamas savybes;

Keičiamumas: dokumentas turi būti lengvai keičiamas. Specifikaciją lengva keisti, jei ji parašyta griežtai prisilaikant turinio atskyrimo principo ir visi reikalavimai turi unikalius numerius.

Naudojimo patogumas: mažai kas skaito visą dokumentą, kiekvienam reikia tik to, kas jam svarbu. Todėl dokumentas turi būti parašytas taip, kad juo būtų galima naudotis kaip žinynu [31].

2.7. Reikalavimų klasifikacija remiantis ISO 9126 ir FURPS+ standartais

Specifikacijos plėtojimas ir standartizavimas yra sudėtingas procesas. Rašant reikalavimų specifikacijas galima remtis įvairiais standartais. Vienas iš pagrindinių projekto tikslų yra funkcinių ir nefunkcinių reikalavimų surinkimas. Apžvelgiami ISO 9126 ir FURPS+ standartai.

2.7.1. ISO 9126 - tarptautinis PĮ kokybės vertinimo standartas

ISO 9126 yra programinės įrangos kokybės tarptautinis standartas, kuris buvo sukurtas 1991 metais ir per dešimtmetį dar patobulintas, tam kad nustatytų karkasą programinės įrangos kokybės įvertinimui [42].

Šio standarto tikslas – apibrėžti programinės įrangos kokybės vertinimo šabloną. ISO/9126 standartas nekelia reikalavimų programinei įrangai, tačiau pateikia kokybės modelį, kuris gali būti taikomas bet kokiai programinei įrangai [42].

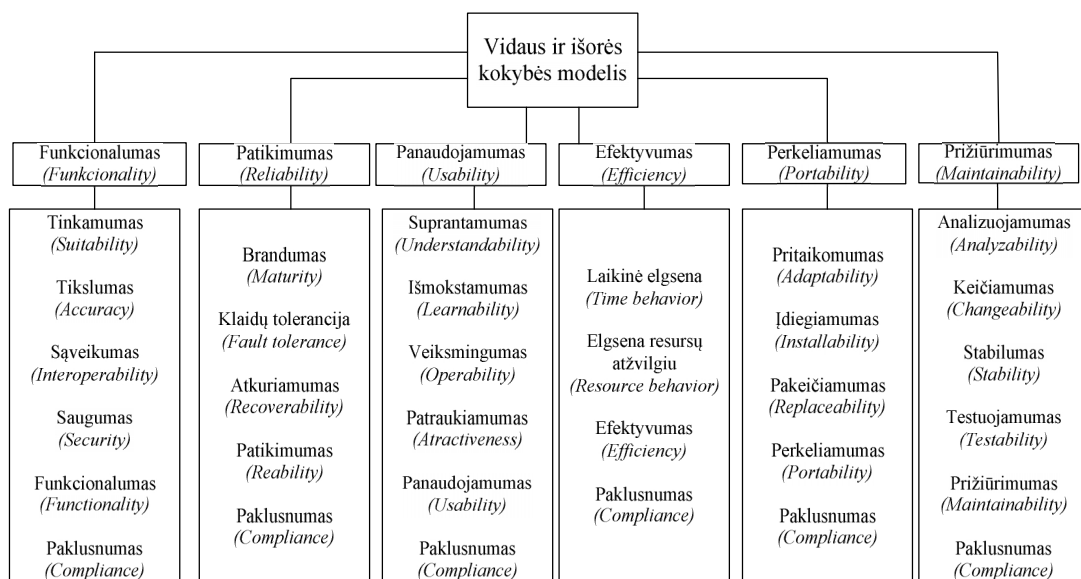
Standartas sudarytas iš 4 dalių:

1. Kokybės modelio;
2. Išorinės metrikos - taikomos vykdant PĮ veiksmus;
3. Vidinės metrikos - tos, kurios nepriklauso nuo PĮ vykdymo (statinės priemonės);
4. Kokybės naudojimo metrikos - galimos tik tada, kada galutinis produktas yra naudojamas realiomis sąlygomis.

Kokybės modelis (*Quality model*) yra pirmoji ISO 9126 standarto dalis, kurioje yra apibrėžtas struktūrizuotas PĮ kokybės charakteristikų ir sudėtinių charakteristikų rinkinys [42].

Vidinis ir išorinis kokybės modeliavimas. Modelis grindžiamas šešiomis charakteristikomis: funkcionalumas, patikimumas, naudingumas, veiksmingumas, prižiūrimumas, ir perkeliamumas. Kaip parodyta 2 paveiksle, kiekviena iš šių savybių turi savo sudėtines charakteristikas [42].

Kokybės modelio naudojimas yra pagrįstas savybėmis: veiksmingumo, našumo, saugumo ir patenkinimo. Faktinis programinės įrangos produkto įvertinimas nėra ISO / IEC 9126 standarto dalis, bet jis apibrėžtas ISO / IEC 14.598 [ISO01]: Kad būtų galima atsižvelgti į skirtingų produktų skirtingus reikalavimus, modelis turi būti plėtojamas, įvertinant skirtingas sudėtines charakteristikas ir parenkant tinkamas metrikas [42].



2. pav. ISO/IEC 9126 Vidaus ir išorės kokybės modelis [42]

Standartas suteikia pagrindą organizacijoms, kaip apibrėžti kokybės modelį PĮ produktui. Tačiau šis standartas kiekvienai organizacijai palieka užduotį tiksliai nurodyti savo kokybės modelį.

2.7.2. FURPS+ modelis

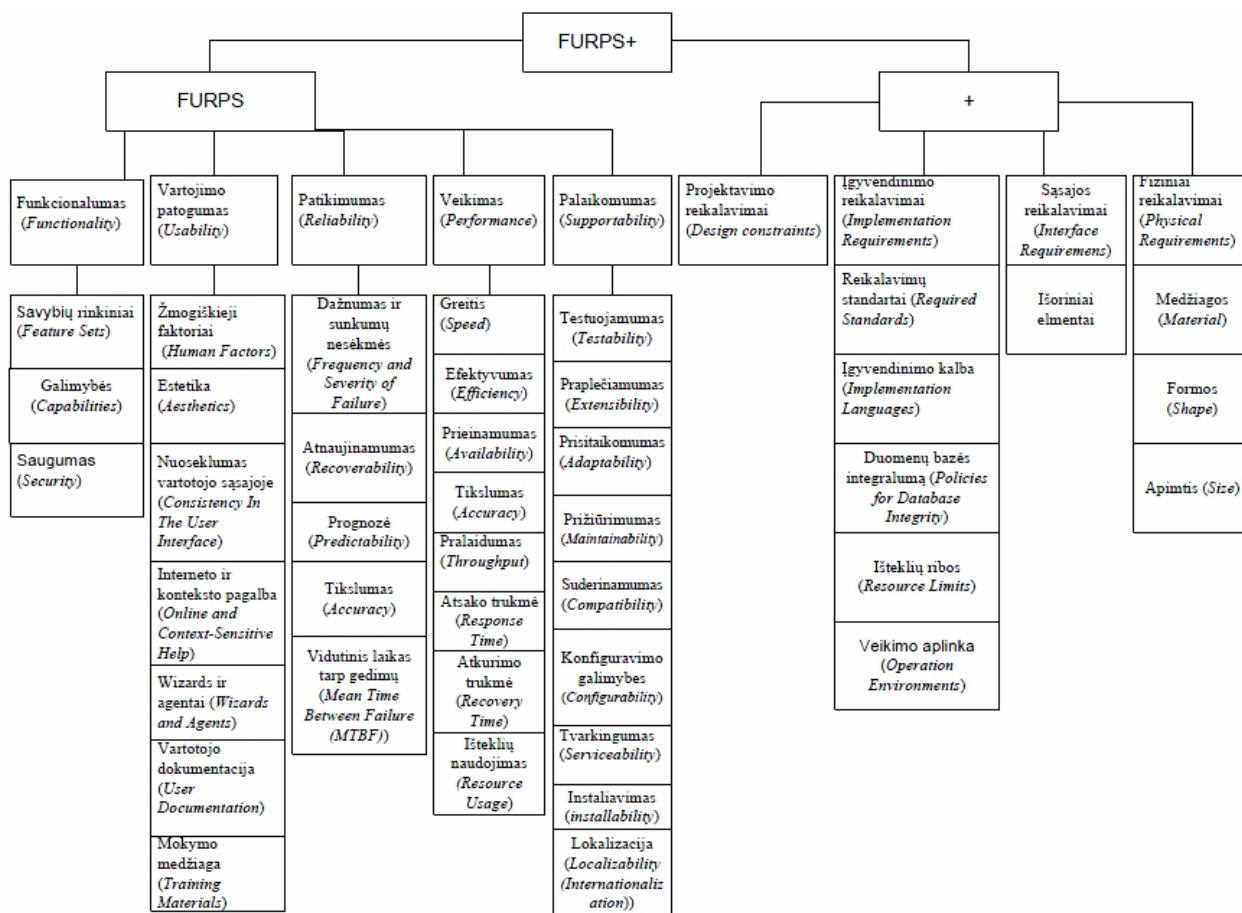
FURPS+ yra reikalavimų modelis [11], apibrėžiantis vartotojo reikalavimus sistemai. Reikalavimai skirstomi į dvi kategorijas: funkcinis ir nefunkcinis. Vienas iš būdų suskirstyti juos į kategorijas yra FURPS + modelis. Santrumpa FURPS skirta aprašyti reikalavimų kategorijas ir sudėtines kategorijas [11].

- Funkcionalumas (*Functionality*);
- Vartojimo patogumas (*Usability*);
- Patikimumas (*Reliability*);
- Veikimas (*Performance*);
- Palaikomumas (*Supportability*);

“+” „pius“ FURPS+ modelyje parodo, kad jis dar turi papildomus reikalavimus:

- Projektavimo reikalavimai (*Design constraints*);
- Įgyvendinimo reikalavimai (*Implementation Requirements*);
- Sąsajos reikalavimai (*Interface Requirements*);
- Fiziniai reikalavimai (*Physical Requirements*);

FURPS+ modelis pateiktas 3 pav.



3. pav. FURPS+ modelio reikalavimų klasifikavimo schema [11]

2.8. Reikalavimų specifikavimo šablonų analizė

Šiuo metu yra daug įvairių metodų ir priemonių, kurios skirtos kompiuterizuotoms informacinėms sistemoms kurti. Reikalavimams specifikuoti naudojami šablonai, kurie turi savo struktūrą ir grupavimą. Tokių šablonų naudojimas reikalavimų specifikavime suteikia galimybę tiksliau išsiaiškinti užsakovo pateiktus reikalavimus ir juos struktūrizuoti.

2.8.1. SRS šablono analizė

SRS (*Software Requirements Specification*) yra samprata apie kliento reikalavimus. Tai abipusė apsidraudimo politika, garantuojanti kliento ir organizacijos tarpusavio reikalavimų suvokimą [25].

SRS aprašomas sistemos elgesys. Specifikuojami panaudojimo atvejai, aprašant vartotojų sąveikas su sistema. SRS visi panaudojimo atvejai įvardijami funkciniais reikalavimais. SRS šablone aprašomi ir nefunkciniai reikalavimai [25].

SRS šablono naudojimas:

- Nustato susitarimą tarp kliento ir tiekėjo;
- Pateikia apytikrį išlaidų planą ir tvarkaraštį;
- Padeda vartotojams nustatyti, ar PĮ atitinka reikalavimus;
- Padeda sumažina plėtros pastangas.

Šablonas apima sekančias reikalavimų kategorijas [10]:

- Funkciniai reikalavimai (Functional Requirements);
- Patikimumo ir palaikomumo reikalavimai (Reliability & Maintainability Requirements);
- Veikimo reikalavimai (Operational Requirements);
- Sąsajos reikalavimai (Interface Requirements);
- Tikrinimo reikalavimai (Verification Requirements);
- Dokumentacijos reikalavimai (Documentation Requirements);
- Kokybės reikalavimai (Quality Requirements);
- Saugumo reikalavimai (Safety Requirements);
- Našumo reikalavimai (Performance Requirements);
- Išteklių reikalavimai (Resource Requirements);

SRS šablono pagrindu sudaryta reikalavimų specifikacija pasižymi šiomis charakteristikomis (žr. lentelė 2) [25]:

2. lentelė. SRS šablono charakteristikos [25]

SRS Kokybės charakteristikos	Reikšmė
Pilnumas	SRS tiksliai apibrėžia visas situacijas su kuriomis bus susidurta, sistemos gebėjimas sėkmingai jas spręsti. Pagrindinių reikalavimų apibrėžimai siejasi su funkcionalumu, našumu, projektavimo ribomis, išorinėmis sąsajomis;
Nuoseklumas	SRS funkcijų vykdymo lygiai ir reikiamos kokybės savybės yra suderintos (saugumas, patikimumas ir pan.) Nei vienas reikalavimas neturi prieštaravimų su kitais reikalavimais;
Tikslumas	SRS tiksliai apibrėžia sistemos galimybes;
Modifikuojamumas	Logiška hierarchinė SRS struktūra palengvina būtinus pakeitimus (grupuojant susijusius klausymus bei atskiriant juos nuo nesusijusių klausymų);
Testuojamumas	SRS turi būti išdėstyta pagal vienareikšmiško vertinimo kriterijus;
Sekamumas	Kiekvienas reikalavimas turi būti identifikuotas; Reikalavimų ištakos bei nuorodos yra pasiekiamos; Atliktų modifikacijų sekimas iki dabartinės būsenos;
Nedviprasmiškas	Kiekvienas reikalavimas turi vieną ir tik vieną interpretaciją; Formalios reikalavimų specifikavimo kalbos vartojimas;
Efektyvus	Projekto dalyviai gali suprasti, analizuoti, priimti arba patvirtinti sprendimus;

2.8.2. URS šablonas

URS (*User Requirements Specification*) aiškiai apibrėžia, vartotojo norus ką sistema darys. Dokumentas skirtas užrašyti vartotojų poreikiams. URS yra tiltas tarp verslo ir sistemos reikalavimų. Dokumentas sudaromas dėl bendro sutarimo su vartotoju, ką sistema turi atlikti [37].

Vartotojų reikalavimų dokumentas [37]:

- Formalizuoja reikalavimus iš visų vartotojų;
- Atvaizduoja reikalavimus;

Pateikia bendrą informaciją apie sistemos aplinką, funkcinius poreikius ir kokybės atributus.

URS parengimas susideda iš dviejų žingsnių [37]:

1. Šaltinių identifikavimo;
2. Reikalavimų organizavimo.

Dokumento šaltiniai - produkto vizija, vartotojų procedūriniai dokumentai (gidai, procedūrų aprašymas, mokymosi medžiaga), kita esamos sistemos dokumentacija ir kt. [37].

Kiekvienas reikalavimas turi unikalų identifikatorių, siekiant išvengti reikalavimų pasikartojimų šablone. Reikalavimų pilnumas URS šablone užtikrinamas sutelkiant dėmesį ne į sistemos funkcijas, o į visus galimus scenarijus ir vartotojo tikslus. Reikalavimų nuoseklumas URS šablone užtikrinamas, sprendžiant visus nesuderinamumus tarp reikalavimų, tai išvengiama specifikuojant reikalavimus prisilaikant struktūros. Kiekvienas reikalavimo pakeitimas turi būti dokumentuotas, tai reiškia kad kiekvieno reikalavimo istorija turi būti fiksuojama. Taip pat URS šablone reikalavimas turi būti aprašomas tik vieną kartą. Šablone aprašytus reikalavimus galima atsekti, nes kiekvienas reikalavimas turi unikalų identifikatorių ir šaltinį.

Reikalavimai išdėstyti URS testuojami našumu. Vartotojų reikalavimų specifikacija nėra skirta techniniam dokumentavimui, skaitytojas bendromis žiniomis apie sistemą turi sugebėti suprasti reikalavimus, išdėstytus URS [37].

2.8.3. Volere šablono analizė

Volere šablonas – tai daugelio metų reikalavimų inžinerijos tyrinėjimų ir praktikos rezultatas. Šio šablono pagrindu atliekama reikalavimų analizė, jie suskirstomi į tipus bei sudaromas reikalavimų specifikacijos dokumentas [18,42]. S. Robertson ir J. Robertson [33] rekomenduoja kiekvieną reikalavimą aprašyti vienu sakiniu, išvengiant sudėtinių sakinių, nes tokie sakiniai gali turėti daugiau nei vieną reikalavimą.

Volere šablonas naudojamas vartotojų reikalavimams užregistruoti pradiniame sistemos kūrimo etape. Šablone reikalavimų tipai suskirstyti skyriais. Šablone kaupiami pradiniai reikalavimai, kuriuos pateikia vartotojai. Šablonas gali būti modifikuojamas, pritaikant jį išskirtiniam ar konkrečiam atvejui ar dalykinei sričiai. Šablonas modifikuojamas įterpiant naujus

skyrius ar juos išmenant, taip specifikuojant specifines dalykinės srities dalis. Volere šablone apimami šie reikalavimų tipai [32]:

- Funkciniai reikalavimai (*Functional requirements*);
- Nefunkciniai reikalavimai (*Nonfunctional requirements*);
- Projekto apribojimai (*Project constraints*);
- Projekto varovai (*Project drivers*);
- Projekto išeiga (*Project issues*);
- Testavimo reikalavimai (*Testing requirements*).

Reikalavimas turi būti stebimas per visą sistemos kūrimo laiką, todėl logiška, kad numeravimas yra unikalus. Įvykių numeriai pritaikomi panaudojimo atvejams numeruoti. Reikalavimai siejami su panaudos atvejais, nurodant jų numerius.

- Funkciniai reikalavimai ir reikalavimai duomenims. Specialioje formoje aprašomi visi funkciniai reikalavimai (5, 6 pav.)

Reikalavimas #:	Reikalavimo tipas:	Įvykis/panaudojimo atvejis #:
Aprašymas:		
Pagrindinias:		
Šaltinis:		
Tinkamumo kriterijus:		
Užsakovo patenkinimas:	Užsakovo nepatenkinimas:	
Priklausomybės:	Konfliktai:	
Papildoma medžiaga:		
Istorija:		

4. pav. Volere šablonas funkciniam reikalavimams aprašyti [38]

Reikalavimas #:	45	Reikalavimo tipas:	I0	Ivykis/panaudojimo atvejis #:	6
Aprašymas:	<i>Sistema turi signalizuoti, jei oro stotis neperduoda duomenų</i>				
Pagrindimas:	<i>Trikdžiai perduodant duomenis gali nurodyti, kad oro duomenų registravimo stotis sugedusi ir plikšalai prognozuoti naudojami duomenys yra nepilni</i>				
Šaltinis:	<i>Kelių priežiūros inžinierius</i>				
Tikimo kriterijus:	<i>Sistema informuoja vartotoją, jei kurios nors oro stoties perduotų kiekvieno tipo parametrų kiekis per valandą neatitinka numatyto parametrų registravimo dažnumo</i>				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės:	<i>Nėra</i>		Konfliktai:	<i>Nėra</i>	
Papildoma medžiaga:	<i>Philips oro parametrų matavimo stoties specifikacija</i>				
Istorija:	<i>Užregistruotas 2001 spalio 15 d.</i>				

5. pav. Užpildytos reikalavimų formos pavyzdys [38]

Volere šablonas naudojamas reikalavimams aprašyti, vadovaujantis proceso, nusakančiais, kokio etapo metu reikalavimai turi būti aprašyti.

Visi reikalavimų tipai yra suskirstyti į potipius, taip užtikrinamas reikalavimų konkretumas ir nepersidengiamumas. Volere reikalavimų specifikavimo šablone reikalavimai siejami su panaudojimo atvejais ar įvykiais. Panaudojimo atvejis parodo tai, ką produktas atliks, todėl jis turi daug susijusių funkcinių reikalavimų. Kiekvienas sistemos panaudojimo atvejis gilinamas į išsamias funkcinės savybes, išreiškiant panaudojimo atvejus su tekstiniu aprašu. Taigi, kiekvienas FR yra susietas su vienu ar daugiau panaudojimo atvejų. Be to, panaudojimo atvejis taip pat gali turėti ir nefunkcinių reikalavimų bei apribojimų.

Volere šablonas yra aiškiai apibrėžtas ir turi savo struktūrą. Šablonas lankstus ir pritaikytas dirbti su dauguma reikalavimų valdymo įrankių (pvz., *Caliber*, *DOORS*, *Requisite*, *IRqA* ir t.t.). Volere taiko modeliavimo technikas, pavyzdžiui, UML, veiklos procesų, duomenų ir būsenų modelius [32].

2.8.4. FSD reikalavimų specifikavimo šablonas

Reikalavimo specifikavimo šablonas (FSD- Functional Specifications Documents) naudojamas programinės įrangos funkcionalumui aprašyti [15].

Funkciniai reikalavimai apibrėžia programinės įrangos funkcijas ir kaip sistema turi elgtis kai pateikiami sistemai duomenys ar sąlygos. Tai gali būti skaičiavimai, duomenų manipuliavimas ir kitos funkcijos. Funkciniai reikalavimai turi unikalų pavadinimą, numerį,

santrauką ir logiškumą. Šablone specifikuojami sistemos tikslai, funkciniai ir nefunkciniai reikalavimai.

Šablonas naudojamas [3]:

- Sistemos elgesio specifیکavimui;
- Panaudos atvejams išgauti. Kiekvienas panaudos atvejis aprašo vieną ar kelis funkcinis reikalavimus;
- Padėti skaitytojui suprasti kam reikalingas reikalavimas;

FSD šablono pagrindu sudaryta reikalavimų specifیکacija pasižymi šiomis charakteristikomis:

- Padeda išvengti reikalavimų dubliavimosi ir prieštaravimų;
- Padeda apskaičiuoti reikiamus išteklius;
- Nustatyti nuoseklius ryšius visoje programinės įrangos gyvavimo cikle;
- Galimybė techniškai valdyti dokumentą [2].

Reikalavimų šablonas apima 27 puslapius MS Word™ formatu, kurį galima lengvai modifikuoti pagal projekto poreikius. Kiekvienas reikalavimų tipas turi savo skyrių. Visi reikalavimų tipai yra suskaidyti į smulkesnius. Šablone apimami šie reikalavimų tipai:

Funkciniai reikalavimai;

Sistemos reikalavimai;

Įvesties ir išvesties įvykių reikalavimai;

Nefunkciniai reikalavimai;

Vartotojo sąsajos reikalavimai;

Projekto apribojimo reikalavimai;

Aplinkos reikalavimai;

Duomenų reikalavimai;

Projektavimo reikalavimai;

Funkcinės specifیکacijos tikslas yra aiškiai apibrėžti ką sistema darys vartotojui orientuota ne technine kalba. Ji turėtų aprašyti, kokie duomenys bus įvesti į sistemą, kokie išvesties rezultatai, kokios verslo taisyklės taikomos įvykiams.

2.8.5. Reikalavimų specifیکavimo šablonų palyginimas

Apžvelgtų reikalavimų specifیکavimo šablonų Volere, SRS, FSD, URS palyginimas atliktas, atrinkus kriterijus pagal nagrinėjamų šablonų charakteristikas.

Išanalizuotų reikalavimų specifikavimo šablonų palyginimas pateiktas 3 lentelėje.

3. lentelė. Šablonų palyginimas

Savybės	Šablonai			
	SRS	URS	FRD	Volere
Reikalavimų klasifikavimas skyriais	Taip	Taip	Taip	Taip
Reikalavimų unikalus identifikavimas	Taip	Taip	Taip	Taip
Reikalavimų susietumas	Taip	Ne	Taip	Taip
Reikalavimų sekamumas	Taip	Taip	Ne	Taip
Panaudojimo atvejai	Taip	Taip	Ne	Taip
Panaudos atvejų sąsajos	Taip	Taip	Ne	Taip
Konfliktų fiksavimas	Ne	Ne	Ne	Taip
Šablono modifikavimas	Taip	Taip	Taip	Taip
Reikalavimų detalumas	Taip/Ne	Taip/Ne	Taip/Ne	Taip/Ne
Modifikavimas	Taip	Taip	Taip	Taip
Nemokamas	Taip	Taip	Taip	Taip

Iš pateiktos 3 lentelės duomenų galima teigti, kad geriausiai visas reikalavimų specifikavimo charakteristikas išpildo Volere šablonas. Dėl tokių kriterijų, kaip: reikalavimų tipų pasirinkimas, detalumas, sekamumas ir kitų.

2.9. Reikalavimų specifikavimo įrankių analizė

Reikalavimų valdymo įrankiai tai priemonės naudojamos reikalavimams kurti, gauti, atnaujinti ar pašalinti. Tai pagrindinės funkcijos reikalavimams valdyti. Kad reikalavimų valdymas būtų efektyvesnis reikalingos papildomos funkcijos, pavyzdžiui, kaip galimybė importuoti ir eksportuoti reikalavimus, integruoti su įrankiais, apimant artefaktus kitose programinės įrangos kūrimo etapuose, ryšių specifikavimas tarp reikalavimų, kuris vadinamas atsekamumu.

2.9.1. RequisitePro įrankis

IBM Rational RequisitePro yra dalis IBM Rationale Suite reikalavimų valdymo įrankis. Šis įrankis naudojamas reikalavimams modeliuoti ir jų saugojimui reliacinėje duomenų bazėje. Reikalavimų pakeitimai saugomi versijų istorijoje, todėl yra aišku kas buvo pakeista ir kada.

Reikalavimai ir artefaktai gali būti susieti vienas su kitu naudojant sąsajas. Šios sąsajos padeda atlikti poveikio analizę siekiant nustatyti kaip vieno artefakto pakeitimas paveikė kitą.

IBM Rational RequisitePro [18] programinė įranga leidžia kurti reikalavimus, naudojant panaudos atvejus. RequisitePro įrankis pagerina komandinį darbą ir sumažina projekto riziką.

Įrankis Rational RequisitePro [16,17,18], skirtas sėkmingai valdyti pastovius ar nuolat besikeičiančius reikalavimus, pagerinti sekamumą. Įrankis palengvina netekstinių reikalavimų elementų (pvz. modelių, diagramų, grafikų) įtraukimą į reikalavimų dokumentą. Naudojant užklausas ir filtrus nesunku išgauti reikalingą informaciją, kurios reikia vartotojui. Įrankis turi internetinę prieigą, taip komandinis darbas tampa efektyvesnis, ir vartotojas gali greitai ir efektyviai peržiūrėti ir valdyti reikalavimus, net tuo atveju, jei įrankis nėra įdiegtas.

Šis įrankis taip pat integruojamas su Microsoft Word, taigi reikalavimai ir kiti artefaktai gali būti specifikuojami Microsoft Word ir RequisitePro, bei galima juos valdyti ir atnaujinti jų informacija iš šių įrankių. Be to, Microsoft Word taip pat gali būti integruojamas ir su kitais įrankiais Rational Suite.

Įrankis puikiai suderinamas su Microsoft SQL Server, Oracle, ir Microsoft Access duomenų bazėmis. Be to, reikalavimų informacija pateikiama tokia forma, kuri prieinama ir modifikuojama kitu IBM produktų pagalba (pvz. Rational Software Architect, Rational Software Modeler, Rational Systems Developer, Rational SoDA®, Rational Unified Process and Rational TestManager®, ir daug kitų).

Kadangi RequisitePro yra reikalavimų valdymo įrankis, visi artefaktų tipai ir artefaktai kurie yra sukurti vadinami reikalavimų tipais ir reikalavimais RequisitePro įrankyje [16, 17, 18].

2.9.2. DOORS įrankis

Antrasis analizuojamas įrankis yra DOORS (*Dynamic Object Oriented Requirements System*), tai informacijos valdymo ir atsekamumo (IMT) įrankis [2]. IBM Rational Doors programinė įranga yra reikalavimų valdymo įrankis, kuris padeda sumažinti išlaidas, padidinti efektyvumą bei pagerinti kokybę per optimizuotus reikalavimų ryšius. DOORS teikia išsamią reikalavimų valdymo aplinką. Visos suinteresuotos šalys gali bendradarbiauti reikalavimų procese naudojant internetinę naršyklę reikalavimų duomenų bazėje. Serveris palaiko duomenų bazines su visais projektais, reikalavimais, ryšiais ir t.t., vartotojas gali prisijungti prie duomenų bazės naudojant kliento programas. Remiantis vartotojų teisėmis, jie gali peržiūrėti ar keisti informaciją duomenų bazėje.

DOORS naudoja duomenų bazę saugoti projektus, kaip ir kiti įrankiai. Šioje duomenų bazėje gali būti saugomi keli projektai ir kiekvienas projektas gali turėti savo informacijos modelį. Modelis palaiko kelis modelius su jų ryšiais ir objektais atstovaujančius tam tikrus reikalavimus. Naudojant įrankį DOORS, reikalavimai tvarkomi kaip atskiri objektai. Kiekvienas reikalavimas gali turėti neribotą atributų skaičių [2].

DOORS palaiko du semantinius ryšio tipus: vidinis ir išorinis. Vidinis ryšys gali būti sukuriamas tarp dviejų objektų toje pačioje duomenų bazėje, o išorinis ryšys gali būti naudojamas sujungti išorinius artefaktus.

Verifikavimo matricos sudaromos tiesiogiai arba išvedant įrankio palaikomais formatais, įskaitant *RTF*, *MS-Word* [2]. DOORS taip pat palaiko failų kūrimą naudojant Framemaker priemones siekiant techninius dokumentus. Palaiko atskirus failus CSV ir TSV, kurie naudojami mainams tarp reikalavimų ir įrankio.

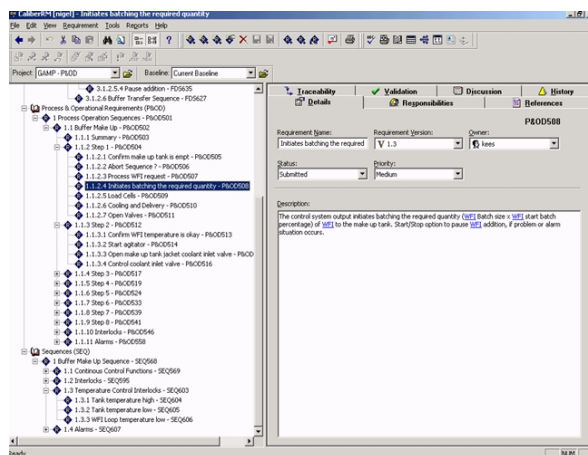
2.9.3. CaliberRM įrankis

CaliberRM yra reikalavimų valdymo įrankis, kuris užtikrina galutinio vartotojo poreikius [6]. CaliberRM įrankis suteikia galimybę analitikams, programuotojams, testuotojams ir kitoms suinteresuotosioms šalims tiksliai fiksuoti ir perduoti reikalavimus viso gyvavimo ciklo eigoje.

CaliberRM padeda kurti aukštesnės kokybės reikalavimus ir skatina bendradarbiavimą tarp suinteresuotųjų šalių, įsitikinant, kad svarbiausi aspektai atliekami pirmiausiai. Pokyčių valdymo galimybės užtikrina, kad pakeitimai būtų vykdomi ir valdomi per visą gyvavimo ciklą [6].

CaliberRM savybės [4, 6]:

- Reikalavimų sekamumas;
- Turi sąsają su: Web naršykle, Eclipse, Microsoft Visual Studio;
- Galimybė dirbti keliems asmenims su tuo pačiu projektu;
- Ataskaitų pateikimas apie pakeitimus.
- Pranešimai – apie pakeistus reikalavimus,
- Galimybė valdyti reikalavimų dokumentų versijas.
- Centralizuotoje ir saugioje saugykloje saugomi visi projekto reikalavimai, lengvai konfigūruojami projekto reikalavimų poreikiai.
- Galimybė palaikyti tūkstančius reikalavimų, kurie valdomi kaip atskiri objektai.
- Tvarko reikalavimus ir su jais susijusius duomenis hierarchijomis.



6. pav. CaliberRM įrankio reikalavimų valdymo langas

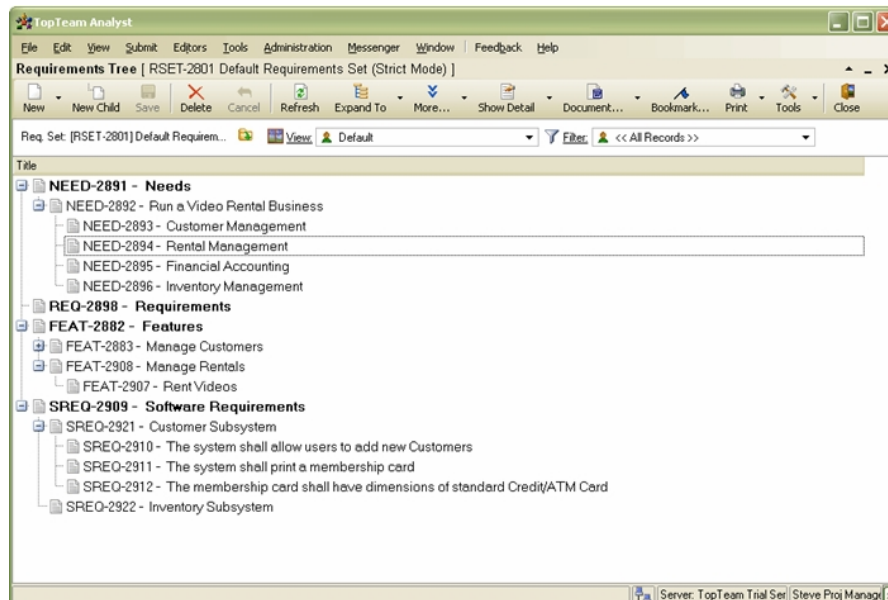
2.9.4. TopTeam™ Analyst įrankis

TopTeam™ Analyst - tai įrankis, skirtas reikalavimams surinkti ir valdyti, padedantis programinės įrangos kūrėjams pasiekti projekto tikslus [36].

TopTeam™ įrankio savybės:

- Hierarchinė reikalavimų struktūra, kurią galima peržiūrėti ir redaguoti įtrauktime WYSIWYG (Volere) dokumento redaktoriuje;
- Palaikomi įvairūs reikalavimų tipai – funkciniai ir nefunkciniai reikalavimai, verslo taisyklės, sąsajų kūrimas tarp kitų reikalavimų ir sistemos elementų;
- Panaudos atvejų diagramų įrankiai;
- Panaudos atvejų srautai ir įvykių žingsnių seka, gali būti automatiškai atvaizduojami sekų diagramoje;
- Reikalavimų sekamumui naudojamos keturios sąsajos, kurios skirtos ryšiams kurti ir atvaizduoti;
- Ataskaitų generavimas;
- Galingas teksto redaktorius – leidžiantis geriau išreikšti reikalavimus, lentelėmis ir paveikslėliais;
- Pakeitimų istorijos saugojimas – registruojama kiekvieno reikalavimo pakeitimo informacija. Kodėl jis buvo keistas, kada ir kas jį keitė;
- Diskusijų galimybė - vartotojų bendravimas naudojant diskusijas (naudojant elektroninį paštą);
- Pranešimai – automatiniai pranešimai apie pakeistus reikalavimus.

TopTeam™ Analyst - reikalavimų valdymo įrankis suteikia visas priemones, kurių reikia dokumentų, prioritetų nustatymui ir reikalavimų valdymui, pagal vartotojų poreikius [36].



7. pav. Reikalavimų hierarchijos langas TopTeam™ Analyst įrankyje

TopTeam™ Analyst palaiko ne tik tekstinius reikalavimus, jis taip pat palaiko panaudos atvejus reikalavimų specifikavimo procese. Kadangi TopTeam™ Analyst turi papildomus artefaktus jis yra mažiau lankstus negu kiti reikalavimų valdymo įrankiai.

2.9.5. Reikalavimų specifikavimo įrankių palyginimas

4 lentelėje pateikta apžvelgtų įrankių savybių palyginimas. Reikalavimų valdymo įrankiai lyginami tarpusavyje atsižvelgiant į kuriamam metodui svarbius kriterijus: sekamumą, šablonų bibliotekas, papildomo funkcionalumo suteikimą, integraciją su kitais produktais ir kiti. Įrankių palyginimo rezultatai pateikti 2.3 lentelėje.

Atlikus specifikavimo įrankių analizę, pastebėta, kad visi įrankiai turi bendrų savybių, trūkumų bei privalumų.

4. lentelė. Reikalavimų specifikavimo įrankių palyginimas

Savybė	RequisitePro	DOORS	TopTeam™ Analyst	CaliberRM
Keletas vartotojų gali dirbti	Taip	Taip	Taip	Taip

Savybė	RequisitePro	DOORS	TopTeam™ Analyst	CaliberRM
prie to paties projekto				
Grafinių elementų įtraukimas į DB	Ne	Taip	Taip	Ne
Šablonų bibliotekų gausa	Taip	Taip	Taip	Ne
Sekamumas	Taip	Taip	Taip	Taip
Ms Word dokumento ataskaitos	Taip	Taip	Taip	Taip
Integracija su MS Project paketu	Taip	Taip	Taip	Ne
Integracija su kitais produktais (pvz., Rational Software Architect,)	Taip	Taip	Ne	Ne
Laisvai platinama versija (nemokama)	Taip	Ne	Ne	Taip
Reikalavimų atributų filtravimas užklausomis	Taip	Taip	Taip	Taip
Sekamumo matricos	Taip	Taip	Taip	Taip
Sekamumo medžiai	Taip	Taip	Taip	Taip
Naujų ryšių kūrimas	Ne	Taip	Ne	Ne
Ryšiai saugomi atskiroje lentelėje	Ne	Ne	Taip	Taip
Naujų atributų kūrimas	Taip/Ne	Taip	Ne	Ne
Unikalus identifikavimas nurodytų objektų	Taip	Taip	Taip	Taip
Pranešimai apie pakeitimus	Ne	Taip	Taip	Taip

Taigi apžvelgus reikalavimų valdymo įrankių gerąsias ir blogąsias savybes, nuspręsta pasirinkti RequisitePro įrankį, reikalavimų specifikavimui. Kadangi, IBM firmos RequisitePro produktas yra nemokamas ir laisvai platinamas.

2.10. FR ir NFR reikalavimų integralumo metodų ir principų analizė

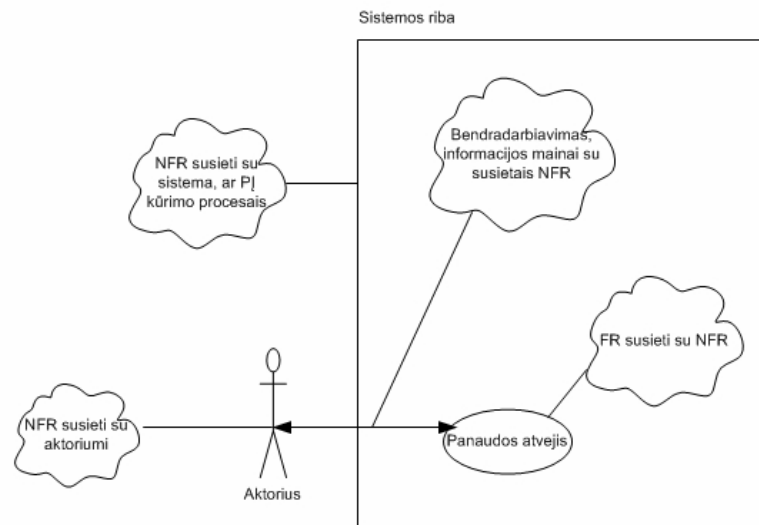
Dabartinės organizacijos daugiausia dėmesio skiria funkciniam reikalavimams (FR), naudodami UML panaudojimo atvejus (*Use Case*) kaip pagrindinę priemonę reikalavimams išgauti ir modeliuoti. Siekiama skatinti specialistus daugiau dėmesio skirti nefunkciniam reikalavimams (NFR), kurie yra labai svarbūs, užtikrinant sklandų perėjimą nuo panaudojimo atvejų modeliavimo. Siekiant tai padaryti susitelkiama kaip integruoti NFR į esamus FR. Reikalavimas vadinamas integruojamu, jei, sujungus jį su kitais reikalavimais, yra gaunamas tarpusavyje suderintų reikalavimų rinkinys [35].

2.10.1. FR ir NFR integralumas, naudojant panaudos atvejų ir tikslais grindžiamą metodą

Lawrence Chung siūlo integruoti NFR su FR į UML panaudojimo atvejų modelį [35]. Kai pateikiami tikslūs NFR, jie integruojami į tam tikrus panaudojimo atvejų diagramos taškus, vadinamus NFR asociacijos taškais. Kadangi reikalavimų inžinieriai naudoja įprastą panaudojimo atvejų modeliavimą, jie taip pat apsveria ir visus reikalingus NFR. NFR gali būti analizuojami, naudodami NFR modelį.

Viena iš praktikos problemų yra NRF aprašymas specialiuose reikalavimų panaudojimo atvejų skyriuose. Reikalavimų inžinierius privalo būti tikras ar NFR yra aprašytas visam panaudojimo atvejui ar tik kuriai nors daliai [35].

Šis modelis naudoja panaudojimo atvejų standartinį modeliavimą. Reikalavimų inžinieriai privalo sujungti NFR su apibendrintu panaudojimo atvejo elementu. Šis modelis nustato savybes, tam kad NFR būtų susieti su specializuotais elementais. 8 paveikslėlyje pateikti NFR asociacijos taškai.



8. pav. NFR asociacijos taškai panaudos atvejų diagramoje[35]

8 paveikslėlyje pateikta UML veiklos diagrama, atvaizduojanti NFR integravimo su FR procesą. Tai pasikartojantis ir tarpinis procesas, kuriame patikslinami projekto artefaktai. Pakartojant ankstesnius žingsnius galima išvengti klaidų [35].

Pirmas žingsnis - nustatomos sistemos ribos ir visi NFR. Nustatomi sistemos apribojimai, kurie sudaromi pagal žmogaus numatytus uždavinius ir sistemos automatizuotus uždavinius. Sistemos ribos gali apimti sistemos dalis, posistemes ar visą sistemą. Kai sistemos ribos nustatytos, nustatomi globaliniai NFR, kurie siejami su sistemos ribomis [35].

Antras žingsnis - identifikuojami aktoriai ir susiję NFR. Aktorių identifikavimas atliekamas pagal išorinius ryšius. Siejame juos Apibendrinimo / Specializavimo ryšiais. Identifikuoti aktoriai siejami su NFR [35].

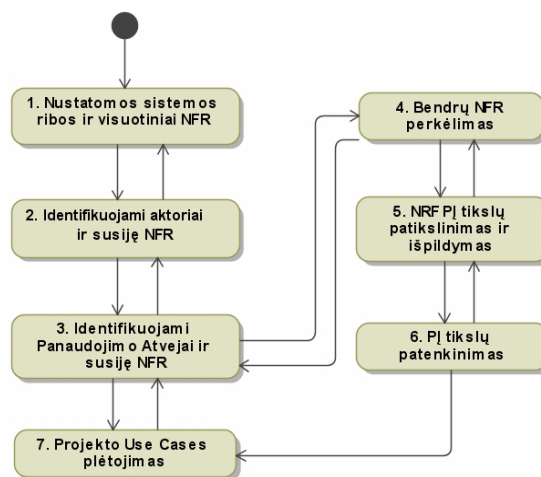
Trečias žingsnis - identifikuojami panaudojimo atvejai ir susiję NFR. Nustatomas sistemos funkcionalumas ir atvaizduojami panaudojimo atvejais, kurie susiejami su Aktoriais. Sistemine panaudojimo atvejus Apibendrinimo / Specializavimo, „Extends“ arba „Include“ ryšiais. Identifikuojamas aktoriaus panaudojimo atvejis, kuris susijęs su NRF [35].

Ketvirtas žingsnis – bendrų NFR perkėlimas. Ankstesni žingsniai identifikuoja NRF, nustatome ar šie NFR gali būti susiję su kitais panaudojimo atvejo modelio elementais. Kiekvienas NFR susietas su panaudojimo atveju gali būti susijęs ir su kitu panaudos atveju. Nustatant vieną ar daugiau panaudojimo atvejų, NFR perkeliame ir susiejame su nauju apibrėžtu, apibendrintu panaudojimo atveju (-ais) [35].

Penktas žingsnis – NFR PĮ tikslų patikslinimas ir išpildymas. Remiantis NFR modeliu, tobulinami NFR PĮ tikslai, naudojant IR (AND) ar AR (OR) dekompoziciją. Nustatomos architektūrinės ar projektinės alternatyvos kiekvienam NFR PĮ tikslo mazgui. Įvertinamos sąlygos ir pasirenkami PĮ tikslai, kurie geriausiai išpildo NRF PĮ tikslus [35].

Šeštas žingsnis – PĮ tikslų patenkinimas. Kai atrinkti PĮ tikslai, parenkami sprendimai, kurie įgyvendina PĮ tikslus. Parenkami sprendimai, kurie geriausiai patenkina kiekvieną PĮ tikslą [35].

Septintas žingsnis – projekto panaudos atvejo plėtojimas. Remiamasi klasių diagrama ir PĮ tikslais. Plėtojama programinės įrangos architektūra, kuri apima posistemių modeliavimą paketų, priklausomybių diagramomis. Elgesio modeliai atvaizduojami sąveikų diagramomis (sekų diagrama), taip atvaizduojant sąveikas tarp posistemių, realizuojant panaudojimo atvejus [35].



9. pav. Veiklos diagrama apibūdinanti integralumo procesą [35]

2.10.2. NFR integravimas į konceptualų modelį

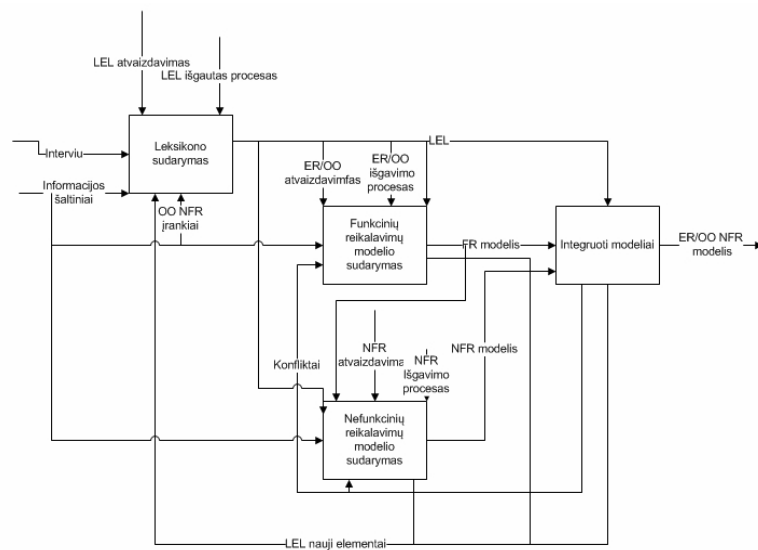
Luiz Marso Cysneiros pasiūlė, integruoti nefunkcinius reikalavimus į konceptualų modelį [8], problema, kaip tinkamai atvaizduoti nefunkcinius reikalavimus.

Šis metodas siūlo naudoti žodyną, kalbos išplėstą leksikoną (*Language Extended Lexicon*) (LEL), integruojant nefunkcinius reikalavimus į esybių (ER) ir objektais grindžiamą (*Object Oriented*) modelį. Tai pasiekama, naudojant leksikono simbolius, sudarant nefunkcinių reikalavimų grafus ir konceptualų modelį (taip pat naudojant ER ar OO modelius). Taigi, bendras leksikono įrašas yra sąsaja nuo vieno vaizdavimo į kitą [8].

Kadangi leksikonas yra pagrindas, sudarant NFR grafus ir konceptualų modelį, tai pirmiausia turi būti prieinama prie LEL.

Kai visi nefunkciniai reikalavimai yra atvaizduoti konceptualiame modelyje, būtinas verifikavimo procesas. Verifikavimas apima nefunkcinių reikalavimų atvaizdavimą grafuose, taip pat ir konceptualiame modelyje. Kiekvienas nefunkcinis reikalavimas esantis viename modelyje turi būti ir kitame [8].

10 paveiksle pavaizduota SADT (*Structured Analysis and Design Technique*) diagrama, atvaizduoja siūlomą metodą. Tai meta-modelis, kuris naudoja ER ar OO modelius [8].



10. pav. SADT diagrama [8]

LEL meta-modelis skirtas išsiaiškinti kalbos naudojimą [8]. Šitas modelis paremtas idėja, kad cirkuliacinis kalbos terminų aprašymas gerina aplinkos supratimą. Kiekvienas žymėjimas yra aprašomas natūralios kalbos sakiniais.

Sąvokos išaiškina simbolio prasmę ir pagrindinius ryšius tarp kitų objektų.

LEL grindžiamas dviem principais: cirkuliaciniu ir minimaliu žodynu. Cirkuliacinis principas teigia, kad sąvokų aprašymai ir elgesio reakcija naudoja kitus leksikos objektus. Minimalaus žodyno principas teigia, kad sąvokų aprašymai, elgesio reakcija ir visi terminai, kurie nėra leksikono objektai, turi būti pateikti žodyne natūralia kalba [8].

Pirmas žingsnis integravimo procese yra perkelti nefunkcinius reikalavimus į NFR grafą, nustatant nefunkcinius reikalavimus LEL simboliais grafe, kurie išreikšti ER klasių diagramoje. LEL naudojamas, sudarant tiek NFR grafus tiek ER ar OO modelius.

Naudojantis taisyklėmis, siūloma integruoti nefunkcinius reikalavimus į konceptualų modelį [8]:

1. Kiekvienas pavadinimas atvaizduotas duomenų klasėje, šakniniame NFR grafe turi būti LEL simbolis. Jeigu nors vienas simbolis nerandamas, kuris reikalingas, LEL atnaujinamas.
2. Kiekvienas objektas, kuris priklauso ER ir kiekvienai klasei, klasių diagramoje turi būti pavadinti naudojant LEL simbolius. Jeigu nerandama simbolio, kuris išreiškia tai ko reikia, LEL atnaujinamas .
3. Kiekvienas nefunkcinių reikalavimų grafas randamas ER ar klasių diagramoje.
4. Ši taisyklė taikoma tik ER modeliui. Programinės įrangos inžinierius gali nuspręsti ar nefunkciniai reikalavimai susiję su objektais ar objektų ryšiais.
5. Ši taisyklė taikoma tik ER modeliui. Visi duomenų atributai, kurie yra nurodyti NFR grafe, turi egzistuoti objekte ar ryšyje. Jeigu taip nėra, reikalavimų inžinierius gali pridėti trūkstamus atributus ar objektus.
6. Ši taisyklė taikoma tik klasių diagramoms. Visi duomenų atributai, kurie nurodyti NFR grafe, privalo būti klasės atributai. Paprastai visi atributai yra susiję su operacijomis. Jei tai neįmanoma, tada programinės įrangos inžinierius prideda trūkstamus atributus ar operacijas į klasę. Programinės įrangos inžinierius patikrina, kuri operacija turi būti sukurta arba, kurioje vietoje turi būti įtrauktas naujas atributas [8].

2.10.3. Esamų sprendimų analizė

Reikalavimų integralumas siejamas su dauguma procesų. 5 lentelėje pateikiami mokslininkų darbai susiję su reikalavimų integravimu.

5. lentelė. Mokslininkų darbai susiję su reikalavimų integravimo metodais

Autorius	Klasių diagrama	Panaudos atvejų diagrama	Sekų diagrama	Priklausomybių diagrama	Būsenų diagrama	Bendradarbiavimo diagrama	Reikalingas UML išplėtimas
Lawrence Chung [5,6]		✓					✓
Ana Moreira [11]		✓		✓			✓
Luiz Marso Cysneiros [5]	✓	✓	✓			✓	✓
Evgeni Dimitrov [13]		✓		✓	✓		

L
awre
nce
Chun
g
siūlo
integr
uoti
NFR
su FR

į UML panaudojimo atvejų modelį [35]. Tai įgyvendinama, naudojant NFR ir jų struktūrą, siejant šiuos NRF su keturiais panaudojimo atvejo modelio elementais: aktoriais, panaudojimo atveju, aktorių panaudojimo atvejo asociacijomis ir sistemos ribomis. Šios asociacijos pavadintos: "Aktoriaus asociacijos taškas", "Panaudos atvejo asociacijos taškas", "Aktoriaus panaudojimo atvejų asociacijos taškas" ir "Sistemos ribų asociacijos taškas".

Ana Moreira pasiūlė modelį, integruojantį kokybės atributus su FR, naudojant UML panaudojimo atvejų diagramą ir sąveikos diagramą [28]. Ji pasiūlė kokybės atributams šabloną (pvz., apibūdinimas, šaltinis), integruojant šiuos kokybės atributus su FR, naudojant standartinės UML diagramas (pvz., panaudojimo atvejų diagramą, sąveikos diagramą) papildant jas specialiomis notacijomis.

Luiz Marso Cysneiros pasiūlė sisteminį požiūrį, užtikrinantį konceptualų NFR modelį, naudojant žodyno pagrindą (LEL), pritaikytą funkciniams ir nefunkciniams sistemos reikalavimų projektavimui [8]. Šis metodas siūlo naudoti žodyną, kalbos išplėstą leksikoną (*Language Extended Lexicon*) (LEL), integruojant nefunkcinius reikalavimus į esybių (ER) ir objektais grindžiamą (*Object Oriented*) modelį.

Evgeni Dimitrov aprašė tris UML grindžiamus inžinerinius metodus. Trys metodai: 1) aspektų vykdymo vaizdavimas, naudojant UML, 2) UML išplėtimas, nagrinėjant vykdymo aspektus ir 3) UML formalaus aprašymo suderinimas. Pasiūlė išplėsti UML panaudojimo atvejų ir būsenų perėjimo diagramas [9].

2.11. Reikalavimų specifikacijos vartotojų analizė

2.11.1. Vartotojų aibė, tipai ir savybės

Vartotojai yra tie žmonės, kurie renka ir specifikuoja reikalavimus iš suinteresuotų šalių (sistemų analitikai – projektuotojai ir t.t.). Išskirtos keturios vartotojų grupės:

Analitikai - tai vartotojų grupė, kurie naudodamiesi specifikacija parenka priimtinausius sprendimus kuriamai sistemai.

Užsakovai – tai vartotojų grupė, kurie pateikia reikalavimus kuriamai IS.

Reikalavimų specifikavimo specialistai – tai vartotojų grupė, kurie yra susiję ir atsakingi už reikalavimų tikslų surinkimą, specifikavimą ir reikalavimų integralumą

2.11.2. Vartotojų tikslai ir problemos

Vartotojų tikslas - galimybė parengti specifikaciją taip, kad ja būtų galima lengvai naudotis. Panaudojant reikalavimų integralumą siekiama susisteminti reikalavimų gausą. Tai padėtų atrinkti reikiamus reikalavimus įvairiais pjūviais pagal vartotojų poreikius. Kadangi reikalavimų yra daug sunku juos valdyti, todėl integruoti reikalavimai turėtų leisti ištraukti tam tikrus reikalavimus iš bendro reikalavimų rinkinio.

Problema - reikalavimų gausa.

Vartotojų siekiami tikslai ir problemos patiktos 6 lentelėje.

6. lentelė. Vartotojų siekiami tikslai ir problemos

Vartotojas	Tikslai	Problemos
IS analitikas	Remiantis sudaryta specifikacija priimti tinkamus sprendimus, IS kūrimui.	Dviprasmiški ir netikslūs reikalavimai.
IS užsakovas	Pateikti kuo tiksliau ir išsamiau sistemos reikalavimus.	Reikalavimų kitimas per visą sistemos kūrimo procesą.
IS reikalavimų specifikavimo specialistas	Integralios specifikacijos rašymas.	Išsibarstę reikalavimai po visą dokumentą.

2.12. Analizės išvados

1. Atlikus reikalavimų analizę pastebėta, kad reikalavimų inžinerijoje vieno universalus reikalavimo apibrėžimo nėra. Siekiant bent dalinai išspręsti šią problemą, priimta reikalavimus klasifikuoti į tipus, tokius kaip verslo reikalavimas, vartotojo reikalavimas, sistemos reikalavimas, techninis reikalavimas ir kt.
2. Apžvelgti reikalavimų klasifikavimo standartai FURPS+ ir ISO 9126, kurie naudojami reikalavimų klasifikavimui. Reikalavimai klasifikuojami, atsižvelgiant į produktų reikalavimus, modeliai plėtojami, įvertinant skirtingas sudėtingas charakteristikas ir parenkant tinkamas metrikas.
3. Atlikta FR ir NFR integralumo analizė, nustatyta, kad panaudojimo atvejų modelis naudojamas, kaip integralumo pagrindas.
4. Atlikus reikalavimų specifikuojimo šablonų analizę, nustatyta, kad Volere šablonas pasižymi teigiamomis savybėmis ir geriausiai išpildo visas reikalavimų specifikuojimo charakteristikas lyginant su kitų šablonų atžvilgiu.
5. Atlikus reikalavimų specifikuojimo įrankių analizę, nustatyta, kad RequisitePro įrankiu galima sėkmingai valdyti pastovius ar nuolat besikeičiančius reikalavimus, pagerinti sekamumą, sieti reikalavimus vienus su kitais, sumažinti projekto riziką ir pagerinti jo kokybę.
6. Atlikus besikertančių reikalavimų analizę pastebėta, kad norint užtikrinti reikalavimų integralumą būtina specifikuoti ir dokumentuoti besikertančius reikalavimus, kurie turi įtakos tiek specifikacijos kokybei tiek galutinio produkto kokybei.
7. Besikertančių reikalavimų identifikavimas iš egzistuojančių dokumentų rankiniu būdu reikalauja daug laiko ir darbo, be to didelė klaidų tikimybė. Be to, kai kurie besikertantys reikalavimai negali būti pilnai suprasti, todėl jie tvarkomi reikalavimų etape. Taigi, geriau valdyti besikertančius reikalavimus iš anksto kada jie yra išgaunami ir kaupiami, negu juos išgauti iš egzistuojančių reikalavimų dokumentų.
8. Atlikus panašių sprendimų analizę, išsiaiškinta, kad reikalavimų integralumas siejamas su daugumą procesų. Pastebėta, kad norint integruoti funkcinius reikalavimus su nefunkciniais reikalavimais reikia išplėsti UML kalbą.

3. Reikalavimų specifikavimo integralumą užtikrinantis modelis

3.1. Reikalavimų integralumą užtikrinančio metodo procesų modelis

Modelis pateiktas 11 paveiksle, užtikrina reikalavimų integralumą. Modelis atvaizduotas hierarchine notacija, kurioje kiekviena veikla turi hierarchinę tvarką.

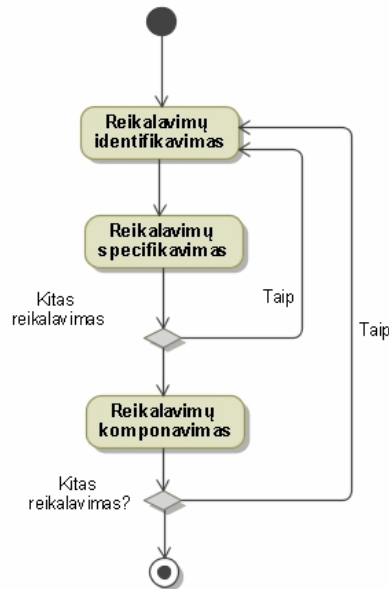
Integralumo modelį sudaro trys pagrindiniai uždaviniai: reikalavimų identifikavimas, specifikavimas ir komponavimas. Kiekvienas uždavinys sudaromas iš smulkesnių uždavinių. Reikalavimų identifikavimo uždavinys susideda iš šešių smulkesnių uždavinių (1.1) Suinteresuotų šalių identifikavimo, (1.2) Šaltinių identifikavimo, (1.3) Reikalavimų išgavimo, (1.4) Pakartotinių katalogų naudojimo, (1.5) Dekompozicijos, (1.6) Reikalavimų klasifikavimo. Reikalavimų specifikavimas sudarytas iš: (2.1) Priklausomybių identifikavimo, (2.2) Kontribucijų identifikavimo, (2.3) Privalomų reikalavimų identifikavimo ir (2.4) Prioritetų identifikavimo. Trečiasis uždavinys sudarytas iš: (3.1) Sąveikos taškų identifikavimo, (3.2) Besikertančių reikalavimų identifikavimo, (3.3) Konfliktų valdymą ir (3.4) Kompozicijos taisyklių nustatymą.



11. pav. Reikalavimų integralumo proceso modelis

Reikalavimų integralumą užtikrinantis procesas gali būti atvaizduojamas, naudojant įvairias priemones. Pasirinkta UML veiklos diagrama šio proceso atvaizdavimui. 12 paveiksle pateikta veiklos diagrama atvaizduojanti užduotis. Kiekviena užduotis yra modeliuojama kaip veiklos būsena, o sprendimo taškai atvaizduojami, naudojant pasirinkimą. Kai naujas reikalavimas identifikuojamas, kartojamas reikalavimų identifikavimo procesas. Procesas

užbaigiamas, kai visi reikalavimai turi šabloną o sąveikos taškai turi nustatytas kompozicijos taisykles.



12. pav. Pagrindinių užduočių veiklos diagrama

3.2. Reikalavimų integralumą užtikrinantis procesų modelis

Metodo elgsenos modeliui atvaizduoti pasirinktos veiklos diagramos. Jų pagalba atvaizduojami reikalavimų specifikacijos modelio užpildymo procesai. Specifikacijos procesų modelis būtinas teisingam duomenų užpildymui. Todėl kiekvieną procesą aprašysime detaliau.

Modelis pavaizduotas 11 paveiksle, užtikrina reikalavimų integralumą. Šis modelis atvaizduoja procesą, susidedantį iš trijų etapų.

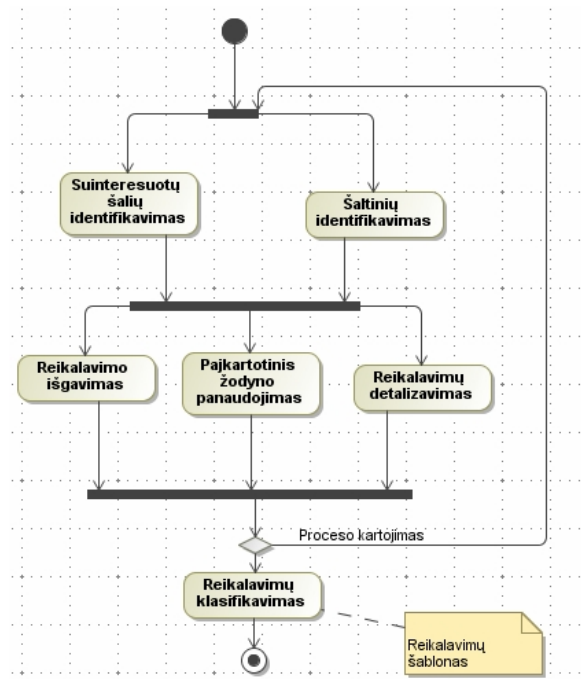
1. Reikalavimų išgavimas, identifikavimas;
2. Reikalavimų specifikavimas;
3. Reikalavimų komponavimas;

3.3. Reikalavimų identifikavimo procesas

Šio proceso tikslas yra nustatyti visos sistemos reikalavimus. Kiekvienas reikalavimas nusako sistemos savybę, kurią turi tenkinti sistema.

Reikalavimų identifikavimo procesas susideda iš: (1) Sinteresuotų šalių identifikavimo, (2) Šaltinių identifikavimo, (3) Reikalavimų išgavimo, (4) Pakartotinių katalogų naudojimo, (5)

Dekompozicijos, (6) Reikalavimų klasifikavimo. 12 paveiksle pateiktas procesas, kuris naudojamas šioms užduotims atlikti.



13. pav. Reikalavimų identifikavimo procesas

Kaip matote paveiksle 13 uždavinius galima atlikti keletą kartų palaipsniui ir kai kuriuos iš jų tuo pačiu metu. Šio proceso pabaigoje gaunamas reikalavimų šablonas su identifikuotais reikalavimais.

Veiklos modelis pavaizduotas 12 paveiksle, atvaizduoja funkcinių ir nefunkcinių reikalavimų identifikavimą.

Reikalavimų gavyba – sudėtingas ir svarbus procesas, kurio metu būtina identifikuoti reikalavimų šaltinius ir iš identifikuotų šaltinių išgauti reikalavimus.

Reikalavimų šaltinių prigimtis, priklauso nuo projekto dalykinės srities. Procesas skirstomas taip:

- Suinteresuotų šalių identifikavimas;
- Šaltinių identifikavimas;
- Terminų žodyno pakartotinis panaudojimas;
- Reikalavimų išgavimas;
- Dekompozicija;
- Reikalavimų klasifikavimas;

Kiekvienas procesas aprašytas detaliau.

Suinteresuotų šalių identifikavimas. Reikalavimų inžinerijos proceso pagrindas suinteresuotų šalių poreikių nustatymas, tikslas identifikuoti visus asmenis, organizacijas, kurie turi įtakos reikalavimams. Todėl suinteresuotos šalys yra asmenys, apimantys užsakovus ar klientus (kurie moka už sistemą), vartotojai (kurie sąveikauja su sistema, darbe), organizacijos (kurios bendradarbiauja su sistema). Identifikuojant suinteresuotas šalis, ieškome asmenų ar organizacijų, kurie [14]:

- vadovauja, pristato, naudojami ar tvarko sistemą po įdiegimo;
- dalyvauja sistemos kūrime kaip: architektai, kūrėjai, testuotojai, kokybės inžinieriai ar projekto vadovai;
- yra atsakingi už verslą ir procesus, kuriuos sistema turi palaikyti;
- turi finansinių interesų, pavyzdžiui, kurie moka arba yra atsakingi už produkto pardavimą;
- veikiami sistemos;

Iš identifikuotų suinteresuotų šalių sąrašo, pasirenkamas atstovaujantis asmuo ar grupė su kuriais galima išsiaiškinti reikalavimus, spręsti konfliktus.

Šaltinių identifikavimas. Šios užduoties tikslas yra surinkti visus esančius dokumentus, pavyzdžiui, organizacijos diagramas, proceso modelius ar standartus, katalogus, kurie gali padėti identifikuoti reikalavimus.

Siekiant išsiaiškinti reikalavimus, analizuojami dokumentai, kurie surinkti šaltinių identifikavimo etape, įtraukiant suinteresuotų šalių interviu užrašus, sistemos srities supratimui. Ieškomi žodžiai ir sakiniai, kurie išreiškia savybes ir užduotis, kurias programa turi vykdyti. Tolimesnis interviu su interesusotomis šalimis yra reikalingas (kai bendros idėjos yra naudojamos bendram objektų orientuotam modeliui).

Reikalavimų išgavimas. Reikalavimų išgavimo technika priklauso nuo laiko ir turimų resursų ir, žinoma, nuo informacijos rūšies. Reikalavimų išgavimo būdai aptarti (2.4 skyriuje).

Siekiant išgauti reikalavimus, analizuojami dokumentai, kurie buvo surinkti šaltinių identifikavimo procese, įtraukiant ir suinteresuotų šalių interviu įrašus. Dokumentuose ieškomi veiksmazodžiai ir daiktavardžiai, kurie išreiškia savybes ar užduotis, kurias sistema turi teikti.

Terminų žodyno panaudojimas. Siekiant palengvinti reikalavimų identifikavimą, siūloma naudoti jau egzistuojančius žodynus, aptartus Chung ir Wieger [7, 35, 39]. Šiose

žodynuose yra informacijos šaltiniai apie nefunkcinių reikalavimų sąvokas ir terminologijas, kurie taip pat prisideda prie pakartotinio panaudojimo. Kiekvienas įrašas žodyne aptariamas, atsižvelgiant į suinteresuotos šalies nuomonę, ar kiekviena sąvoka (iš nefunkcinių reikalavimų) naudojama sistemoje ar ne.

Reikalavimų dekompozicija. Reikalavimai gali būti išskaidyti į paprastesnius reikalavimus. Tai ryšys, kuris skaido aukšto lygio NFR į smulkesnius NFR. Kiekviena dekompozicija – NFR palikuonis (suskaitytas NFR) gali visiškai arba iš dalies atitikti tėvinį. Pvz., valdant transakcijas su saugumu. Saugumo reikalavimas yra gana platus. Nagrinėjant tokį reikalavimą, NFR gali reikėti suskirstyti į smulkesnius, kad būtų rastas optimaliausias sprendimas. Taigi saugumas gali būti suskirstytas į integralumą, konfidencialumą ir prieinamumą ir t.t.

Reikalavimų klasifikavimas. Visų tipų reikalavimai yra svarbūs ir turi būti traktuojami tinkamais terminais. Nepaisant to, reikalavimų klasifikavimas yra svarbus reikalavimų specifikavime.

Reikalavimų klasifikavimas buvo pasiūlytas programinės įrangos literatūroje, pavyzdžiui, tai pasiūlė Sommerville [34]. Atsižvelgiant į tai, metode apibrėžiami du reikalavimo tipai: funkciniai ir nefunkciniai reikalavimai.

Taigi nuspręsta sudaryti šabloną remiantis atliktų šablonų analize. Šablonas sudarytas atsižvelgiant į sudarytą modelį žiūrėti 7 lentelė.

7. lentelė. Reikalavimų specifikavimo šablonas

Reikalavimas	Klasifikavimas	Įvykis / PA
Reikalavimo numeris turi būti unikalus.	Funkciniai, nefunkciniai reikalavimai.	Gali būti pritaikomi PA numeruoti, tačiau gali būti nesutapimų. Reikalavimas susiejamas su PA, nurodant jų numerius.
Aprašymas	Skirtas reikalavimo paskirčiai apibrėžti. Tai tekstinis reikalavimo apibrėžimas. Jame turi atsispindėti užsakovo arba vartotojo pageidavimai. Geriausia apsiriboti vienu aprašymo sakiniu.	

Pagrindimas	Detalesnis reikalavimo aprašymas
Šaltinis	Informacijos šaltinis, pvz., suinteresuotoji šalis, dokumentas, sistemos informacija, katalogas ir verslo procesai.
Tikimo kriterijus	Tai kvantifikuotas tikslas, kurį turi tenkinti sistema. Nors reikalavimas aprašomas vartotojo sąvokomis, tačiau tinkamumo kriterijus rašomas tiksliai nusakant jo išmatuojamumą, kad priimtus sprendimus būtų galima testuoti reikalavimo tenkinimo aspektu. Tai tartum savotiškas standartas kuriamai sistemai, kurio turi prisilaikyti projektuotojas. Tai labai svarbus reikalavimo specifikacijos punktas.
Užsakovo patenkinimas	Užsakovo nepatenkinimas
Patenkinimo rangas atspindi, kiek bus patenkintas užsakovas, jei reikalavimas bus sėkmingai įvertintas. Rangui galima naudoti skalę nuo 1 iki 5. 1 reiškia, kad užsakovas nelabai kreips dėmesį, jei reikalavimas bus įvertintas, o 5 reiškia maksimalų vartotojo patenkinimą reikalavimo įvertinimu. Analogiškai vertinamas ir nepasitenkinimo rangas	
Priklausomybės	Konfliktai
Priklausomybės tai kiti reikalavimai, turintys įtaką nagrinėjamam reikalavimui. Pavyzdžiui, jei vienas reikalavimas keičiasi, tai turi keistis ir kitas. Arba vieno reikalavimo duomenys betarpiškai siejasi su kito reikalavimo duomenimis. Gali būti, kad vienas reikalavimas negali egzistuoti be kito reikalavimo įvertinimo.	Tai reikalavimai, kurie prieštarauja nagrinėjamam reikalavimui.

Papildoma medžiaga	Ne viską reikia pateikti reikalavimo specifikacijoje. Jei yra kas nors papildomai aprašančio reikalavimą, tuomet gali būti panaudota nuoroda į šią medžiagą. Tačiau čia reikia pateikti tik betarpiškai su reikalavimu susijusias nuorodas.
Istorija	Čia užregistruojama, kada reikalavimas pirmą kartą buvo iškeltas, kada pakeistas pašalintas ar patikrintas.
Kompozicijos taisyklės	Kompozicijos taisyklės apibrėžia tvarką kaip reikalavimas bus taikomas konkrečiam sąsajos taškui.
Prioritetų sąrašas	
Suinteresuotų šalių prioritetai	Išreiškiamas reikalavimo svarbumo laipsnis, kurį suteikia suinteresuota šalis.
Kontribucijų sąrašas	
Kontribucija	Kontribucijos ryšys tarp dviejų reikalavimų apibrėžia būdą, kaip vienas reikalavimas veikia kitą.
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Tai reikalavimas, kuris apima visas nagrinėjimo reikalavimo priklausomybes, kad jį patenkintų.

Vienas šablonas kiekvienam reikalavimui bus pildomas iteracijomis, naudojant pasiūlytą procesą. Remiantis aprašu aukščiau, užpildomos eilutės su pavadinimu, šaltiniu, suinteresuota šalim, aprašu, istorija, užsakovo tenkinimu, tikimo kriterijus. Detalizuojamas ir klasifikuojamas kiekvienas identifikuotas reikalavimas šablone. Likusios šablono dalys užpildomos reikalavimų specifikavimo procese.

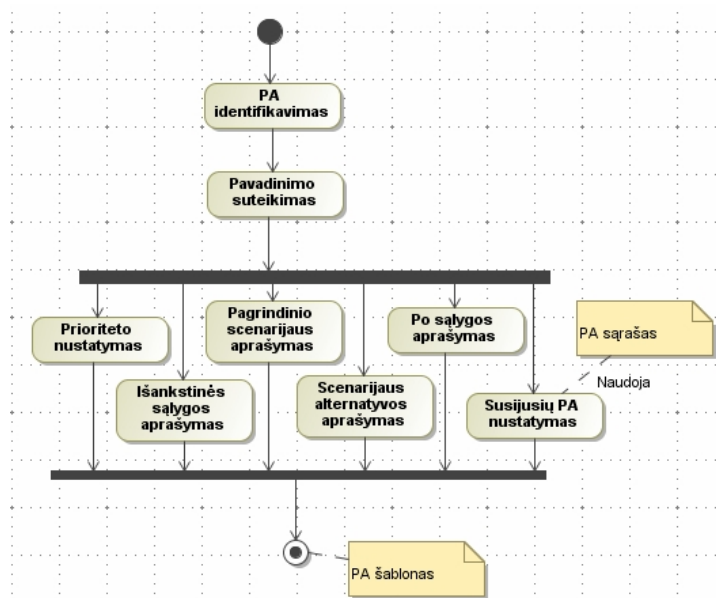
3.4. Panaudojimo atvejų aprašymo veiklos modelis

Sekantis žingsnis, rašant reikalavimų specifikaciją yra panaudos atvejų specifikavimas. Šioje veikloje, giliname kiekvieną sistemos panaudojimo atvejį į išsamias funkcinės savybes, išreiškiant panaudojimo atvejus su tekstiniu aprašu. Taigi, kiekvienas FR yra susietas su vienu ar

daugiau panaudojimo atvejų. Rezultatas, kiekvienas panaudos atvejis turi aprašą lentelė 8. Tokį 8 lentelės formatą pasiūlė C. Larman [23]. Panaudojimo atvejų veiklos diagrama pateikta 14 paveiksle.

8. lentelė. Šablonas panaudos atvejams specifikuoti [23]

Panaudos atvejis (<i>Use Case Nr</i>)	Unikalus panaudos atvejis
Pavadinimas (<i>Name</i>)	Panaudojimo atvejo pavadinimas
Prioritetas (<i>Priority</i>)	Panaudos atvejo svarba
Išankstinė sąlyga (<i>Precondition</i>)	Tekstinis sąlygos aprašas, kuris turi būti patenkintas prieš panaudos atvejo įgyvendinimą
Pagrindinis scenarijus (<i>Main scenario</i>)	Išsami seką apibūdinti sąveika tarp vartotojo ir sistemos.
Scenarijaus alternatyva (<i>Alternative scenario</i>)	Išplėtimai ar pakeitimai pagrindinio scenarijaus
Po sąlygos (<i>Postcondition</i>)	Tekstinis sąlygos aprašas, kuris turi būti patenkintas po panaudos atvejo įgyvendinimo.
Susiję panaudos atvejai (<i>Use-cases related to the current use- case.</i>)	Naudojimo atvejų sąsajos su naudojamu panaudos atveju.

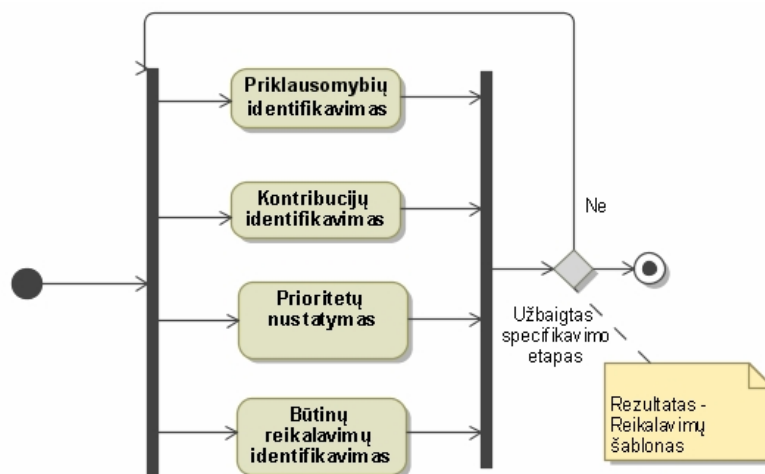


14. pav. Panaudojimo atvejų aprašymo veiklos modelis naudojant C. Larman'o šabloną

3.5. Reikalavimų specififikavimo procesas

Veiklos modelis, pavaizduotas 15 paveiksle, yra neatskiriama bendro specififikacijos užpildymo modelio procesų dalis, tai reikalavimų specififikavimas.

Reikalavimų specififikavimo procesas sudarytas iš: (1) Priklausomybių identifikavimo, (2) Kontribucijų identifikavimo, (3) Būtinų reikalavimų identifikavimo ir (4) Prioritetų identifikavimo.



15. pav. Reikalavimų specifikavimo procesas

Priklausomybių nustatymas. Tai reikalavimų priklausomybių sąrašas, kurį reikalavimas turi įvykdyti. Pagal Wirfs-Brock [40], priklausomybės yra įsipareigojimas įvykdyti užduotį ar sužinoti tam tikrą informaciją. Kadangi reikalavimai gali būti apibrėžti kaip susietų reikalavimų rinkiniai, šie reikalavimai nusakomi - priklausomybėmis. Reikalavimas neturintis priklausomybės, parodo, kad kažkas yra praleista arba kad reikalavimas neturi prasmės, taigi reikia pakartoti reikalavimų identifikavimo procesą, tiksliau nustatyti reikalavimą. Ši informacija yra įrašoma į šablono priklausomybių eilutę.

1 Pavyzdys. Kiekvienam identifikuotam reikalavimui turi būti nustatyta bent viena priklausomybė. Pavyzdžiui, jei norite prisijungti prie sistemos, sistema turi patikrinti vartotojo vardą ir slaptažodį.

Kontribucijos nustatymas. Kontribucija, tai ryšys tarp dviejų reikalavimų, kuris apibrėžia būdą, kaip vienas reikalavimas veikia kitą. Kontribucijos gali būti bendradarbiaujančios (teigiamai veikiančios ir padedančios įvykdyti reikalavimą) ir jos atvaizduojamos „+“, arba prieštaraujančios (neigiamai veikiančios ir kliudančios įvykdyti reikalavimą) ir jos atvaizduojamos „-“. Šie ryšiai yra vienos krypties, reiškia jei reikalavimas A daro teigiamą įtaką reikalavimui B, atvirkštine tvarka žymėti nebūtina, kontribucijos lentelė detaliau aptarta K. Wiegerts [39].

2 Pavyzdys. Pavyzdžiui, sąsajos taškas r3 (žr. skyrelį 3.11) turi du vienas kitam prieštaraujančius reikalavimus {r1, r2} ir tik vienas iš jų gali būti įgyvendintas. Šie du reikalavimai yra priešaringi (konfliktuojantys), nes saugumo protokolas (pvz., šifravimo) paprastai sumažina sistemos reakciją (todėl duomenys gali būti neišsiunčiami per 5 sekundes).

Kontribucijos ryšiai tarp kai kurių reikalavimų randami kataloge pagal L.Chung ir K.Wiegers, kitus gali būti sunku identifikuoti ir reikia pasitelkti keletą sričių ekspertus, pavyzdžiui, tokių kaip saugumo ir realaus laiko sistemoms [39, 35].

Šie ryšiai yra įtraukiami į kontribucijos eilutę reikalavimų šablone. Jeigu reikalavimas neveikia kito reikalavimo tada įrašoma <nėra>.

Prioritetų nustatymas. Ne visi reikalavimai yra vienodai svarbūs, todėl jiems reikia suteikti prioritetus. Prioritetas nusako reikalavimo svarbumo laipsnį, kurį suteikia užsakovas. Reikalavimo prioritetas yra kontekstinė priklausomybė.

Reikalavimų inžinerijoje, reikalavimų prioritetams yra daug metodų. Pavyzdžiui, Brackett [5] siūlo skirstyti reikalavimus į tris grupes: privalomas, pageidautinas ir neesminis. Pagal Karlsson'o metodą [19], kiekvienam reikalavimui priskiriamas skaičius (skaičių skalėje nuo 1 iki 5) reiškiantis reikalavimo svarbą, kai 1 - nereikalaujamas (klientui nereikia jo), 2 - nėra svarbus (klientas sutiktų jei jo ir nebūtų), 3 - gana svarbus (klientas tai vertintu), 4 - labai svarbus (klientas norėtų, kad reikalavimas būtų įgyvendintas) ir 5 - privalomas (klientas negali išsiversti be jo) [19].

Metode naudojamas Anos Moreiros pasiūlytą reikalavimų prioritetų nustatymo metodą. Lentelėje 9 pateiktas reikalavimų prioritetų svoriai, bei reikšmės.

9. lentelė. Reikalavimų svarba suinteresuotosioms šalims [28]

Labai svarbu]0.8, 1.0]
Svarbu]0.5, 0.8]
Vidutiniškai]0.3, 0.5]
Nėra taip svarbu]0.1, 0.3]
Nerūpi]0.0, 0.1]

Būtinų reikalavimų identifikavimas. Sudaromas reikalavimų sąrašas, pagal kurį reikalavimas turi įvykdyti savo priklausomybes. Šiame žingsnyje identifikuojami ryšiai tarp visų reikalavimų, apimant FR, NFR ir besikertančius reikalavimus. Jeigu šis reikalavimas nereikalauja jokių kitų reikalavimų, įrašoma <Nėra>. Šis sąrašas įrašomas eilutėje Būtinasis reikalavimas 7 lentelėje. „Būtinų reikalavimų“ ryšiais skiriasi nuo dekompozicijos ryšio, kur reikalavimai yra skaidomi į detalesnius reikalavimus. Jei vienam reikalavimui reikia kito reikalavimo, kad užbaigtų savo elgesį, arba jei reikalavimas yra apribotas kitu reikalavimu.

Pavyzdžiui, „Prisijungimas“ reikalavimui reikia reikalavimo „Atsako laikas“, nes sistema turi reaguoti laiku, kai vartotojas prisijungia.

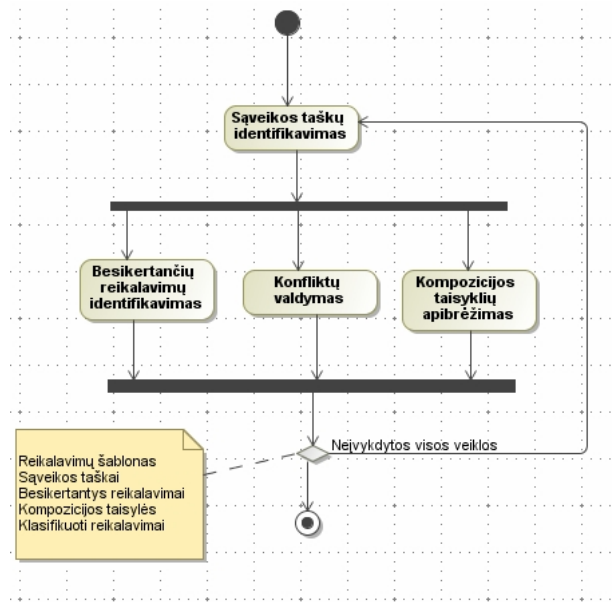
Pavyzdys. Reikalavimui „Atnaujinti“ būtinas reikalavimas yra „Informacijos Patvirtinimas“, norint įvykdyti reikalavimo priklausomybes.

Šie žingsniai gali būti vykdomi lygiagrečiai kiekvienam identifikuotam reikalavimui. Šis procesas užbaigiamas, kai visi reikalavimai turi užbaigtą šabloną.

3.6. Reikalavimų komponavimo procesas

Sekantis procesas, tai reikalavimų komponavimas. Šios užduoties tikslas yra sukomponuoti reikalavimus ir suteikti kūrėjams visą sistemos vaizdą bei nustatyti ir valdyti konfliktus tarp reikalavimų. Komponavimo procesas sudarytas iš: (1) Sąveikos taškų identifikavimo, (2) Besikertančių reikalavimų identifikavimo, (3) Konfliktų valdymo ir (4) Kompozicijos taisyklių nustatymo.

Paveikslas 16 iliustruoja procesą šiems žingsniams, uždavinius galima atlikti keletą kartų, palaipsniui ir kai kuriuos iš jų tuo pačiu metu. Be to, palaikomas kompozicijos plėtojimas, neatsižvelgiant į reikalavimo prigimtį. Tai reiškia, kad galima komponuoti besikertančius su besikertančiais reikalavimais, taip pat, besikertantį su nesikertančiu reikalavimu. Šis procesas baigiamas, kai visi sąveikos taškai turi kompozicijos taisykles. Šio proceso pabaigoje gaunamas užbaigtas reikalavimų rinkinio šablonas su nustatytais sąveikos taškais, besikertančiais reikalavimais, reikalavimus su prioritetais ir kompozicijos taisyklėmis.



16. pav. Reikalavimų komponavimo procesas

Sąveikos taškų identifikavimas. Nustatomi taškai, kuriems reikalingos kompozicijos taisyklės. Šiuos taškus vadiname – sąveikos taškais. Sąveikos taškas yra taškas, kuriame vienas ar keli reikalavimai, turi būti sukomponuoti kartu. Jis sudarytas iš reikalavimų rinkinio, kurie turi būti surinkti kartu. Vienas iš reikalavimų atlieka pagrindinio reikalavimo vaidmenį, kuris „suriša“ likusius reikalavimus. Tai gali būti atvaizduojama dviejų matmenų lentele, žiūrėti 10 lentelę.

10. lentelė. Sąveikos taškų identifikavimas

Reikalavimas ($R_i = 1, \dots, n$)	Būtinasis reikalavimas ($R_i, i = 1, \dots, n$)					Sąveikos taškas ($MP_i, i = 1, \dots, n$)
	R_1	R_2	R_3	...	R_n	
R_1		✓	✓			$MP_1 R_1 = (R_1, R_2, R_3)$
R_2			✓			$MP_2 R_2 = (R_2, R_3)$
...						
R_n		✓				$MP_n R_n = (R_n, R_2)$

Lentelė yra pildoma simboliu "✓", jei stulpelyje „Reikalingas reikalavimas“ yra reikalaujamas eilutės „Reikalavimai“. Taigi, sąveikos taškas reikalavimui R_1 yra (MP_1), sudaromas reikalavimų rinkinys, kurie turi būti susieti su R_1 . Pavyzdžiui, $MP_1 = \{R_1, R_2, R_3\}$.

3 Pavyzdys. Pavyzdžiui, kompozicijai parenkamas apribojantis reikalavimas $r_1 =$ "sistema turėtų naudoti saugumo protokolą, kai duomenys siunčiami internetu", kuris kerta pagrindinį reikalavimą $r_3 =$ "Vartotojo vardas ir slaptažodis siunčiami į serverį". Be to, kitai kompozicijai parenkamas apribojantis reikalavimas $r_2 =$ "atsakymo laikas neturi viršyti 5 sekundžių", kerta pagrindinį reikalavimą r_3 . Taigi, reikalavimo r_3 sąveikos taškas yra $\{r_1, r_2\}$.

Kai kompozicijos taisyklės apibrėžiamos kiekvienam sąveikos taškui, identifikuojami reikalavimai suinteresuotosioms šalims, kurie komponuojami kiekvienam sąveikos taškui, konfliktų sprendimui. 11 lentelėje pateiktos susijusios suinteresuotosios šalys su sąveikos taškais, kur kiekviena eilutė pateikia reikalavimų sąrašą suinteresuotosioms šalims ir sąveikos taškams.

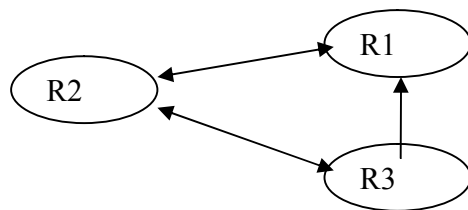
11. lentelė. Suinteresuotųjų šalių susiejimas su sąveikos taškais

Suinteresuotoji šalis	Sąveikos taškai ($MP_i = 1, \dots, n$)			
	MP_1	MP_2	...	MP_m
S_1	$R_1 R_2$			$R_2 R_4$
S_2	R_2			
...				
S_n		$R_3 R_k$		$R_3 R_4$

Besikertančių reikalavimų identifikavimas. Besikertantis reikalavimas yra reikalavimas, kuris negali būti atvaizduotas be pasirinktos dekompozicijos.

Reikalavimas yra besikertantis, jei jis būtinas dvejims ar keliems kitiems reikalavimams. Besikertančių reikalavimų uždavinys užbaigiamas, atsižvelgiant į informaciją, skiltyje "Būtinasis reikalavimas" 7 lentelėje, t.y., jeigu daugiau nei vienas laukelis yra užpildytas su "✓" reikalavimų stulpelyje "Būtinasis reikalavimas" tai jis yra besikertantis reikalavimas. Pavyzdžiui, R2 yra besikertantis reikalavimas, nes jis būtinas R1 ir R3.

3 Pavyzdys. $r1 = \text{"Sistema turėtų naudoti saugumo protokolą, kai duomenys siunčiami internetu"}$, $r2 = \text{"Vartotojo vardas ir slaptažodis siunčiami į serverį"}$. $r3 = \text{"Atsakymo laikas neturi viršyti 5 sekundžių"}$. Šiuo atveju besikertantis reikalavimas yra reikalavimas r3.



17. pav. Besikertančio reikalavimo pavyzdys

Konfliktų valdymas. Šio uždavinio tikslas yra nustatyti ir išspręsti konfliktines situacijas tarp reikalavimų. Konfliktinės situacijos nustato kūrėjai, konfliktai gali atsirasti tam tikrose sąveikos taškuose. Sąveikos taškai nusako konkrečias vietas reikalavimuose, kur reikalavimas (besikertantis ar nesikertantis) turėtų būti išpildytas. Atsižvelgiant į tai, konfliktai iškyla bet kuriuo metu tarp dviejų ar daugiau reikalavimų, kurie veikia vienas kitą neigiamai (žr. kontribucijos eilutė lentelėje 7.), ir turi tokį pat prioritetą, taip pat turi priklausyti tam pačiam sąveikos taškui.

Siekiant išspręsti konfliktus, siūlomos šios veiklos:

- 1) **Aptikti konfliktus.** Iš pradžių turime nustatyti sąveikos taškus, kurie turi konfliktuojančius reikalavimus (t.y. reikalavimus, kurie turi neigiamą įnašą su kitu besikertančiu reikalavimu pagrindiniame reikalavime).
- 2) **Išspręsti optimizacijos problemą.**

Programinės įrangos projekte, reikalavimų inžinieriai visada turi siekti maksimalaus suinteresuotų šalių pasitenkinimo, nepažeidžiant turimų išteklių. 1 lygtis išreiškia suinteresuotųjų šalių pasitenkinimą, atsižvelgiant į suinteresuotųjų šalių svarbumo lygį ir prioritetus.

Lygtis 1 gali būti naudojamos, pasirenkant reikalavimus, taip, kad maksimaliai padidintumėte suinteresuotų šalių pasitenkinimą. Be to, reikalavimas kiekviename sąveikos taške gali būti rūšiuojamas pagal suinteresuotųjų šalių pasitenkinimą, kuriant reikalavimų rangą.

$$\sum_{n \in S} \sum_{r \in M_i} svarba(s_i) * prioritetas(s_i, r_i) * X_i \quad (1) [40]$$

Suinteresuotųjų šalių prioritetai: prioritetas: $S \times R \rightarrow R$. Naudojami svoriai suinteresuotų šalių prioritetams išreikšti.

Suinteresuotųjų šalių svarbumo lygiai: svarbumas: $S \rightarrow R$. Naudojami svoriai suinteresuotų šalių svarbumo lygiams išreikšti.

Kur x_i yra sprendimo kintamasis, kuris įgauna reikšmę 1, kai reikalavimas turi būti įgyvendintas ir reikšmę 0, kai ne.

- 3) **Derybos su interesuotosiomis šalimis:** Jei sąveikos taškas vis dar turi konfliktinių reikalavimų, tuomet reikalingos derybos tarp suinteresuotųjų asmenų. Ankstesnės veiklos rezultatas gali būti naudojamas, siekiant padėti suinteresuotosioms šalims spręsti konfliktus ir susitarti dėl rezultato. Pavyzdžiui, suinteresuotoji šalis gali sutikti su mažesniu svoriu, kuris išreiškia prioritetą [40]. Kai visi konfliktai išsprendžiami, specifikacija yra peržiūrima ir pertvarkoma siekiant nustatyti papildomus konfliktus.

5 Pavyzdys. Pavyzdžiui, sąsajos taškas (pavyzdys 3) turi du vienas kitam prieštaraujančius reikalavimus $\{r1, r2\}$ ir tik vienas iš jų gali būti įgyvendintas. Šie du reikalavimai yra prieštaringi (konfliktuojantys), nes saugumo protokolas (pvz., šifravimo) paprastai sumažina sistemos reakciją (todėl duomenys gali būti neišsiųsti per 5 sekundes).

Darant prielaidą, kad šie reikalavimai buvo surinkti iš dviejų suinteresuotųjų šalių ($\{s1, s2\}$), ir kiekviena suinteresuotoji šalis priskyrė prioritetus kiekvienam reikalavimui, pateiktus lentelėje 3.3. Be to, organizacijos atžvilgiu suinteresuotoji šalis $s1$ yra labai svarbi (svarbumo laipsnis - 1), o suinteresuotoji šalis $s2$ nėra tokia svarbi (svarbumo laipsnis - 0,3). Taigi, galime rasti optimalų sprendimą pritaikant 1 lygtį:

$$\sum \sum svarba(s) * prioritetas(s,r) * X_i$$

Galime įgyvendinti tik vieną iš šių reikalavimų, todėl vienintelis galimas sprendinys yra $\{x1 = 1, x2 = 0\}$ ir $\{x1 = 0, x2 = 1\}$. Taigi, optimalus sprendimas yra vienas norint įgyvendinti $r1$ yra $\{x1 = 1, x2 = 0\}$.

Kompozicijos taisyklės. Kompozicijos taisyklės apibrėžia tvarką, kaip reikalavimai bus vykdomi konkrečiame sąveikos taške. Kompozicijos taisyklės traktuojamos dviem skirtingomis perspektyvomis: 1) tvarka, kaip reikalavimas bus sukomponuotas su pagrindiniu reikalavimu, kuris apibrėžia sąveikos tašką; ir 2) vykdymu, kaip reikalavimai bus integruoti su pagrindiniu reikalavimu.

Seka (žymimas $R1 \gg R2$): nurodo seką reikalavimų ir reiškia, kad R2 vykdomas tik tada jei R1 reikalavimas užbaigtas.

Negalimas (žymimas $R1 [> R2$): reiškia, kad R2 pertraukia R1 vykdymą, kai jis pradeda savo vykdymą.

Lygiagretumas (žymimas $R1 || R2$): reiškia lygiagrečius reikalavimų vykdymus, kad R1 reikalavimas turi būti sinchronizuojamas su R2 reikalavimu.

Pasirinkimas (žymimas $R1 [] R2$): reiškia pasirinkimą ir reiškia, kad tik vienas reikalavimas bus patenkintas (R1 arba R2) [5].

6 Pavyzdys. Prieš patenkinant funkcinį reikalavimą „Prisijungimas“, pirmiau reikia patenkinti reikalavimą „Galiojanti Kortelė“. Šie du reikalavimai, bus patenkinti tik tada jei „Tinkamumas“ ir „Daug prisijungimų būdų“ yra garantuoti (lygiagrečiai). Užbaigiant šią kompoziciją siūloma lygiagrečiai naudoti reikalavimus „Atsako laikas“ ir „Tikslumas“. Jeigu kažkas negerai vyksta su aukščiau išvardintais reikalavimais, „Klaidų pranešimas“ nutraukia elgesį ir pradeda savo.

*((Tinkamumas || Daug prisijungimų būdų)
>> ((Galiojanti Kortelė>> Prisijungimas)
||
Atsako laikais
||
Tikslumas))
[> Klaidų pranešimas*

Šios taisyklės leidžia vykdyti kompozicijas aukštesniame abstrakcijos lygmenyje, kur kiekvienas reikalavimas pateikiamas su pavadinimu ir susiejamas su kitu reikalavimu, taikant kompozicijos taisykles. Be to, galimas kompozicijų išplėtimas, nepriklausomai nuo reikalavimų prigimties. Tai reiškia, kad galima sukomponuoti vieną besikertantį su kitu besikertančiu reikalavimu, taip pat, besikertantį su nesikertančiu reikalavimu.

3.7. Reikalavimų integralumą užtikrinantis modelis

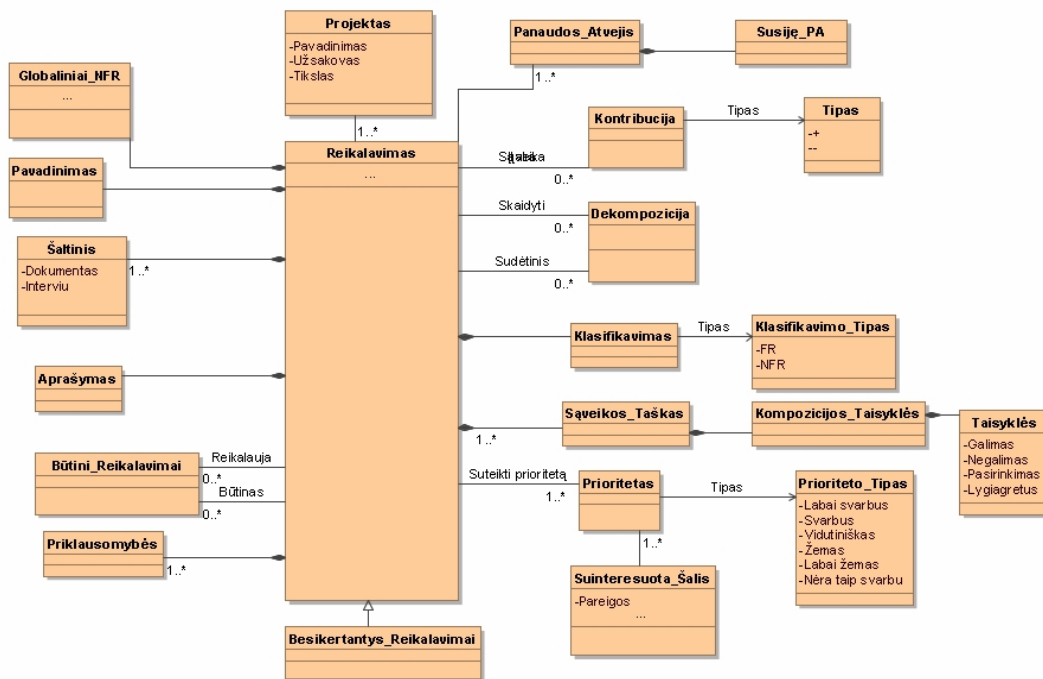
Reikalavimų klasę sudaro: Pavadinimas, Aprašymas, Klasifikavimas (funkcinis arba nefunkcinis, kurie turi savo unikalius identifikatorius), Šaltinis (bent vieną šaltinį) ir Priklausomybių sąrašą (bent vieną priklausomybę), Tikimo kriterijų, Istorija, Užsakovo tenkinimo kriterijų.

Reikalavimas gali turėti kontribucijų sąrašą (nusakomą "+" arba "-")

Reikalavimas gali turėti dekompozicijos sąrašą. Taip pat reikalavimas turi suinteresuotų šalių sąrašą, kurie nustato po vieną prioritetą, (iš: "labai svarbus", "svarbus", "vidutinis", "žemas", "labai žemas" arba "nėra taip svarbu").

Specialus reikalavimų tipas pavadintas - besikertančiu reikalavimu.

Reikalavimai gali turėti būtinų reikalavimų sąrašą. Sąveikos taškas jungia vieną arba kelis „būtinus“ reikalavimus. Remiantis nustatytais sąveikos taškais ir būtinais reikalavimais sudaromos kompozicijos taisyklės. Sąveikos taškams apibrėžiamos kompozicijos taisyklės. Reikalavimų integralumą užtikrinantis modelis pateiktas 18 paveiksle.



3.8. Reikalavimų integralumą užtikrinančio metodo taisyklės

Siekiant taisyklingai užpildyti duomenis duomenų saugykloje reikia vadovautis šiomis taisyklėmis. Sudarytos taisyklės padeda išvengti nereikalingų procesų kartojimą, taip sumažinamos laiko sąnaudos.

Reikalavimai $R = \{R_1, R_2, \dots, R_{|R|}\}$: Reikalavimai renkami iš įvairių suinteresuotųjų šalių (pvz., galutinių vartotojų, programuotojų, vadybininkų), kurie išreiškia skirtingą požiūrį apie sistemą;

Funkciniai reikalavimai $FR = \{FR_1, FR_2, \dots, FR_{|FR|}\}$: Šio tipo reikalavimai nusako vidinį sistemos ar komponento darbą (funkciją) ir yra R poaibis $FR_i \subseteq R$

\subseteq

Nefunkciniai reikalavimai $NFR = \{NFR_1, NFR_2, \dots, NFR_{|NFR|}\}$: Tai reikalavimai, nusakantys sistemos veikimą, bet ne specifinį elgesį ir yra R poaibis $NFR_i \subseteq R$.

\subseteq

Kompozicijos taisyklės $Cr = \{Cr_1, Cr_2, \dots, Cr_{|Cr|}\}$: Kompozicijos taisyklė Cr_j , apibrėžia apribojančių reikalavimų rinkinį $Csj \subseteq R$, kertantį pagrindinį reikalavimų rinkinį Bsj

\subseteq

\subseteq

R .

Suinteresuotosios šalys: $S = \{S_1, S_2, \dots, S_{|S|}\}$;

Suinteresuotųjų šalių prioritetai: prioritetas: $S \times R \rightarrow R$. Naudojami svoriai suinteresuotųjų šalių prioritetams išreikšti.

Suinteresuotųjų šalių svarbumo lygiai: svarbumas: $S \rightarrow R$. Naudojami svoriai suinteresuotųjų šalių svarbumo lygiams išreikšti.

Reikalavimo integralumą užtikrinančio metodo taisyklės:

1. Visi reikalavimai priklauso vienam projektui.

$$\forall r(\text{Reikalavimas}) \rightarrow \exists! p(\text{Projektas}) \wedge \text{Pr iklauso}(p, r)$$

2. Ne visi nefunkciniai reikalavimai aprašomi visai sistemai. Todėl reikalavimai skirstomi į dalinius ir globalinius;

$$\overline{\forall nfr((\text{Nefunkcinis_Reikalavimas}(nfr)) \wedge \text{Sistema}(s)) \wedge \text{Aprašomi_Sistemai}(s, nfr))}$$

$$\Leftrightarrow \exists nfr(\text{Nefunkcinis_Reikalavimas}) \wedge \overline{\text{nusakantis}(s, nfr)}$$

3. Nefunkciniai reikalavimai, kurie nepriskiriami visai sistemai turi būti susieti su panaudos atveju;

$$\overline{\forall nfr((\text{Nefunkcinis_Reikalavimas}) \wedge \text{Sistema}(s) \wedge \text{Pr iklauso}(nfr, s))} \rightarrow \\ \exists pa((\text{PanaudosAtvejis}(pa) \wedge \text{Pr iklausoPA}(nfr, pa))$$

4. Besikertantis reikalavimas yra toks, kuris turi įtakos kito reikalavimo įgyvendinimui; Reikalavimas A sąveikauja su reikalavimu B, jei programinė įrangos dekompozicijoje buvo nustatyta, kad B negali būti išpildytas neįtraukiant A. Taigi, projektavime B gali būti patenkintas jei patenkinamas ir A. Reikalavimai, kurie sąveikauja su kitais, yra apibrėžiami kaip besikertantys reikalavimai (Šiuo atveju A);

5. Konfliktai nustatomi, naudojant matricą, kurioje atvaizduojami reikalavimai, o jų sankirtoje nustatomas reikalavimų poveikis vienas kitam. Jei reikalavimas daro teigiamą įtaką žymimas „+“ ir neigiamą „-“;

6. Konfliktai tai reikalavimai, kurie prieštarauja nagrinėjamam reikalavimui. Tai reiškia, kad vienas NFR (ar FR) poroje turi neigiamą poveikį tam pačiam funkcionalumui.

Pavyzdžiui, reikalavimas, kad “sistema suskaičiuotų trumpiausią kelią iki paskirties vietos”. O kitas reikalavimas gali nurodyti “surasti greičiausią kelią iki paskirties vietos”. Šie reikalavimai gali prieštarauti vienas kitam, kadangi trumpiausias kelias ne visada yra greičiausias kelias;

7. Konfliktų sprendimas pagrįstas ekspertų, kūrėjų sprendimu;

8. Reikalavimams nustatomos priklausomybės, tai reikalavimai, turintys įtaką nagrinėjamam reikalavimui.;

$$\overline{\forall r(\text{Reikalavimas}(r) \rightarrow \exists p(\text{Priklausomybė}(p) \wedge \text{Pasižymi}(p, r))}$$

Pavyzdžiui, jei vienas reikalavimas keičiasi, tai turi keistis ir kitas. Arba vieno reikalavimo duomenys betarpiškai siejasi su kito reikalavimo duomenimis. Gali būti, kad vienas reikalavimas negali egzistuoti be kito reikalavimo įvertinimo.

$$r1(\text{Reikalavimas1}) \wedge \exists p(\text{Priklausomybė}(p, r2(\text{Reikalavimas2}))) \\ \rightarrow r1(\text{Reikalavimas1_negzistuoja_be_priklausomybė}(p, r2))$$

9. Visi reikalavimai turi turėti kokia nors sąsają, reikalavimas negali būti nepriklausomas nuo nieko;

$$\forall R(\text{Reikalavimas}(R) \rightarrow \exists r(\text{Ryšys}(r)) \wedge \text{Sieja_su_kitu_reikalavimu}(r, R))$$

10. Kiekvienas reikalavimas turi turėti unikalų identifikatorių;

$$\forall r(\text{Reikalavimas}(r) \rightarrow \exists! i(\text{Identifikatorius}(i) \wedge \text{Identifikuoja}(i, r)))$$

11. Kiekvienas reikalavimas privalo turėti tikimo *kriterijų*. Jis priklauso nuo atliekamo veiksmo.

$$\forall r(\text{Reikalavimas}(r) \rightarrow \exists tk(\text{TikimoKriterijus}(tk) \wedge \text{Nusakomas}(tk, r)))$$

Pavyzdžiui, jei veiksmas yra įrašyti tam tikrus duomenis, tuomet tikimo kriterijus galėtų numatyti, kad būtų galimybė duomenis atrinkti, o be to jie turi tenkinti tam tikrą standartą. Atliekant skaičiavimus, gauti rezultatai turi atitikti numatomus rezultatus;

12. Reikalavimai turi turėti šaltinius. Šaltinis - tai asmuo, kuris iškelė reikalavimą. Tradiciškai tai tam tikromis žiniomis (susijusiomis su reikalavimu) pasižymintis ir pareigas užimantis asmuo;

$$\forall r(\text{Reikalavimas}(r) \rightarrow \exists sal(\text{Šaltinius}(sal) \wedge \text{Nusakomas}(sal, r)))$$

13. Sąveikos taškas yra apribojančių reikalavimų rinkinys, kuris kerta tą patį reikalavimą. Tegul

kompozicijos taisyklė Crk apibrėžia apribojančių reikalavimų rinkinį $Csk \subseteq R$, kuris kerta

pagrindinį reikalavimų rinkinį $Bsk \subseteq R$. Tegul Sci būna kompozicijos rinkinys, kur ri yra

pagrindinis reikalavimas. Taigi, reikalavimo ri sąveikos taškas yra apibrėžiamas:

$$Mi = \prod_{k \in Sci} Csk$$

14. Sąveikos taškas yra pagrindas, siekiant aptikti ir išspręsti konfliktus tarp sukomponuotų reikalavimų, nes taškas aiškiai nurodo sąveiką tarp reikalavimų;
15. Reikalavimas negali tapti būtinu reikalavimu savaime;
16. Būtinasis reikalavimas negali būti kartojamas būtinam reikalavimui;
17. Reikalavimas gali turėti vieną ar kelis prioritetus, bet suinteresuota šalis gali nustatyti prioritetą tik vieną kartą vienam reikalavimui.
18. Kompozicijos taisyklės visada gali būti apibrėžiamos sąveikos taškui;

$$\forall st(\text{Sąveikos_Taškas}) \rightarrow \exists kt(\text{Kompozicijos_Taisyklė}(kt) \wedge \text{Nusakomas}(kt, st))$$
16. Reikalavimas gali neturėti kompozicijos taisyklės;

3.9. Reikalavimų integralumo metodikos apibendrinimas

1. Pateiktas metodas užpildo spragą, tradicinių reikalavimų inžinerijos metoduose, teikia sistemingą būdą identifikuojant, modeliuojant, atvaizduojant ir komponuojant reikalavimus.
2. Metodas, artimas kitiems reikalavimų metodams, susideda iš: išgavimo, analizės ir specifikavimo, konfliktų sprendimo ir reikalavimų valdymo. Pagrindinis skirtumas tarp šio metodo ir klasikinių reikalavimų metodų, kad šis metodas valdo visus reikalavimus, įskaitant ir besikertančius reikalavimus. Norint tai pasiekti reikia:
 - Veiksmingų egzistuojančių priemonių, reikalavimams identifikuoti, reikalavimų dokumentuose;
 - Gebėti modeliuoti reikalavimus;
 - Gebėti išgauti išsamius ir nuoseklius visų reikalavimų aprašymus bei atvaizduoti juos;
 - Gebėti komponuoti reikalavimus, aiškiai suprasti sistemą, analizuoti ir identifikuoti kritiškus reikalavimus;
 - Proceso, leidžiančio išspręsti konfliktus tarp reikalavimų ir suinteresuotų šalių;
 - Galimybė atsekti reikalavimus, vėlesnėse kūrimo stadijose ar jų šaltinius;
3. Pateiktas metodas apima:
 - Apima besikertančių reikalavimų identifikavimą, funkcinių ir nefunkcinių reikalavimų;
 - Išvengia atsiskyrimo (skilimo) požymių. Tai pasiekama, laikant visus reikalavimus vienoje formoje;
 - Konfliktų identifikavimą ir sprendimą.

4. Metodą sudaro trys pagrindiniai uždaviniai: reikalavimų identifikavimas, specifikuojimas ir jų komponavimas. Reikalavimų identifikavimas yra pasiekiamas, išsamia problemų analize darbo dokumentuose, egzistuojančiuose kataloguose ar kitose informacijos šaltiniuose, kurią pateikia suinteresuotos šalys. Visi identifikuoti reikalavimai yra specifikuojami naudojant unikalų šabloną, kuris aprašo visų tipų reikalavimus, besikertančius ir nesikertančius. Toks reikalavimų specifikuojimo būdas apimantis visus reikalavimus ir jų specifikuojimas vienoje formoje užtikrina reikalavimų integralumą.
5. Pateiktas metodas suteikia galimybę sudaryti integralią reikalavimų specifikuojimą, reikalavimų kompozicijai panaudojant sąveikos taškus ir taisykles. Taisyklių naudojimas reikalavimų rinkiniams, sistemos projektuotojams padeda lengviau ir greičiau suvokti bei susisteminti reikalavimus. Metodas padidina klaidų reikalavimų specifikuojimo aptikimo ir pašalinimo efektyvumą. Konfliktai tarp reikalavimų aptinkami ir sprendžiami ankstyvojoje projekto stadijoje
6. Reikalavimų specifikuojimo užpildymo modelio vykdomų procesų atvaizdavimui pasirinktos veiklos diagramos. Kiekvienas procesas pavaizduotas atskira veiklos diagrama. Specifikuojimo procesų modelis būtinas teisingam duomenų užpildymui, taip užtikrinant reikalavimų integralumą.

4. REQUISITE PRO IR MAGIC DRAW ĮRANKIŲ PRITAIKYMAS METODUI

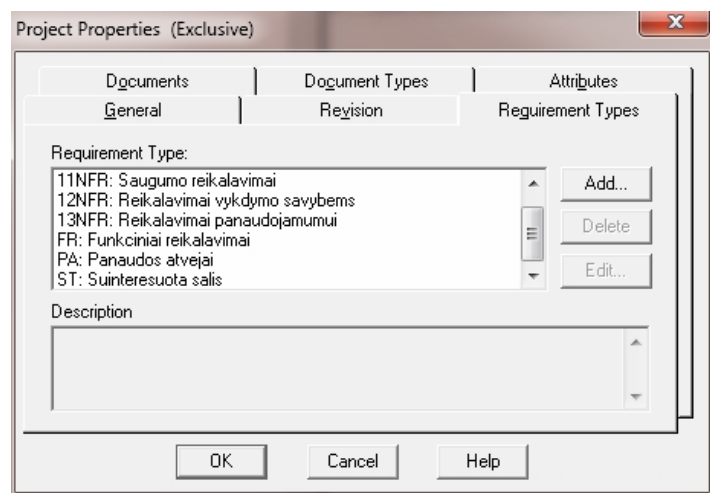
Metodo realizacijai pasirinktos priemonės yra RequisitePro ir MagicDraw įrankiai. RequisitePro įrankis naudojamas reikalavimams sekti ir valdyti, o MagicDraw grafiniam reikalavimų atvaizdavimui, remiantis sudaryta metodika. Siekiant įgyvendinti sukurtą metodą būtinas naujų elementų ir atributų sukūrimas įrankiams.

Visų pirma reikalinga dalykinė sritis, kuriai bus renkami testiniai reikalavimai bei sudaryta reikalavimų specifikacija remiantis sukurta metodika.

4.1. Šablono kūrimas RequisitePro įrankiu

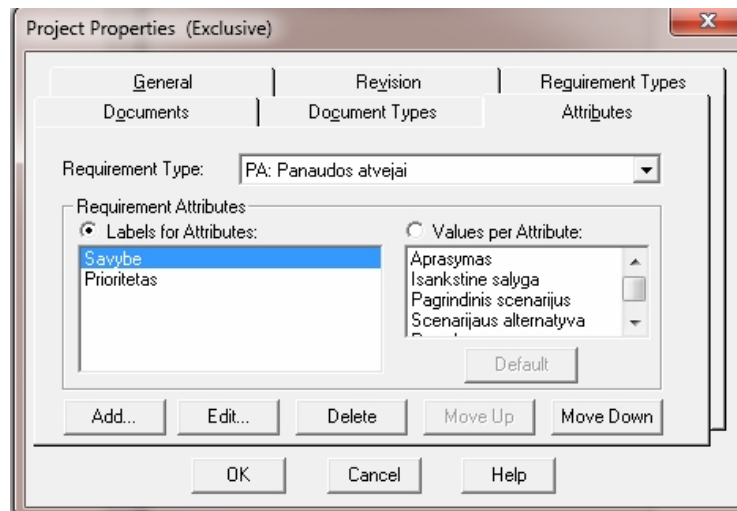
Reikalavimų valdymui sudarytas šablonas RequisitePro įrankiu. Šablonas kurtas, atsižvelgiant į reikalavimų tipus ir jų atributus naudojamus metodikoje.

Naudojantis RequisitePro *Project Properties* pagalba sukurti reikalavimų tipai, kurie turi pavadinimą ir pavadinimą atitinkantį prefixą. Reikalavimai klasifikuojami skyriais pagal reikalavimo tipą, atsižvelgiant į metodiką, kuri naudoja du reikalavimų tipus: funkcinis ir nefunkcinis reikalavimus. Nefunkciniai reikalavimai klasifikuojami remiantis Volere šablonu ir Chungo katalogu. Reikalavimai identifikuojami parametrais: numeriu, tipu ir panaudojimo atvejo numeriu. 19 paveiksle pateikti sukurtų reikalavimų tipai.



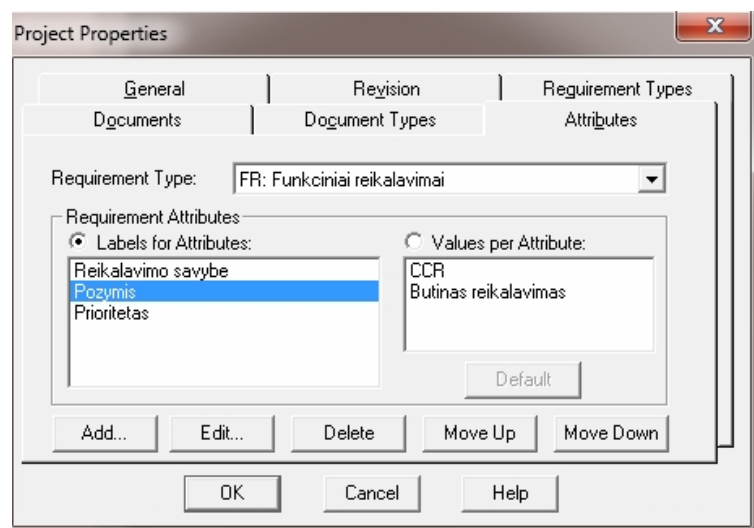
19. pav. Sukurti reikalavimų tipai RequisitePro įrankyje

Remiantis aprašytu metodu, kiekvienam reikalavimo tipui sukurti atributai ir atributų reikšmės, būtinos reikalavimų specifikavimui. 20 paveiksle pateiktas panaudos atvejų reikalavimo atributai ir atributų savybės. Panaudos atvejų reikalavimų atributai sukurti remiantis C. Larman'o šablonu. Kiekvienas panaudos atvejis turi unikalų numerį. Panaudojimo atvejo atributų reikšmės aprašytos 8 lentelėje.



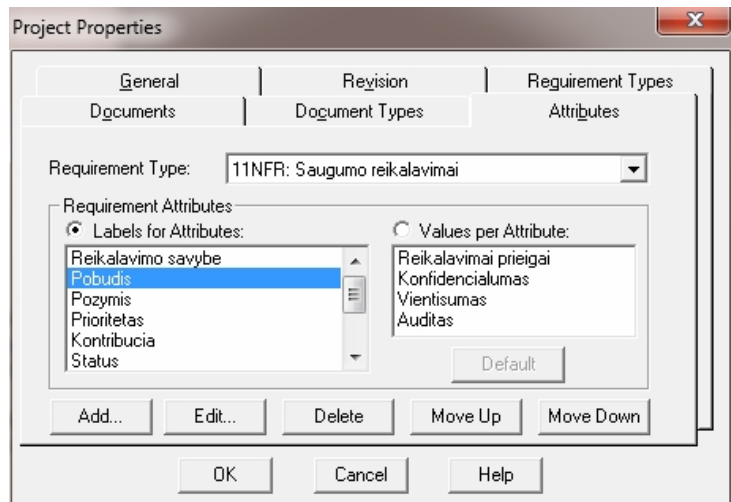
20. pav. Panaudos atvejo atributai ir reikšmės

Funkciniams ir nefunkciniams reikalavimams sukurti atributai ir savybės reikalingos metodikai išpildyti (žr. 7 lentelę). 21 paveiksle pateikti funkcinų reikalavimų atributai ir savybės. Funkcinių reikalavimų atributai: reikalavimo savybė, požymis ir prioritetas. Atributas reikalavimo „Požymis“ nusako reikalavimo charakteristiką apibrėžtą metodikoje, tai būtinas reikalavimas ir besikertantis reikalavimas. Reikalavimas gali turėti vieną, abu arba nei vieno iš šių požymių.

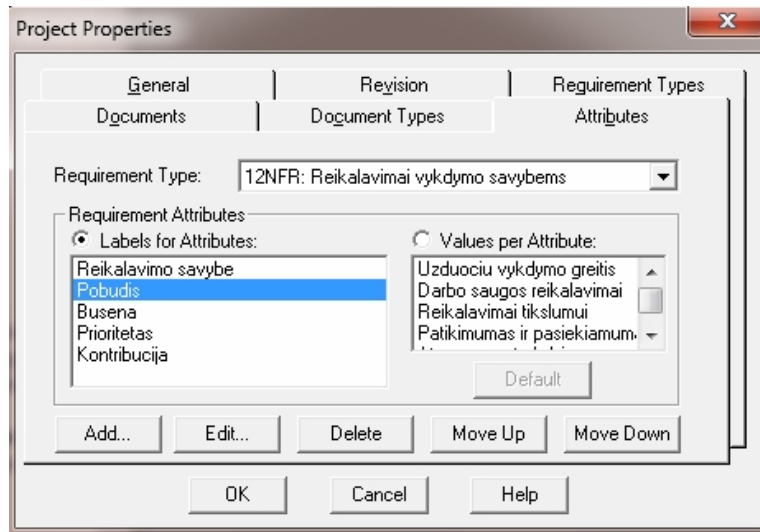


21. pav. Funkcinių reikalavimų atributai ir reikšmės

Nefunkciniai reikalavimai skaidomi į potipius. Kadangi dauguma nefuncinių reikalavimų nusakomi painiais ir stambiais reikalavimais, pavyzdžiui, saugumas, prieinamumas, naudojamumas ir kt. Taigi reikalavimas saugumas turi potipius: reikalavimai prieigai, konfidencialumas, auditas ir vientisumas. 22, 23 paveiksle pateikti nefunkcinio reikalavimo atributai ir reikšmės. Nefunkcinio reikalavimo atributai: reikalavimo savybė, pobūdis, prioritetas, kontribucija. Reikalavimai turi savybes (pagrindimą, šaltinį ir tikimo kriterijų). Norint šias savybes tvarkyti *RequisitePro* aplinkoje ir jomis manipuliuoti, kiekviena jų gali būti fiksuojama kaip atskiras reikalavimas (pasirinkus atitinkamą atributo *Savybė* arba *Reikalavimo savybė* reikšmę). Nefunkciniai reikalavimai kaip ir funkciniai turi požymio atributą, kuris nusako ar tai besikertantis ar būtinas reikalavimas.

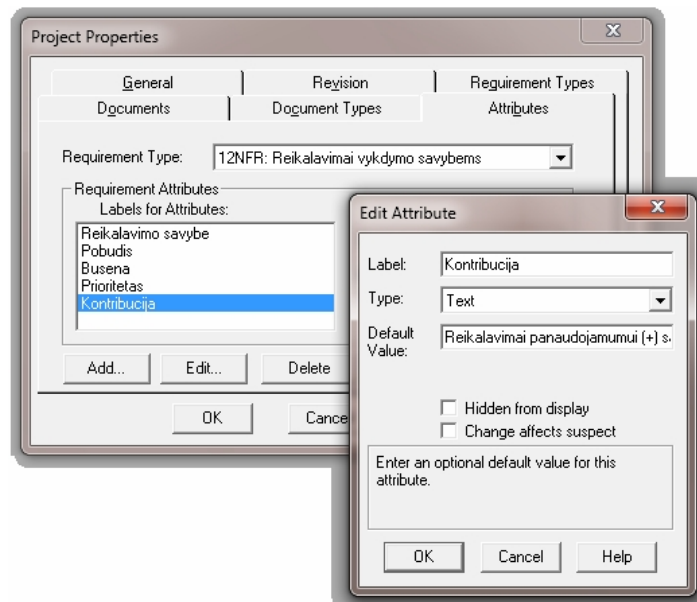


22. pav. Nefunkcinio reikalavimo „Saugumas“ atributai ir reikšmės



23. pav. Nefunkcinio reikalavimo „Vykdymo savybių“ atributai ir reikšmės

Nefunkciniai reikalavimai turi kontribucijų atributą su iš karto nurodytu poveikiu reikalavimui, poveikis nustatytas remiantis Chungo [23] katalogu 24 paveikslas. Kontribucijos naudojamos konfliktams nustatyti tarp reikalavimų. Šiuo atveju reikalavimui „Vykdymo savybės“ teigiamą įtaką daro reikalavimo tipui „Panaudojamumas“. Visiems nefunkciniams reikalavimų tipams, nustatyta galima kontribucijos įnaša.



24. pav. Nustatyta kontribucijos tipas reikalavimui „Vykdymo savybė“

Sudarytas šablonas turintis reikalingus reikalavimų tipus ir atributus atitinkančius sudarytą metodą. Šablonas skirtas reikalavimams specifikuoti pagal metodikoje sudarytą šabloną IBM Rational RequisitePro aplinkoje. Jo pagalba galima MS Word dokumente aprašytus reikalavimus sukelti į reikalavimų tvarkymo įrankio aplinką, priskiriant juos reikiamam tipui ir nurodant būtinas charakteristikas.

Šablono sudėtis

Šabloną sudaro šie reikalavimų tipai:

1. 1SS: Suinteresuota šalis
2. 8PA: Panaudojimo atvejis
3. 9FR: Funkciniai reikalavimai
4. 10NFR: Sistemos išvaizda
5. 11NFR: Saugumo reikalavimai
6. 12NFR Reikalavimai vykdymo savybėms
7. 13NFR: Reikalavimai panaudojamumui
8. 14NFR: Veikimo reikalavimai

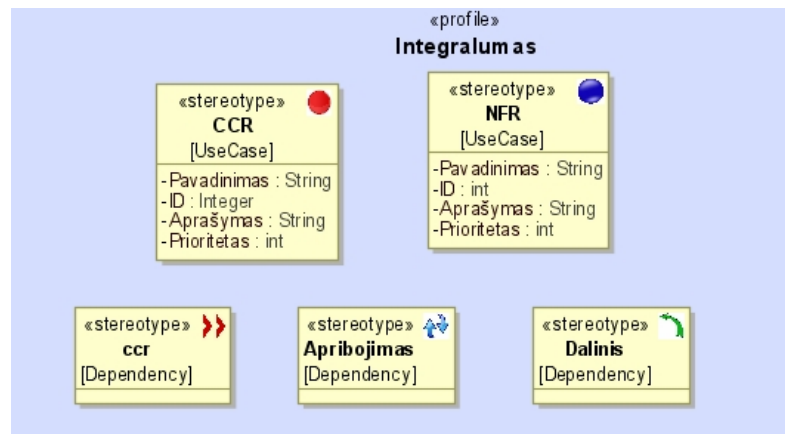
4.2. Reikalavimų integralumo metodo atvaizdavimas UML

4.2.1. Reikalavimų integralumo profilio kūrimo procesas

Reikalavimų integralumo profilis buvo kuriamas, atsižvelgiant į metodikoje naudojamus ryšius ir atributus. Tai atlikus ruošiamas UML profilis ir sukuriamas reikalavimų integralumo modulis, sukuriama reikiama diagrama ir jos elementai.

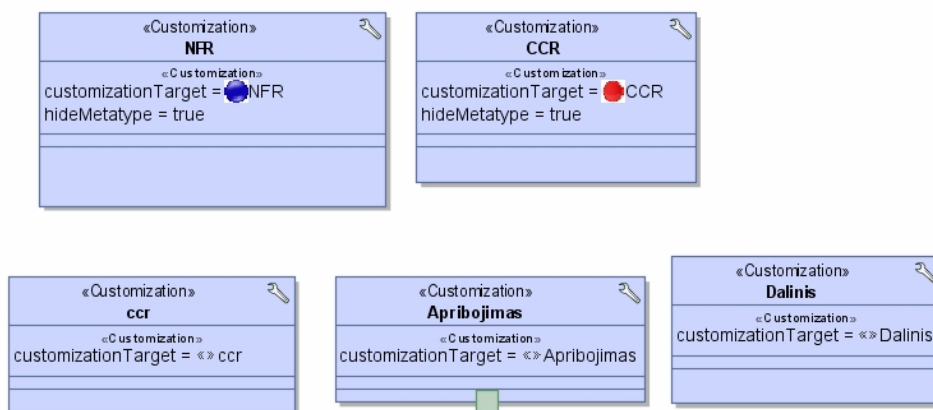
4.2.2. Stereotipų modulio kūrimas

25 paveiksle pavaizduoti sudaryti stereotipai. Detalus stereotipų aprašymas pateiktas 12 lentelėje.








25. pav. Metodo UML panaudos atvejo modelis

Sukurtam MagicDraw profiliui svarbiausia dalis yra stereotipų nustatymo modelis (26 paveikslas).

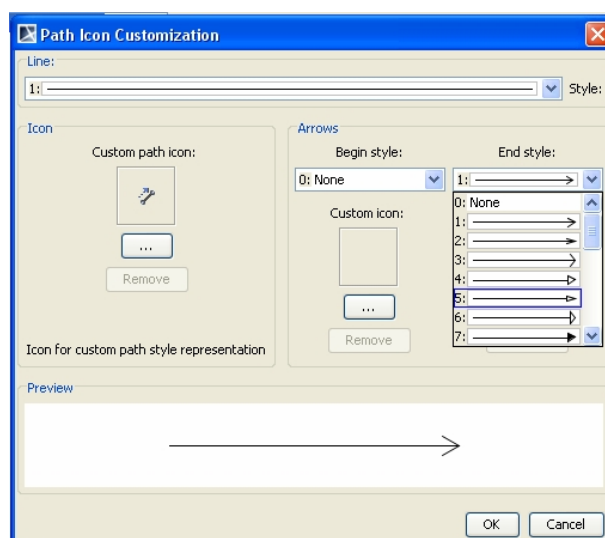


26. pav. Profilio stereotipų nustatymo modelis

12. lentelė. Stereotipų apibūdinimo lentelė

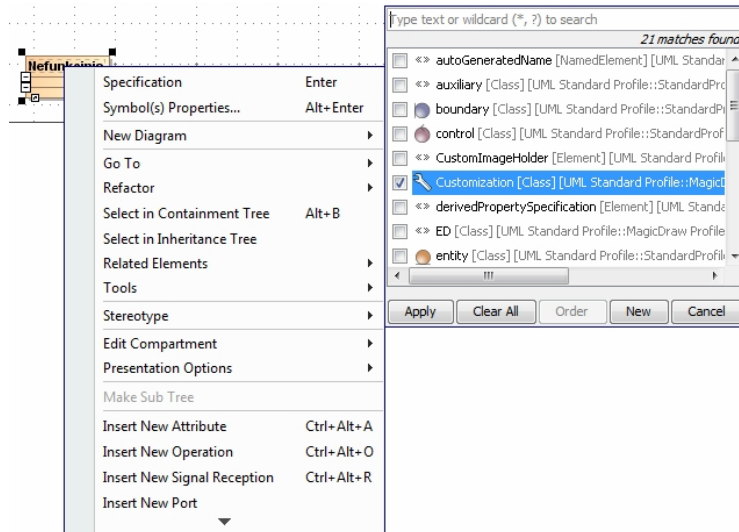
Steoretipo pavadinimas	Meta klasė	Žymių (Tag) reikšmes	Aprašymas	Žymėjimas
CCR	Class	Id, text	Besikertantis reikalavimas	
NFR	Class	Id, text	Nefunkcinis reikalavimas	
ccr	Dependency	-	Ryšys tarp elementų	
Apribojimas	Dependency	-	Ryšys tarp elementų	
Dalinis	Dependency	-	Ryšys tarp elementų	

Ryšiai kuriami pasirinkus ryšio tipą (27 pav.). Pasirenkamas linijos tipas, pradžios ir pabaigos simboliai.



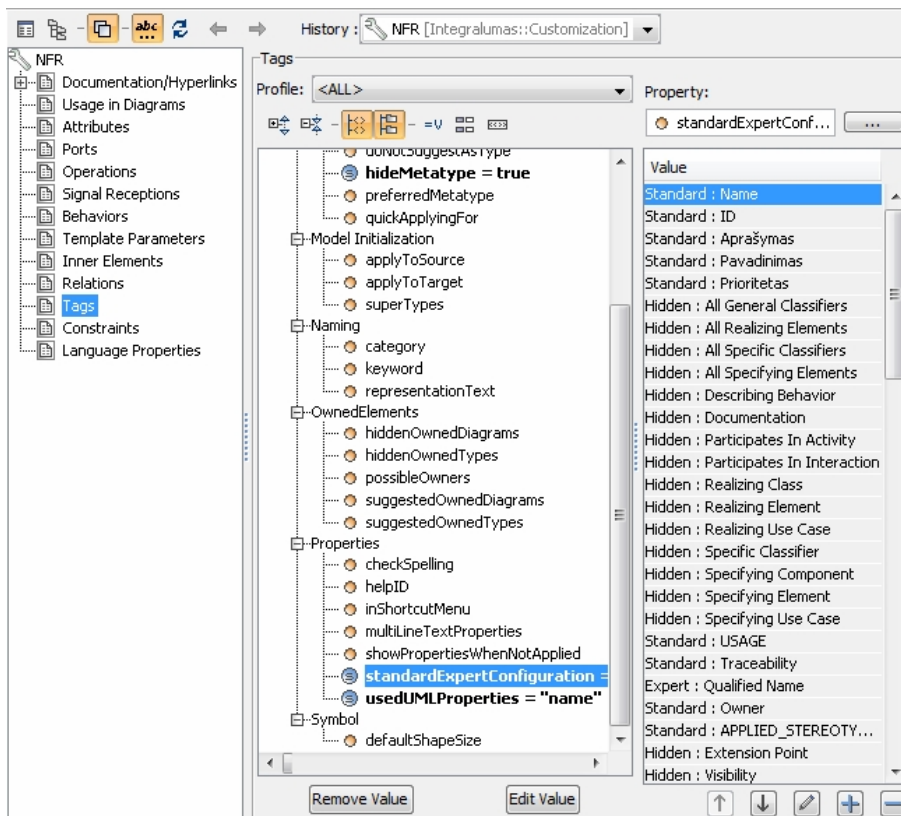
27. pav. Ryšio tipo nustatymas

Sukūrus UML klasių modelį, pradedamas profilio nustatymų kūrimas. Sukurtas naujas nustatymų klasės elementas - Nefunkcinis pateiktas (27 pav.).



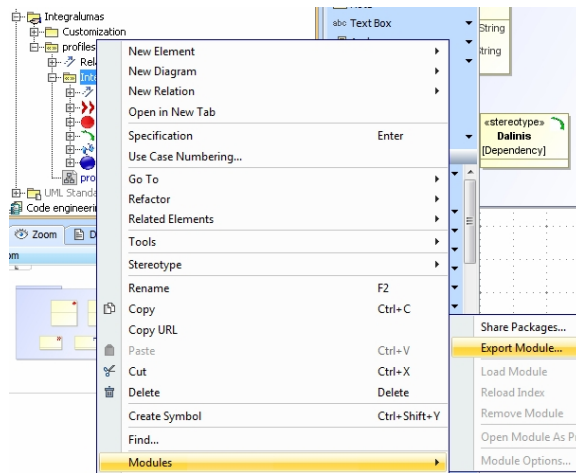
28. pav. Stereotipo „Customization“ priskyrimas

Elementui pagrindiniai nustatymai vykdomi žymių (*angl. Tags*) meniu punkte (28 paveikslas).



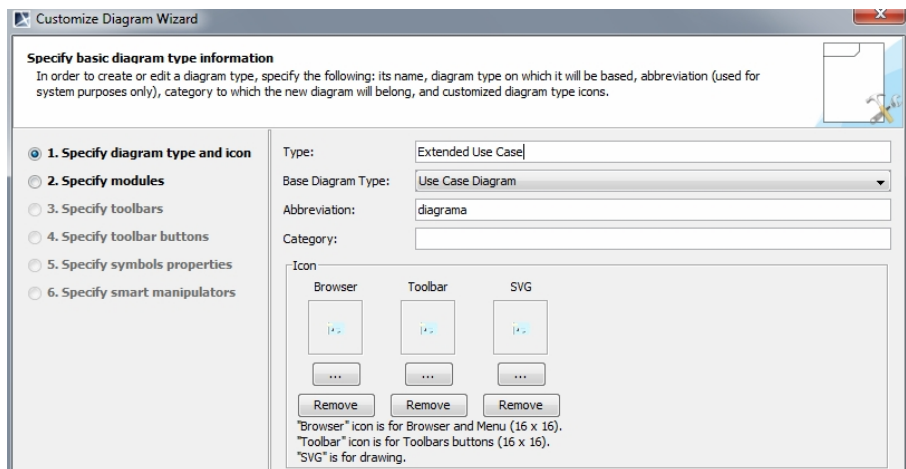
29. pav. Elemento nustatymai pagal žymes

Naujai sukurtas stereotipų modulis eksportuojamas iš meniu juostos pasirinkus „Export Module“. Paketo eksportavimas pateiktas 30 paveiksle.



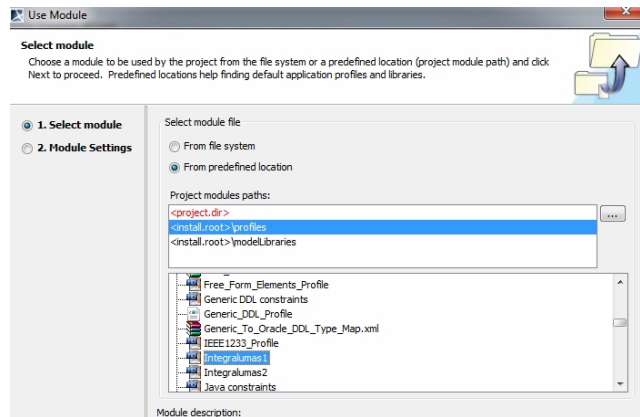
30. pav. Paketo eksportavimas

Diagramos kūrimo procesas susideda iš 6 žingsnių. Pateiktas atidarytas diagramų kūrimo vedlio langas 31 paveiksle. Vadovaujantis vedliu įvykdomi visi žingsniai. Pirmiausiai, įvedamas diagramos tipas ir pasirenkamas diagramos bazinis tipas (panaudos atvejų diagrama, klasių diagrama, sekų diagrama ir t.t.). Kuriamos diagramos pirmas žingsnis pateikiamas 31 paveiksle. Remiantis sudarytu metodu mums tinkamiausia diagrama yra Use case, kurioje galėtume atvaizduoti reikalavimų tipus (funkcinius ir nefunkcinius reikalavimus), atvaizduojant juos panaudos atvejais ir ryšius tarp šių reikalavimų. Detalesniam reikalavimų atvaizdavimui gali būti naudojamos kitos UML diagramos, pavyzdžiui, sekų, veiklos, klasių ir kt.



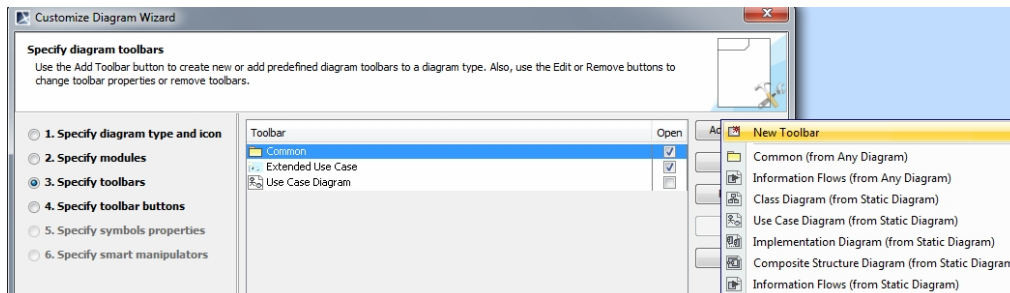
31. pav. Diagramos kūrimo langas

Sekančiame žingsnyje nurodomas profilis, pagal kurį kuriama diagrama. Paspaudus „Add“ mygtuką (32 paveikslas) parenkame ankščiau sukurtą profilį.



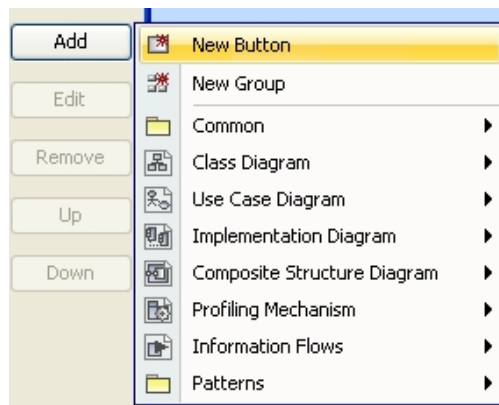
32. pav. Profilio parinkimas

Trečiame žingsnyje vykdomas įrankių juostos specifikuojimas 33 paveikslas.

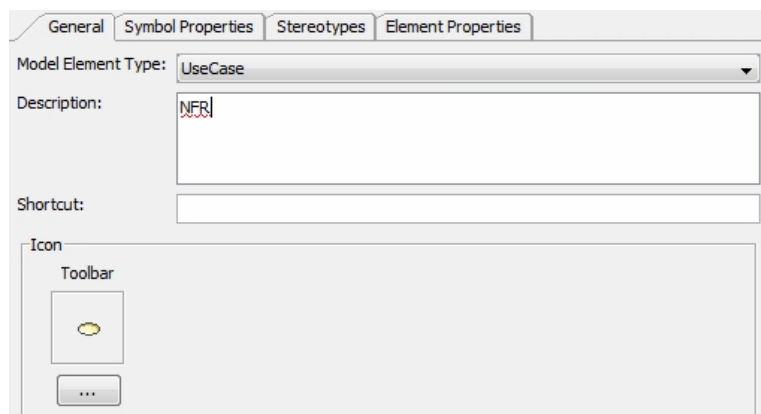


33. pav. Įrankių juostos sudarymas

Ketvirtame žingsnyje atliekamas mygtukų kūrimas. Naujas mygtukas pridedamas komanda „Add“ ir pasirinkus „New Button“ punktą 34 paveikslas. Mygtuko nustatymo lange nurodomas elemento tipas, ikona ir aprašymas (35 paveikslas).

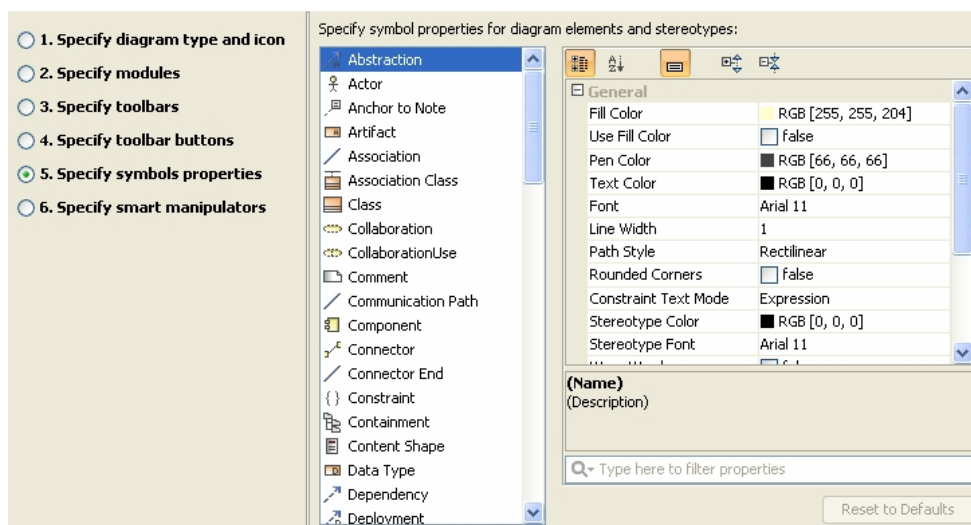


34. pav. Naujo mygtuko pridėjimas



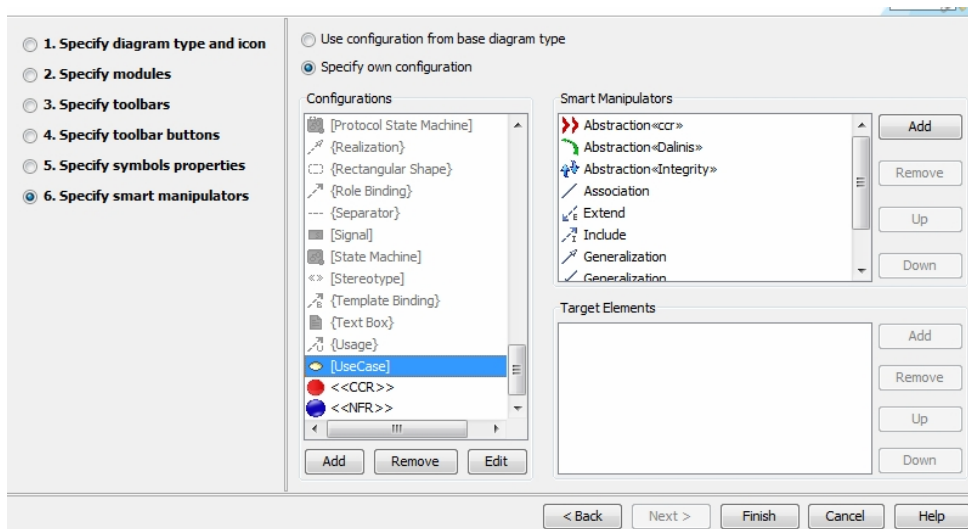
35. pav. Mygtuko kūrimo langas

Penktame žingsnyje nustatomos simbolių savybės pavaizduota 36 paveiksle.



36. pav. Simbolių savybių nustatymo langas

Paskutiniame šeštame kūrimo etape (37 paveikslas), elementams nustatomos manipuliacijos. Siekiant, pateikti greitesnį simbolių atvaizdavimą nurodomi ryšiai kai atitinkamo tipo elementas pažymėtas diagramoje. Atlikus visus prieš tai aptartus veiksmus spaudžiamas mygtukas „Finish“ ir užbaigiamas diagramos kūrimo procesas.



37. pav. Greitų manipuliacijų nustatymo langas

Įvykdžius visus veiksmus reikia išjungti ir iš naujo įjungti MagicDraw UML programą, tai atlikus galima naudoti naujai sukurta diagrama.

4.3. Metodo realizacijai pasirinktų įrankių Requisite Pro ir MagicDraw išplėtimo apibendrinimas

1. RequisitePro įrankio aplinkoje sukurtas reikalavimų specifikavimo šablonas. Į šabloną įtraukti reikalavimų tipai ir reikalavimų savybės, kad išpildytume aprašytą metodą reikalavimų integralumui užtikrinti.
2. Sukurtas šablonas su nustatytomis galimomis kontribucijomis nefunkciniams reikalavimams.
3. Į šabloną įtrauktas reikalavimų požymių atributas, nusakantis ar tai besikertantis, ar būtinas reikalavimas.
4. MagicDraw sukurtas paketo profilis grafiniam reikalavimų atvaizdavimui, remiantis sudaryta metodika, kadangi nėra nusistovėjusio standarto kaip atvaizduoti nefunkcinius reikalavimus UML.

5. Metodo realizacija pasirinktomis priemonėmis

Šiame skyriuje pateikiamas metodo realizavimas pasirinktomis ir sukurtomis priemonėmis. Reikalavimų valdymui ir specifikavimui naudojamas sukurtas šablonas RequisitePro aplinkoje. Ryšiams tarp reikalavimų atvaizduoti naudojamos atributų ir susietumo matricos, susietumo medžiai. Filtrų pagalba filtruojame informaciją pagal reikalavimų charakteristikas.

Norint pateikti kūrėjams pilną sistemos vaizdą būtinas reikalavimų atvaizdavimas. Taigi reikalavimams ir jų ryšiams atvaizduoti, mes pasirinkome UML panaudos atvejų diagramą.

5.1. Pasirinktos dalykinės srities aprašas

Projektuojama sistema valdys bankomatą (ATM), kuris turi magnetinių kortelių skaitytuvą (banko kortelės nuskaitymui), vartotojo konsolę (klaviatūra ir ekranas) (sąveikai su vartotoju), grynųjų išdavimo įrenginį (po 20 Lt kupiūromis), spausdintuvą (spausdinti čekius) ir rakinamą jungiklį, kuriuo operatorius gali įjungti arba išjungti bankomatą. ATM bendradarbiaus su banko kompiuteriu naudodamasis atitinkamu ryšio protokolu. (Bendradarbiavimo dalis neįeina į šią projektuojamą sistemą).

ATM aptarnaus tik vieną klientą tuo pačiu metu. Klientui reiks įdėti banko kortelę ir įvesti asmeninį identifikavimo numerį PIN – kurie bus abu nusiųsti validuoti į banką kiekvienos transakcijos metu. Klientas tada galės atlikti vieną ar kelias transakcijas. Banko kortelė bus laikoma bankomate tol kol klientas nuspręs, kad jis nebenori atlikti daugiau jokių transakcijų. Tuo metu kortelė bus grąžinta klientui – išskyrus kitus specialius atvejus (kai kortelė bus negražinama).

ATM teiks tokias paslaugas vartotojui:

1. Vartotojas galės išsiimti grynuosius pinigus iš bet kokios tinkamos sąskaitos, kuri yra susieta su banko kortele. Išduodama tik 20 Lt kupiūromis. Bankas turi duoti leidimą prieš išduodant pinigus.
2. Vartotojas gali pervesti bet kokią pinigų sumą tarp dviejų banko sąskaitų susietų su jo banko kortele.
3. Vartotojas gali peržiūrėti bet kurios su jo kortele susietos sąskaitos balansą.

5.1.1. Reikalavimų identifikavimo etapas

Remiantis pateiktu reikalavimų integralumą užtikrinančiu metodu, pirmas žingsnis yra reikalavimų identifikavimo etapas, kurio pasėkoje gauname identifikuotus ir klasifikuotus reikalavimus.

1. Remiantis dalykinės srities aprašu, pirmiausiai, identifikuojame suinteresuotas šalis ir nustatome individualius asmenis ar grupes, kurie padės reikalavimų išgavimo procese. Todėl suinteresuotos šalys yra asmenys, apimantys užsakovus ar klientus (kurie moka už sistemą), vartotojus (kurie sąveikauja su sistema darbe). Pavyzdžiui operatorius yra asmuo, kuris gali būti paveiktas sistemos. Kitos suinteresuotos šalys yra savininkai arba pirkėjai ir t.t..

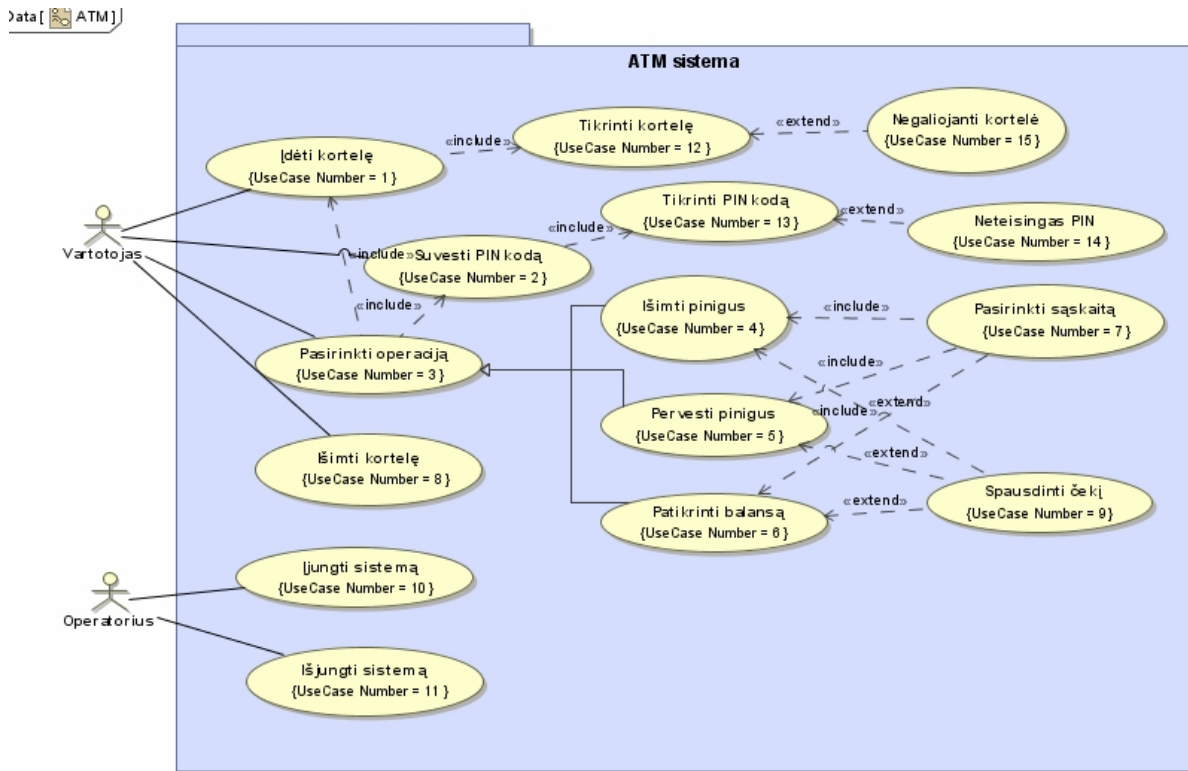
Šiame pavyzdyje atsižvelgiant į dalykinę sritį nustatėme 4 suinteresuotas šalis:

1. Klientas
2. Operatorius
3. Užsakovas
4. Pirkėjas

2. Sekantis žingsnis yra šaltinių identifikavimas. Šios užduoties tikslas sukaupti žinias apie dalykinę sritį. Renkami dokumentai, pavyzdžiui, organizacijos diagramos, proceso modeliai ar standartai, katalogai, kurie gali padėti identifikuoti reikalavimus.

3. Trečiame žingsnyje išgauname reikalavimus, kadangi jau turime identifikuotas suinteresuotas šalis bei susidarę bendrą sistemos vaizdą ankstesniuose žingsniuose. Siekiant išgauti reikalavimus, analizuojami dokumentai, kurie buvo surinkti šaltinių identifikavimo procese, įtraukiant ir suinteresuotų šalių interviu įrašus. Dokumentuose ieškomi veiksmažodžiai ir daiktavardžiai, kurie išreiškia savybes ar užduotis, kurias sistema turi teikti. Pavyzdžiui, „bankomatas turi duoti vartotojui pinigus“, reikalavimas identifikuojamas kaip „išimti pinigus“. Kiti reikalavimai kurie buvo išgauti pirmos iteracijos metu yra tokie: pervesti pinigus, pažiūrėti informaciją, prisijungti, išimti pinigus. Taip pat tekstas „ATM mašinos turi būti prieinamos visose bankuose“, gali reikšti prieinamumo reikalavimą.

Sudarome PA, kad sudarytume bendrą sistemos vaizdą, siekiant identifikuoti kitus reikalavimus. Nustatome sistemos ribas – globalinius NFR. Paveiksle 38 pateikta ATM sistemos panaudojimo atvejo diagrama.



38. pav. Sistemos vartotojai ir funkcijų tarpusavio priklausomybės

Identifikuojame globalinius NFR:

5. Sistema turi veikti 24 valandas per parą.
6. Programa turi būti parašyta C++ kalba.
7. Sistema turi atlikti visas operacijas teisingai.
8. Sistema neturi atskleisti naudingos informacijos sistemos vartotojams.
9. Sistema turi teikti paslaugas lietuvių, anglų, rusų kalbomis

Šiame žingsnyje identifikuojami ir specifikuojami panaudos atvejai, naudojant C. Larmano šabloną. Aprašyti PA sukeliama į RequisitePro aplinką. 38 paveiksle pateiktas panaudos atvejo specifikuojimo langas, pagal sukurtą šabloną.

Rekomenduojama kurti vieną specifikuojimo dokumentą, kuriame būtų pateikti visi reikalavimai pagal sudarytą šabloną. Tačiau toks sprendimas neprivalomas. Įkeliant dokumente įvestą reikalavimą į RequisitePo reikia:

1. Word dokumente pele pažymėti visą reikalavimo tekstą.
2. Spausiti Naujas reikalavimas arba MS Word meniu punktą *RequisitePro* -> *Requirements*->*New*.

3. Atveriamas reikalavimo savybių langas, skiltyje *General* nurodomi naujojo reikalavimo parametrai 39 paveikslas.

Type – pasirenkamas reikalavimo tipas iš sąrašo;

Name – įvedamas reikalavimo pavadinimas; pavadinimo vesti nebūtina, kai reikalavimo tekstas yra gana trumpas ir aiškus.

Text – čia pateikiamas įkeliamas reikalavimas;

Package – pasirenkamas katalogas, į kurį bus įkeltas reikalavimas

Location – nurodo kur yra reikalavimas (dokumente arba duomenų bazėje).

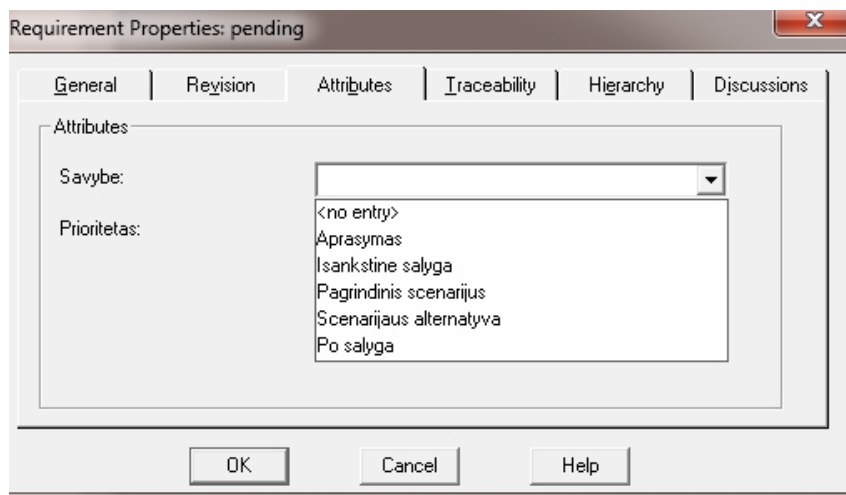
4. Reikalavimo savybių lango skiltyje *Attributes* įvedamas reikalavimo atributų reikšmės. Reikalavimo atributų reikšmės yra neprivalomos.

5. Jeigu reikia, reikalavimo savybių lango skiltyje *Hierarchy* nurodoma hierarchinė įvedamo reikalavimo priklausomybė kitiems reikalavimams. Šią priklausomybę nurodyti reikėtų, kai fiksuojamos šalutinės reikalavimų savybės, pavyzdžiui, reikalavimo tikimo kriterijus.

6. Jeigu įvedamas reikalavimas turi priklausomybių, tada savybių lango skiltyje *Traceability* nurodoma įvedamo reikalavimo priklausomybės sąsaja su kitais reikalavimais. Galimi du sąsajos būdai

From – į įvedamą reikalavimą ateinančios loginės sąsajos iš kitų reikalavimų.

To – iš įvedamo reikalavimo išeinančios loginės sąsajos į kitus reikalavimus



39. Panaudos atvejų specifikuojimas RequisitePro aplinkoje

13. lentelė. PA „Įdėti kortelę“ specifikuojimas

1. PANAUDOJIMO ATVEJIS:	Įdėti kortelę
Vartotojas/Aktorius:	Vartotojas

Aprašas:	Vartotojas norėdamas prisijungti prie sistemos turi įdėti banko kortelę
Prioritetas	Svarbus
Prieš sąlyga:	Turėti banko kortelę
Pagrindinis scenarijus:	1. Vartotojas įdeda kortelę į ATM arba kortelės skaitytuvą; 2. ATM arba kortelės skaitytuvas tikrina kortelę; 3. Vartotojas sėkmingai gali vykdyti operacijas;
Po-sąlyga:	Nėra
Alternatyvus scenarijus	Vartotojas išėmė kortelę
Susiję PA	Tikrinti kortelę, Suvesti PIN kodą

14. lentelė. PA „Suvesti PIN kodą“ specifikuojimas

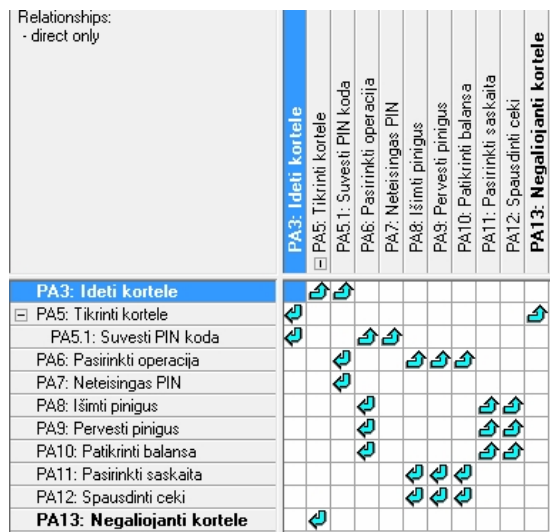
2. PANAUDOJIMO ATVEJIS:	Suvesti PIN kodą
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas turi suvesti PIN kodą norėdamas prieiti prie savo sąskaitos. Vartotojas gali būti neteisėtas, todėl sistema turi identifikuoti vartotoją
Prioritetas	Labai svarbus
Prieš sąlyga:	Vartotojas turi turėti prisijungimo slaptažodį
Pagrindinis scenarijus:	1. Vartotojas suveda PIN kodą 2. Banko duomenų bazė tikrina vartotoją 3. Vartotojas gali sėkmingai vykdyti operaciją
Po-sąlyga:	Nėra
Alternatyvus scenarijus	Jeigu slaptažodis neteisingas rodomas klaidos pranešimas „Slaptažodis neteisingas“
Susiję PA	Tikrinti kortelę

15. lentelė. PA „Tikrinti kortelę“ specifikuojimas

3. PANAUDOJIMO ATVEJIS:	Tikrinti kortelę
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Įdėta kortelė tikrinama, ar kortelė įdėta teisingai, ar kortelės galiojimo laikas nepasibaigęs.
Prioritetas	Labai svarbus
Prieš sąlyga:	Vartotojas turi turėti banko kortelę
Pagrindinis scenarijus:	1. Vartotojas įdeda kortelę į ATM arba į kortelių skaitytuvą; 2. ATM tikrina ar kortelė yra galiojanti; 3. Vartotojas turi pakartoti kortelės įdėjimą;

Po-sąlyga:	Nėra
Alternatyvus scenarijus	Jeigu kortelės galiojimas pasibaigęs, rodoma klaidos žinutė „Kortelė negalioja“
Susiję PA	Įdėti kortelę, negaliojanti kortelė

Sukelti panaudos atvejų reikalavimai į RequisitePro aplinką, atsižvelgus į šablono eilutę „Susiję Pa“ susiejame panaudos atvejus ir atvaizduojame susietumo matricą (paveikslas 40).



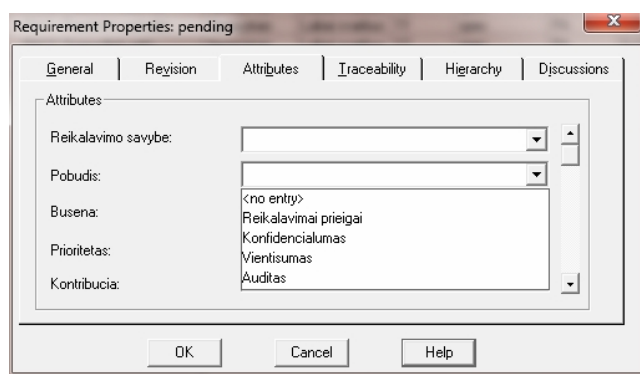
40. pav. ATM sistemos panaudos atvejų susietumo matricą

Panaudos atvejų reikalavimams sukurta atributų matrica, kurios dėka galima keisti reikalavimo atributų savybes, pavyzdžiui, prioritetą, reikalavimo savybes. Atributų matrica pateikia visą išsamią informaciją apie reikalavimą (kada reikalavimas registruotas, kokiam pakete saugomas, kokius iš ir į ryšius turi su kitais reikalavimais, fiksuojamas reikalavimų pokytis, jei jis buvo ištrintas ar koreguotas). Panaudos atvejų atributų matrica pateikta 41 paveiksle.

Requirements:	Savybė	Prioritetas	Unique I	Location	Package	Date	Reason	Traced-from	Traced-to	RootTag#
PA3: Įdėti kortelę	Aprašymas	Svarbus	3	spec	PA	5/25/2011 6	Cleared trace r	ST1	PA5, PA5.1	3
PA5: Tikrinti kortelę	Aprašymas	Labai svarbus	5	spec	PA	5/25/2011 6	Cleared trace r	FR6, FR7, FR	PA13	5
PA5.1: Suvesti PIN kodą	Aprašymas	Labai svarbus	4	spec	PA	5/25/2011 6	Cleared trace r	FR5, PA3, ST	PA6, PA7	5
PA6: Pasirinkti operacija	Aprašymas	Labai svarbus	6	spec	PA	5/25/2011 6	Cleared trace r	PA5.1, ST1(s)	PA8, PA9, PA	6
PA7: Neteisingas PIN	Aprašymas	Labai svarbus	7	spec	PA	5/25/2011 6	Cleared trace r	PA5.1		7
PA8: Išimti pinigus	Aprašymas	Labai svarbus	8	spec	PA	5/25/2011 7	Cleared trace r	FR1, FR2, FR	PA11, PA12	8
PA9: Pervesti pinigus	Aprašymas	Labai svarbus	9	spec	PA	5/25/2011 6	Cleared trace r	FR4, PA6, ST	PA11, PA12	9
PA10: Patikrinti balansą	Aprašymas	Labai svarbus	10	spec	PA	5/25/2011 7	Cleared trace r	FR4, PA6, ST	PA11, PA12	10
PA11: Pasirinkti sąskaitą	Aprašymas	Labai svarbus	11	spec	PA	5/25/2011 7	Deleted trace	FR3, PA8, PA		11
PA12: Spausdinti čekį	Aprašymas	Labai svarbus	12	spec	PA	5/25/2011 6	Cleared trace r	PA8, PA9, PA		12
PA13: Negaliojanti...	Aprašymas	Svarbus	34	Database	PA	5/25/2011 6	Cleared trace r	PA5		13
* <Click here to create a...		Svarbus	empty	Database	None	5/26/2011 9				pending

41. pav. Panaudos atvejų atributų matrica

1. Kiti reikalavimai gali būti išgauti remiantis Volere šablono nefunkcinių reikalavimų klasifikavimu. Nusprendžiame ar kiekvienas įrašas yra reikalingas mūsų sistemai. Pavyzdžiui, sistema gali naudotis keli vartotojai tuo pačiu metu, tada reikia spręsti multi – prieinamumo problemą. Sekantis reikalavimas, sistema turi reaguoti labai greitai, kad būtų aptarnauta kuo daugiau klientų. Tai reiškia užduočių vykdymo greičio reikalavimus.
2. Reikalavimų dekomponavimas. Per reikalavimų identifikavimo etapą, nustatome, kad kai kurie reikalavimai apibrėžia konkretų rezultatą, pavyzdžiui „Prisijungti“, kiti nusakomi painiais ir stambiais reikalavimais, pavyzdžiui, saugumas. Taigi aukščiausio lygio reikalavimas turi būti išskaidytas į smulkesnius. Reikalavimų skaidymas atliktas, sudarant nefunkcinių reikalavimų specifikavimo šabloną, RequisitePro aplinkoje. Kiekvienas nefunkcinis reikalavimas suskaidytas į potipius. 42 paveiksle pateiktas reikalavimo tipo „Saugumas“ potipiai.



42. pav. Reikalavimo „Saugumas“ suskaidymas į potipius

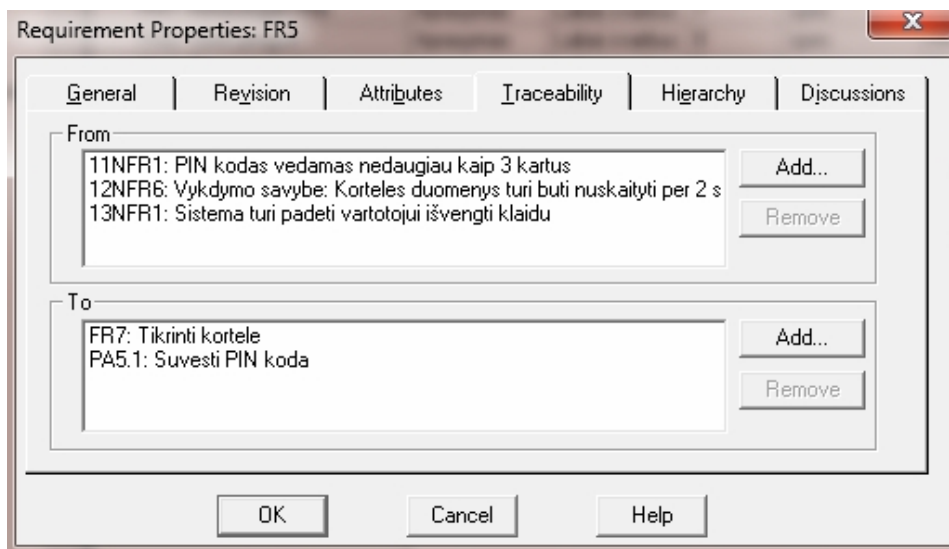
3. Klasifikavimas. Klasifikuojame reikalavimus į funkcinis ir nefunkcinius, laukelyje reikalavimo tipas. Funkciniai ir nefunkciniai reikalavimai sukelti į RequisitePro aplinką taip pat kaip ir panaudos atvejai.

5.2. Reikalavimų specifikavimo etapas

Reikalavimų specifikavimo etape, remiantis metodika reikalavimai specifikuojami detaliau.

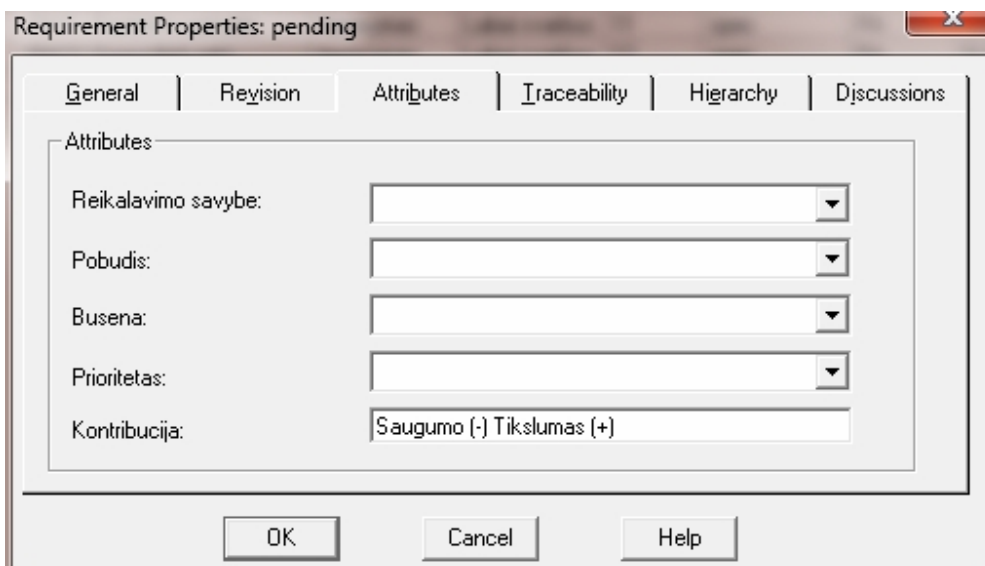
1. Identifikuojame priklausomybes. Tai kiekvieno reikalavimo paslaugų ar žinių sąrašas (pildoma priklausomybių eilutė). Priklausomybės yra paaimamos iš reikalavimo surinkimo etapo

(reikalavimų identifikavimo etapu). Kiekvienam identifikuotam reikalavimui, bent viena priklausomybė turi būti nustatyta. Pavyzdžiui, reikalavimas „Tikrinti PIN kodą“ turi priklausomybę su vienu funkcinio reikalavimu, taip pat reikalavimas susietas su panaudos atveju (žr. 43 paveikslą).



43. pav. Funkcinio reikalavimo „Tikrinti PIN kodą“ priklausomybės

2. Kontribucijų identifikavimas. Kontribucijų ryšys tarp dviejų reikalavimų nusako kaip vienas reikalavimas veikia vienas kitą. Kaip jau ir minėjome 3 skyriuje, kontribucijos gali būti teigiamos arba neigiamos. Kiekvienam reikalavimui nustatome kontribucijas ir užpildome kontribucijos eilutę. Sudarytame šablone iškarto nustatytos kontribucijos tėviniams reikalavimams. Šiuo atveju labai padeda NFR katalogas, kuris suteikia informacijos apie reikalavimus. Tačiau egzistuoja daug situacijų kur katalogas ne visada padeda. Tada reikalinga ekspertų pagalba. Kontribucijos tarp reikalavimų turėtų būti nustatomos ne tik tėviniams reikalavimams, bet ir reikalavimų potipių lygyje. Kadangi tėvinio reikalavimo vaikinis reikalavimas gali turėti skirtingą įnašą kitam reikalavimui. Pavyzdžiui, Tikslumo reikalavimas veikia neigiamai atsako laiko reikalavimą. 44 paveiksle pateiktas nefunkcinio panaudojamumo reikalavimo tipo kontribucija. Kontribucija parodo kylančią konfliktų grėsmę tarp reikalavimų tipų.



44. pav. Nefunkcinio panaudojamumo reikalavimo tipo poveikis (teigiamas ir neigiamas)

Paveiksle 45 pateikta nefunkcinių reikalavimų atributų matrica, RequisitePro įrankio filtrų pagalba atrinkti reikalavimai su kontribucijų poveikiu.

Requirements:	Reikalavimo s	Pobudis	Busena	Prioritetas	Kontribucija
12NFR1: Mašina turi reaguoti laiku, kai klientas naudojami...	Pagrindimas	Uzduociu vyk		Labai svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
12NFR2: Tikslumas: korteles duomenys turi buti nuskaityti...	Pagrindimas	Reikalavimai ti	CCR, Butinas	Labai svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
12NFR3: Tikslumo: ATM turi duoti klientui tikslia pinig...	Pagrindimas	Reikalavimai ti	CCR, Butinas	Labai svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
12NFR4: Vykdyimo savybes: Sistema turi išduoti...	Pagrindimas	Uzduociu vyk	CCR, Butinas	Svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
12NFR5: Tik pilnai įvykdytos operacijos saugomos.	Pagrindimas	Patikimumas ir	CCR, Butinas	Labai svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
12NFR6: Vykdyimo savybe: Korteles duomenys turi.	Pagrindimas	Uzduociu vyk	CCR, Butinas	Labai svarbus	Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)
* <Click here to create a requirement>					Reikalavimai panaudojamumui (+) saugumo reikalavimai (-)

45. pav. nefunkcinių reikalavimų atributų matrica filtruota pagal kontribucijų įtaką

3. Nustatome būtinus reikalavimus. Jei vienam reikalavimui reikalingas kitas reikalavimas, kad jis užbaigtų savo elgesį arba jis yra to reikalavimo apribojimas, tai jis yra būtinas reikalavimas. Ši informacija šablone pildoma Būtinų reikalavimų eilutėje. Pavyzdžiui, Prisijungti reikalavimui būtinas atsako laiko reikalavimas, nes sistema turi reaguoti laiku, kai klientas įdeda kortelę į bankomatą.

4. Suinteresuotų šalių prioritetai. Šiame žingsnyje identifikuojame kiekvieno reikalavimo svarbą sistemoje, kiekvienos suinteresuotos šalies atžvilgiu ir pildome šablone suinteresuotų šalių prioritetų eilutėje.

Įvykdžius visus žingsnius gauname užpildytą šabloną su identifikuotais ir specifikuotais reikalavimais. 16 lentelėje pateiktas reikalavimo „Prisijungti prie sistemos“ užpildytas šablonas.

16. lentelė. Funkcinio reikalavimo „Prisijungti prie sistemos“ detalizavimas

Reikalavimas	Klasifikavimas	PA
2	FR	1
Aprašymas	Prisijungimas prie sistemos	
Pagrindimas	Sistema turi leisti vartotojui prisijungti prie sistemos. Vartotojas jungiasi prie ATM naudojant kortelę.	
Šaltinis	Užsakovas	
Tikimo kriterijus	Galiojanti kortelė	
Priklausomybės	Konfliktai	
Kortelės duomenų tikrinimas Kortelės gražinimas klientui	Nėra	
Papildoma medžiaga		
Istorija		
Prioritetų sąrašas		
Klientas Operatorius Užsakovas Pirkėjas	Labai svarbus Labai svarbus Labai svarbus Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Tikslumas Tikrinti kortelę	

17 lentelėje pateiktas atsako laiko reikalavimo specifikacija. Pastaba, kad kontribucijų ryšys tarp užduočių vykdymo greičio reikalavimo ir kitų reikalavimų nustatytas, naudojant katalogą [6, 30]. Šiuo atveju tikslumo reikalavimas veikia neigiamai užduočių vykdymo greičio reikalavimą. Ši informacija įrašoma į kontribucijų eilutę.

17. lentelė. Nefunkcinio reikalavimo „Užduočių vykdymo greitis“ detalizavimas

Reikalavimas	Klasifikavimas	PA
--------------	----------------	----

3	12	
Aprašymas	Užduočių vykdymo greitis	
Pagrindimas	Vykdymo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės	
Šaltinis	Užsakovas	
Tikimo kriterijus	Klientas turi įdėti kortelę į ATM	
Priklausomybės	Konfliktai	
Reaguoti per 2< sekundės kai klientas įdeda kortelę Reaguoti laiku per 2< sekundės, kai klientui gražinama kortelė	Nėra	
Papildoma medžiaga		
Istorija		
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Tikslumas (-) Tikslumas: kortelės duomenys turi būti nuskaityti 100 %	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

18. lentelė. Funkcinio reikalavimo „Kortelės tikrinimas“ detalizavimas

Reikalavimas	Klasifikavimas	PA
1	FR	2
Aprašymas	Tikrinti kortelę	
Pagrindimas	Kortelė turi būti autentifikuota, kad vartotojas galėtų naudotis sistema	
Šaltinis	Informacijos šaltinis	
Tikimo kriterijus	Tikimo kriterijus	

Priklausomybės	Konfliktai
Kortelės duomenų nuskaitymas Kortelės galiojimo datos patikrinimas	Nėra
Papildoma medžiaga	
Istorija	
Prioritetų sąrašas	
Klientas	Labai svarbus
Operatorius	Svarbus
Užsakovas	Labai svarbus
Pirkėjas	Labai svarbus
Kontribucijų sąrašas	
Kontribucija	Nėra
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu

Kiti sistemos reikalavimai pateikti antrame priede.

Reikalavimai identifikuojami parametrais: numeriu, tipu ir panaudojimo atveju numeriu. Norint užtikrinti reikalavimų integralumą, ryšiams tarp skirtingų reikalavimų tipo atvaizduoti naudojamos susietumo matricos. (Traceability Matrix) – tai ryšius tarp skirtingo ar to paties tipo reikalavimų vaizduojanti lentelė. Ryšiai vaizduojami susikirtimuose tarp eilutės ir stulpelio reikalavimų. 46 paveiksle pateikta funkcinių reikalavimų ir panaudos atvejų sąsajų matrica.

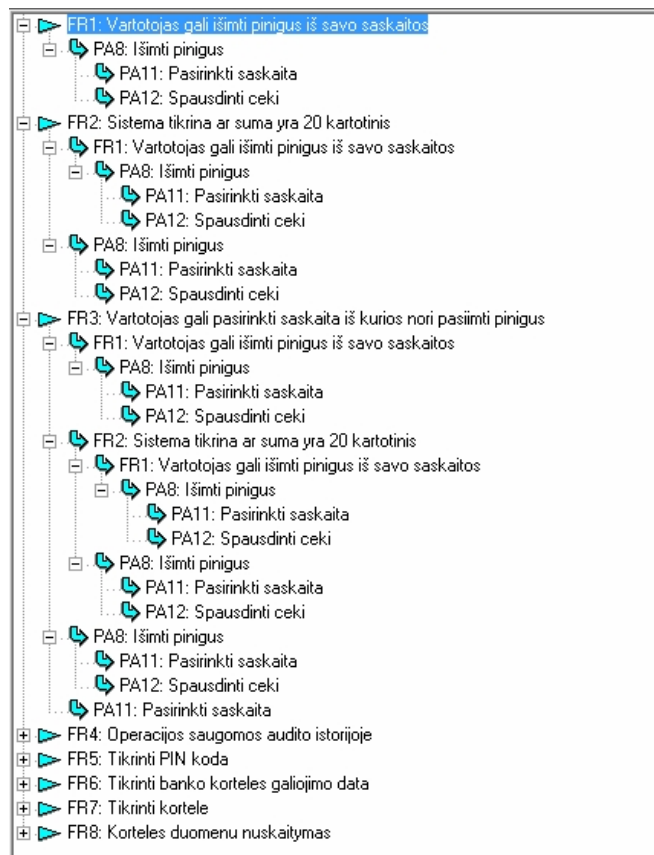
Relationships: - direct only	PA3: Idėti kortelė	PA5: Tikrinti kortelė	PA51: Suvesti PIN kodą	PA6: Pasirinkti operacija	PA7: Netaisyti PIN	PA8: Išimti pinigus	PA9: Pervesti pinigus	PA10: Patikrinti balansą	PA11: Pasirinkti sąskaita	PA12: Spausdinti čekį	PA13: Negaliojanti...
FR1: Vartotojas gali išimti pinigus iš savo sąskaitos											
FR2: Sistema tikrina ar suma yra 20 kartotinis											
FR3: Vartotojas gali pasirinkti sąskaita iš kurios nori pasiimti pinigus											
FR4: Operacijos saugomos audito istorijoje											
FR5: Tikrinti PIN kodą											
FR6: Tikrinti banko kortelės galiojimo data											
FR7: Tikrinti kortelė											
FR8: Kortelės duomenų nuskaitymas											

46. pav. Funkcinių reikalavimų ir panaudos atvejų sąsajų matrica

Norint atvaizduoti visas reikalavimų sąsajas tarp skirtingų reikalavimų tipų, naudojami into ir out medžiai. Paveiksluose 47 ir 48 pateikti into ir out medžiai

- ▶ FR1: Vartotojas gali išimti pinigus iš savo sąskaitos
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu
 - 🔗 12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas
 - 🔗 **12NFR4: Vykdyimo savybės: Sistema turi išduoti pinigus per 5 sekundes klientui**
 - 🔗 **12NFR5: Tik pilnai įvykdytos operacijos saugomos audito istorijoje**
- FR2: Sistema tikrina ar suma yra 20 kartotinis
 - 🔗 12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas
 - FR3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu
 - FR3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu
- ▶ FR2: Sistema tikrina ar suma yra 20 kartotinis
 - 🔗 12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas
 - FR3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu
- FR3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu
- FR4: Operacijos saugomos audito istorijoje
 - 🔗 **12NFR5: Tik pilnai įvykdytos operacijos saugomos audito istorijoje**
- ▶ FR5: Tikrinti PIN kodą
 - 🔗 11NFR1: PIN kodas vedamas ne daugiau kaip 3 kartus
 - 🔗 **12NFR6: Vykdyimo savybė: Korteles duomenys turi būti nuskaityti per 2 s**
 - 🔗 13NFR1: Sistema turi padėti vartotojui išvengti klaidų
- FR6: Tikrinti banko korteleles galiojimo datą
 - 🔗 **12NFR6: Vykdyimo savybė: Korteles duomenys turi būti nuskaityti per 2 s**
- ▶ FR7: Tikrinti kortelę
 - 🔗 **12NFR6: Vykdyimo savybė: Korteles duomenys turi būti nuskaityti per 2 s**
 - FR5: Tikrinti PIN kodą
 - 🔗 11NFR1: PIN kodas vedamas ne daugiau kaip 3 kartus
 - 🔗 **12NFR6: Vykdyimo savybė: Korteles duomenys turi būti nuskaityti per 2 s**
 - 🔗 13NFR1: Sistema turi padėti vartotojui išvengti klaidų
 - FR6: Tikrinti banko korteleles galiojimo datą
 - 🔗 **12NFR6: Vykdyimo savybė: Korteles duomenys turi būti nuskaityti per 2 s**
- ▶ FR8: Korteles duomenų nuskaitymas
 - 🔗 12NFR2: Tikslumas: korteleles duomenys turi būti nuskaityti 100 % tikslumu

47. pav. Reikalavimų sąsajų into medis



48. pav. Reikalavimų out medis

Tokių reikalavimų susiejimas užtikrina reikalavimų integralumą specifikacijoje, reikalavimai nėra išsibarstę po visą dokumentą, nes kiekvienas reikalavimas specifikuojamas vienoje formoje. Svarbiausias žingsnis reikalavimų integralumui užtikrinti tinkamas reikalavimų specifikavimas ir tikslus reikalavimų priklausomybių ir būtinų reikalavimų nustatymas.

5.3. Reikalavimų komponavimo etapas

Šios užduoties tikslas yra dvejopas: gauti platesnį sistemos vaizdą ir valdyti konfliktines situacijas. Konfliktų identifikavimas ankstyvoje projekto stadijoje padeda deryboms su interesuotomis šalimis ir išvengti nesklaidumų prieš architektūrinę sistemos plėtojimą.

1. Identifikuojame sąveikos taškus. Sąveikos taškus identifikuojame sudarydami matricą iš reikalavimų ir būtinų reikalavimų. Matrica pateikta 3 skyriuje. Jei prieš tai etapai buvo pilnai įvykdyti, sąveikos taškus galime nustatyti iš užpildyto šablono atsižvelgdami į reikalavimo pavadinimo ir būtinų reikalavimų eilutes. Sąveikos taškų matrica sudaroma rankiniu būdu, kadangi RequisitePro įrankis neleidžia atvaizduoti kelių tipų atributų matricos.

19. lentelė. Sąveikos taškų lentelė

Būtinai reikalavimai	9FR7: Tikrinti kortelę	12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas	12NFR4: Vykdyimo savybės: Sistema turi išduoti pinigus per 5 s	12NFR5 Tik pilnai įvykdytos operacijos saugomos audito istorijoje	11NFR1: Saugumo: PIN vedamas 3 kartus	12NFR2: Tikslumas: kortelės duomenys turi būti nuskaityti 100 %	12NFR6: Vykdyimo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės	12NFR7: Įvykus klaidai pranešimai pateikiami per 2s	Sąveikos taškas
9FR1: Vartotojas gali išimti pinigus iš savo sąskaitos	✓	✓	✓	✓		✓		✓	MP1R1
9FR2: Sistema tikrina ar suma yra 20 kartotinis	✓					✓		✓	MP2R2
9FR3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus	✓					✓		✓	MP3R3
9FR4: Operacijos saugomos audito istorijoje				✓				✓	MP4R4
9FR5: Tikrinti PIN kodą	✓				✓	✓		✓	MP5R5
9FR6: Tikrinti banko kortelės galiojimo datą	✓					✓	✓	✓	MP6R6
9FR7: Tikrinti kortelę						✓	✓	✓	MP6R7
9FR8: Kortelės duomenų nuskaitymas						✓	✓	✓	MP8R8

Sudarius tokią matricą, pastebėta, kad kai kurių reikalavimų sąveika kitų reikalavimų atžvilgiu buvo praleista, nepastebėta pildant reikalavimų šablonus. Taigi matrica padeda išvengti klaidų, nes visi reikalavimai matomi vienoje formoje.

2. Identifikuojame besikertančius reikalavimus. Kaip jau buvo minėta 3 skyriuje, besikertantys reikalavimai yra tokie, kurie yra būtinai dvejimams ar keliems reikalavimams. Naudojantis sudaryta matricą, besikertančius reikalavimus yra lengva nustatyti atkreipus dėmesį į būtinų reikalavimų stulpelį vieno reikalavimo atžvilgiu, jei būtinas reikalavimas būtinas daugiau nei vieną kartą, tai jis yra besikertantis. Jei stulpelyje žymėjimas (V) pasikartoja daugiau nei vieną kartą, toks reikalavimas yra besikertantis. Toks reikalavimas RequisitePro įrankyje, atributas požymis

reikalavimų savybių lange nustatomas kaip besikertantis reikalavimas (CCR). Kartais ši savybė nustatoma iš karto, kai reikalavimas yra įkeliamas į RequisitePro įrankį, bet tik tuo atveju jei jau tiksliai žinoma, kad šis reikalavimas bus reikalingas kitiems reikalavimams. Taigi reikalavimas „Tikrinti kortelę“ turi du požymius, jis apibūdinamas kaip besikertantis ir kaip būtinas reikalavimas. Iš sudarytos matricos nustatyti šie besikertantys reikalavimai:

9FR7: Tikrinti kortelę

12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas

12NFR4: Vykdyto savybės: Sistema turi išduoti pinigus per 5 s

12NFR5: Tik pilnai įvykdytos operacijos saugomos audito istorijoje

12NFR2: Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu

12NFR6: Vykdyto savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės

Naudojant RequisitePro filtrų pagalbą, atrinkti nefunkciniai reikalavimai, kurie yra besikertantys arba būtini reikalavimai. Tokia matrica, kuri atvaizduoja nefunkcinius besikertančius ir būtinus reikalavimus vienoje vietoje, patogi tuo, kad šių reikalavimų nereikia ieškoti visame specifikacijos dokumente ir išvengiama reikalavimų pasikartojimo.

49 paveiksle pateikta atributų matrica su nustatytais besikertančiais ir būtinais reikalavimais.

Requirements:	Reikalavimo s	Pobudis	Pozymis
▶ 12NFR2: Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu	Pagrindimas	Reikalavimai ti	CCR, Butinas
▶ 12NFR3: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas	Pagrindimas	Reikalavimai ti	CCR, Butinas
▶ 12NFR4: Vykdyto savybės: Sistema turi išduoti pinigus per 5 sekundes...	Pagrindimas	Uzduociu vyk	CCR, Butinas
▶ 12NFR5: Tik pilnai įvykdytos operacijos saugomos audito istorijoje	Pagrindimas	Patikimumas ir	CCR, Butinas
▶ 12NFR6: Vykdyto savybė: Kortelės duomenys turi būti nuskaityti per 2 s	Pagrindimas	Uzduociu vyk	CCR, Butinas

49. pav. Nefunkciniai besikertantys ir būtini reikalavimai

Tokiu pat principu atrenkami ir funkciniai besikertantys reikalavimai. Pagal tiriamą dalykinę sritį buvo identifikuotas vienas funkcinis besikertantis reikalavimas 50 paveiksle.

Requirements:	Reikalavimo s	Pozymis	Prioritetas
▶ 9FR7: Tikrinti kortelę	Pagrindimas	CCR, Butinas reikalavimas	Labai Svarbus
* <Click here to create a requirement>			Svarbus

50. pav. Funkcinis besikertantis ir būtinas reikalavimas

3. Konfliktų valdymas. Kiekvienam sąveikos taškui identifikuojame įnašas tarp sukomponuotų reikalavimų. Šiuo atveju „Vykdyto savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės“ ir „Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu“ sąveikos taške MPR8, pagrindinio reikalavimo „Kortelės duomenų nuskaitymas“ atžvilgiu. Šie reikalavimai

veikia neigiamai vienas kitą (žr. kontribucijų eilutę). Priklausomai tuo prioritetų galima nustatyti konfliktus.

Šiuo atveju „Kortelės duomenys turi būti nuskaityti per 2 sekundės“ ir „Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu“ turi tą patį prioritetą Kliento, Operatoriaus, Užsakovo, Pirkėjo (žr. 4.7 lentelė (**Prioritetų sąrašo eilutę**)) atžvilgiu, taigi reikalingos derybos, kad reikalavimai būtų reitinguoti. Norint išspręsti konfliktus tarp reikalavimų vienam iš reikalavimų turėtų būti sumažintas prioritetas. Pavyzdžiui, visos suinteresuotos šalys sutinka sumažinti prioritetą „Kortelės duomenys turi būti nuskaityti per 2 sekundės“ reikalavimui, tada „Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu“ tampa dominuojančiu reikalavimu „Kortelės duomenų nuskaitymas“ sąveikos taške. Taigi konfliktas potencialiai išspręstas. Galima pasirinkti architektūrinį sprendimą, kad pirmiausia turėtų būti išpildytas dominuojantis reikalavimas.

4. Kompozicijos taisyklės. Naudojant informaciją surinktą per pastaruosius etapus, kiekvienam sąveikos taškui apibrėžiamos kompozicijos taisyklės. Taisyklės nurodo eiliškumą, kaip kiekvienas reikalavimas turi būti išpildytas. Pavyzdžiui, kompozicijos taisyklės sąveikos taškui MPR1, kuris suriša „Vartotojas gali išimti pinigus iš savo sąskaitos“ reikalavimą su visais būtinais reikalavimais yra:

((R7: Tikrinti kortelę >> ((R1: Vartotojas gali išimti pinigus iš savo sąskaitos || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 %) || R14: Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas >> R16: Vykdyto savybės: Sistema turi išduoti pinigus per 5 s)) >> R17: Tik pilnai įvykdytos operacijos saugomos audito istorijoje) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.**

Sąveikos taškui MPR2 (R2: Sistema tikrina ar suma yra 20 kartotinis) taisyklė:

((R7: Tikrinti kortelę >> R2: Sistema tikrina ar suma yra 20 kartotinis) || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 %) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.**

Sąveikos taškui MPR3 (R3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus) taisyklė:

((R7: Tikrinti kortelę >> R3: Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus) || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 %) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.

Sąveikos taškui MPR4 (R4: Operacijos saugomos audito istorijoje) taisyklė:

(R17: Tik pilnai įvykdytos operacijos saugomos audito istorijoje >> R4: Operacijos saugomos audito istorijoje) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.

Sąveikos taškui MPR5 (R5: Tikrinti PIN kodą) taisyklė:

((R7: Tikrinti kortelę >> (R5: Tikrinti PIN kodą || R9: Saugumo: PIN vedamas 3 kartus) || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 %)[>R18: Įvykus klaidai pranešimai pateikiami per 2s.

Sąveikos taškui MPR6 (R6: Tikrinti banko kortelės galiojimo datą):

((R7: Tikrinti kortelę >> R6: Tikrinti banko kortelės galiojimo datą) || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 % || R17: Vykdyimo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.

Sąveikos taškui MPR7 (R7: Tikrinti kortelę) taisyklė:

(R7: Tikrinti kortelę || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 % R17: Vykdyimo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.

Sąveikos taškui MPR8 (R8: Kortelės duomenų nuskaitymas) taisyklė:

(R7: Tikrinti kortelę >> R8: Kortelės duomenų nuskaitymas) || R13: Tikslumas: kortelės duomenys turi būti nuskaityti 100 % || R17: Vykdyimo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės) [>R18: Įvykus klaidai pranešimai pateikiami per 2s.

5.4. Reikalavimų rinkinių atvaizdavimas UML


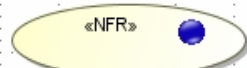
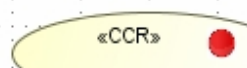


Norint pateikti kūrėjams pilną sistemos vaizdą būtinas reikalavimų atvaizdavimas. Taigi reikalavimams ir jų ryšiams atvaizduoti, pasirinkta UML panaudos atvejų diagrama.


Nors šis metodas akcentuoja pagrindines funkcijas, kurias sistema turėtų vykdyti ateityje, taip pat apibrėžia apribojimus ir tam tikras savybes, kurios daro įtaką visai sistemai ar jos dalims.

Remiantis informacija pateikta šablone, sudaryta išplėsta PA diagrama. Prieš pradėdant kurti diagramą ir naudojant naujus elementus visų pirma vartotojas turi gerai suprasti kam ir kodėl naudojami nauji sukurti elementai, kurie atspindi naujos sukurtos metodikos specifikavimo artefaktus.

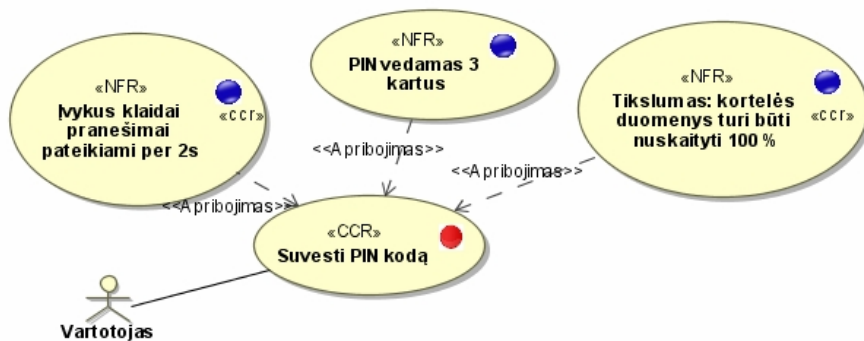
Pavyzdžiui, panaudojimo atvejų diagrama sudaroma pagal šabloną, naudojant artefaktus pateiktus 20 lentelėje. Prieš taikant šias taisykles, reikia išplėsti būtinų reikalavimų ryšį (Šablone - būtini reikalavimai), ryšiams tarp panaudojimo atvejų. Todėl, šie ryšiai priskirti <<include>> ryšiui ir <<extend>> ryšiui. Įtraukti ryšiai atskleidžia panaudojimo atvejų priklausomybes ir žymi kitų reikalavimų įtraukimą. Be to, ryšių išplėtimas priklauso nuo panaudos atvejų priklausomybių, kuris išplečia panaudos atvejį. Funkcinių ir nefuncinių reikalavimų asociacijoms atvaizduoti sukurtas naujas stereotipas <<apribojimas>>. Jei reikalavimas skaidomas į smulkesnius reikalavimus, bet yra neatskiriami nuo tėvinio reikalavimo, tai naudojamas <<dalinis>> ryšys.

20. lentelė. Gairės, PA modeliavimui pagal sudarytą reikalavimų šabloną

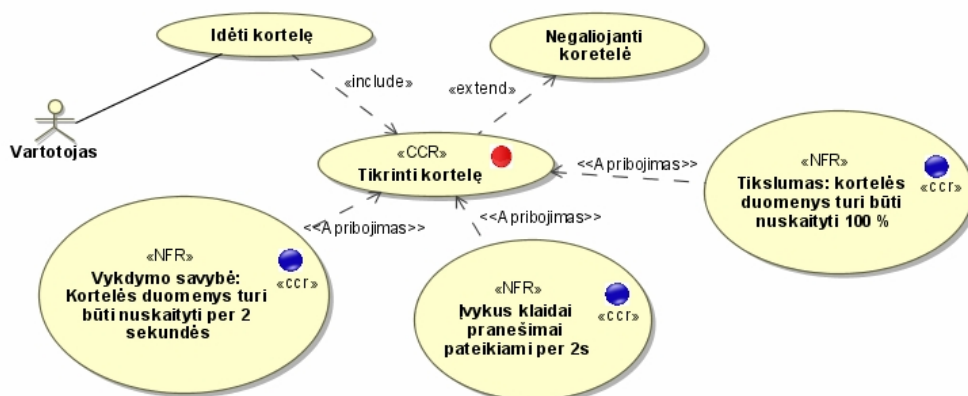
Šablonas	Panaudos atvejų modelis	Žymėjimas
Jei R - funkcinis reikalavimas	Jis atvaizduojamas panaudos atveju.	
Jei R - nefuncinis reikalavimas	Jis atvaizduojamas išplėstu panaudos atveju su prefiksu NFR.	
Jei R - besikertantis reikalavimas	Jis atvaizduojamas išplėstu panaudos atveju su prefiksu CCR.	
Jei būtini reikalavimai turi ryšį T tarp reikalavimų R1 ir R2	Sukuriamas ryšys tarp reikalavimų atitinkantis R1 ir R2. T žymimas <<apriboja>>, <<extend>>, <<apribojimas>> ryšiais.	
Jei R - reikalavimas skaidomas į r1... rn	Sukuriamas <<dalinis>> ryšys tarp panaudos atvejų, atvaizduojantis R	

	ir r1...rn.	
Kiekvienam aktoriui S, sukuriamas naujas panaudos atvejis	Sukuriamas aktorius S ir asociacija tarp jo ir panaudos atvejo.	

Todėl siūloma, vietoj to, kad kurti panaudos atvejų modelį atvaizduojantį visą sistemą, kuriami skirtingi panaudojimo atvejo vaizdai, kur kiekvienas panaudos atvejis atvaizduoja sistemos vaizdą iš tam tikro vieno reikalavimo perspektyvos. 51 ir 52 paveiksluose pateikta panaudos atvejų diagramos vieno reikalavimo atžvilgiu.



51. pav. Panaudos atvejo diagrama „Suvesti PIN kodą“ reikalavimui



5.5. Realizacijos išvados

1. Sukurtos metodikos realizavimui buvo pasirinkta konkreti dalykinė sritis. Remiantis srities problematika, buvo surinkti reikalavimai. Panaudojus reikalavimų specifikavimo proceso metodiką, realizavimo metu buvo sėkmingai išpildyti visi metodikos etapai. Reikalavimų valdymui ir sekimui naudojamas RequisitePro įrankis, o grafiniam reikalavimų atvaizdavimui naudojama UML panaudos atvejų diagrama.
2. Reikalavimų valdymui ir specifikavimui naudojamas sukurtas šablonas RequisitePro aplinkoje. Ryšiams tarp reikalavimų atvaizduoti naudojamos atributų ir susietumo matricos, susietumo medžiai.
3. Tarp reikalavimų buvo rasta konfliktuojančių reikalavimų pora ir sėkmingai išspręstas, sumažinus vienam iš reikalavimų prioritetą, tokiu atveju kitas reikalavimas tapo dominuojančiu reikalavimu.
4. Filtrų pagalba sėkmingai buvo atfiltruoti funkciniai ir nefunkciniai besikertantys ir būtini reikalavimai, kurie buvo atvaizduoti atributų matricoje, vienoje formoje.
5. Naujų požymių įtraukimas į šabloną tokių kaip „Būtinasis reikalavimas“, „Kontribucija“ užtikrina nagrinėjamo reikalavimo integralumą, o tinkamas reikalavimų specifikavimas naudojantis sudarytą metodu ir nustatant unikalius identifikatorius, tipus, priklausomybes, susijusius PA, užtikrina visos specifikacijos integralumą.
8. Reikalavimų specifikavimas vienoje formoje užtikrina reikalavimo integralumą, reikalavimą specifikuojant dvejomis perspektyvomis apimant ne tik kaip reikalavimas sąveikauja su kitais reikalavimais, bet ir kas sąveikauja specifikuojamą reikalavimą.
6. Nors ir visi metodikos etapai buvo įgyvendinti sėkmingai, tačiau didelis darbas tenka analitikui, pasitelkiant logiką ir siūlomus standartus, tokių reikalavimų savybių nustatymui, kaip sąveikos taškų, įnašų, konfliktų.

6. Reikalavimų integralumo metodikos eksperimentinis tyrimas

Eksperimentinis tyrimas atliktas remiantis 5 skyriuje atlikta realizacija, taikant metodą ATM sistemos reikalavimų specifikuojimui. Reikalavimai buvo specifikuojami naudojant sudarytą šabloną funkciniais ir nefunkciniais reikalavimams, naudojant reikalavimų integralumą užtikrinantį procesą. Atsižvelgiant į dalykinę sritį vadovaujantis metodikos reikalavimų identifikavimo etapo procesu, nustatyti ir surinkti sistemos funkciniai, nefunkciniai reikalavimai, PA bei suinteresuotos šalys.

Iš viso buvo specifikuojami 11 panaudos atvejai, nustatytos keturios suinteresuotos šalys, 8 funkciniai reikalavimai ir 12 nefunkcinių reikalavimų.

Vadovaujantis metodikos antro etapo reikalavimų specifikuojimo procesu nustatytos suinteresuotųjų šalių reikalavimų prioritetai. Šiame etape nustatytos reikalavimų priklausomybės. Funkciniams reikalavimams nustatytos 6 priklausomybės. Priklausomybės tarp PA nustatytos remiantis PA šablonu, atsižvelgiant į susijusių PA eilutę, nustatyta 14 priklausomybių tarp PA. Nustatytos 11 sąsajų tarp funkcinų reikalavimų ir panaudos atvejų ir 15 sąsajų tarp funkcinų ir nefunkcinių reikalavimų. Sudarius sąveikos taškų matricą nustatyti 8 sąveikos taškai, bei 8 reikalavimai, kurie atliko pagrindinio reikalavimo vaidmenį.

Atliekant eksperimentą nustatyti 8 būtini reikalavimai, 7 nefunkciniai ir 1 funkcinis reikalavimas. Remiantis sudaryta matrica identifikuoti 5 besikertantys reikalavimai, vienas iš jų yra funkcinis reikalavimas likusieji nefunkciniai reikalavimai.

Vadovaujantis sąveikos taškais ir nustatytomis kontribucijomis nustatyta galimų konfliktuojančių reikalavimų pora. Kitu atveju, didelė tikimybė neaptikti konfliktuojančių reikalavimų, kurie gali iškilti tik projektavimo etape. Konfliktas nustatytas dėl išlanksto nustatytų kontribucijų ir sudarytų sąveikos taškų. Kiekvienam sąveikos taškui sudarytos kompozicijos taisyklės.

Naudojantis sudarytu šablonu nustatomos nagrinėjamo reikalavimo priklausomybės, o nustatant būtinus reikalavimus nustatomi apribojimai nagrinėjamam reikalavimui. Naudojantis metodu reikalavimas specifikuojamas dvejomis perspektyvomis, ne tik kam specifikuojamas reikalavimas daro įtaka (nusakomi priklausomybės), bet ir kas jam daro įtaka (Būtinai reikalavimai). Tokiu specifikuojimo būdu užtikrinamas reikalavimų integralumas.

6.1. Eksperimento rezultatų apibendrinimas

Atlikus eksperimentinį tyrimą ir atlikus reikalavimų specifیکavimą, ATM sistemos paaiškėjo, kad vadovaujantis tokia metodika, didelė atsakomybė tenka analitikui, nes nuo analitiko kompetencijos ir patirties priklauso ar visi konfliktai tarp reikalavimų bus aptikti ir išspręsti reikalavimų specifیکavimo etape, ar bus identifikuoti būtinų reikalavimų rinkiniai ir besikertantys reikalavimai ar teisingai nustatytos reikalavimų priklausomybės. 21 lentelėje pavaizduotas analizės dalyje nagrinėtų metodų palyginimas su sukurtu metodu pagal atrinktus kriterijus, kurie padeda užtikrinti reikalavimų integralumą išlaikant reikalavimų detalumą.

21. lentelė. Metodų palyginimo lentelė

Palyginimo kriterijai	Volere	AORA (A. Moreira)	L. Chungo metodas	Sukurtas metodas
Ar sprendžiami konfliktai tarp reikalavimų?	+	-	-	+
Ar siejami reikalavimai su PA?	+	-	+	+
Ar nustatomi besikertantys reikalavimai?	-	+	-	+
Ar turi šablonus reikalavimams specifikuoti?	+	-	-	+
Ar sudaromi ryšiai tarp FR ir NFR	+	+	+	+
Ar sudaromi ryšiai siejantys FR, NFR ir besikertančius reikalavimus?	-/+	-/+	-	+
Ar nustatomi prioritetai reikalavimams?	-	-	-	+
Ar taikoma dekompozicija?	+	+	-/+	+
Ar nustatomi sąveikos taškai?	-	-	+	+
Ar sudaromos kompozicijos taisyklės?	-	+	-/+	+
Ar nustatomos kontribucijos?	-	+	+	+

7. IŠVADOS

1. Atlikus reikalavimų integralumo metodų ir būdų analizę, pastebėta kad vieningo ir nusistovėjusio metodo nėra. Pastebėta, kad moksliniuose darbuose naudojamos vis kitos priemonės ir principai, siekiant užtikrinti FR ir NFR reikalavimų integralumą, tokios kaip UML diagramos, leksikonai, asociacijos ir kitos.
2. Atlikus reikalavimų specifikavimo šablonų analizę, nustatyta, kad Volere šablonas pasižymi teigiamomis savybėmis, tokiomis kaip reikalavimų sekamumas ir susietumas geriausiai išpildo visas reikalavimų specifikavimo charakteristikas lyginant su kitais šablonais, todėl šis šablonas pasirinktas kaip pagrindas integralumą užtikrinančiam metodui. Tačiau, siekiant užtikrinti detalesnę reikalavimų integralumą, būtinas šablono išplėtimas.
3. Atlikus reikalavimų specifikavimo įrankių analizę nustatyta, kad Requisite Pro įrankis yra tinkamas sukurto metodo taikymui. Šiuo įrankiu galima sėkmingai valdyti pastovius ar nuolat besikeičiančius reikalavimus, pagerinti sekamumą, sieti reikalavimus vienus su kitais, sumažinti projekto riziką ir pagerinti jo kokybę.
4. Atlikus panašių sprendimų analizę išsiaiškinta, kad reikalavimų integralumas siejamas su dauguma procesų. Pastebėta, kad yra technikos stoka, norint integruoti funkcinius reikalavimus su nefunkciniais reikalavimais, norint integruoti reikalavimus reikia išplėsti UML kalbą. Šiam sprendimui įgyvendinti buvo sukurtas Use Case profilio notacijų išplėtimas. Sukurtos notacijos atspindi naujo reikalavimo šablono ir metodo plėtojimo artefaktus.
5. Darbe pateiktas metodas yra sudarytas iš trijų etapų: identifikavimo, specifikavimo ir reikalavimų komponavimo. Identifikavimo etape identifikuojami reikalavimai ir renkama informacija reikalavimų pagrindimui, tokių kaip, suinteresuotų šalių ir šaltinių identifikavimas. Tai pasiekama, analizuojant turimus dokumentus, įtraukiant esamus suinteresuotųjų šalių interviu užrašus, siekiant suprasti sistemos sritį.
6. Darbe pateiktas sudarytas šablonas reikalavimų specifikavimui. Specifikavimo etape aprašomi identifikuoti reikalavimai, naudojant sudarytą šabloną, reikalavimai specifikuojami vienodai, nepriklausomai nuo jiems būdingų savybių. Specializuotas šablonas renka atitinkamą informaciją konfliktų identifikavimui ir jų sprendimui, atsižvelgiant į kontribucijų ir prioritetų eilutes šablone. Be to, šablonas palaiko atgalinį

atsekamumą, naudojant šablone šaltinių ir suinteresuotųjų šalių eilučių informaciją. Šablonas taip pat padeda nustatyti besikertančius reikalavimus, analizuojant būtinų reikalavimų eilutę šablone.

7. Siūlomo metodo komponavimo etape, kūrėjams ir suinteresuotoms šalims pateikiamas bendras sistemos vaizdas, leidžiantis aptikti konfliktus. Tai suteikia galimybę nustatyti kritinius nesuderinamumus prieš pradėdant architektūrinį sistemos plėtojimą. Nustatytų reikalavimų rinkiniai ir kompozicijos taisyklės padeda projektuotojams tiksliau ir nuosekliau projektuoti sistemą. Nes reikalavimų komponavimo etape sukuriama sistemos projektavimo užuomazgos.
8. Reikalavimų specifikavimas vienoje formoje užtikrina reikalavimo integralumą, reikalavimas specifikuojamas dvejomis perspektyvomis, apimant ne tik kaip reikalavimas sąveikauja su kitais reikalavimais, bet ir kas sąveikauja specifikuojamą reikalavimą.
9. Taikydami sudarytą metodą, analitikai, išvengtų reikalavimų pasikartojimų specifikacijų dokumentuose. Projektuotojai ir kūrėjai galėtų tiksliau projektuoti sistemą dėl pašalintų pasikartojančių reikalavimų, atsižvelgiant į reikalavimų rinkinius išgautus pagal vieną reikalavimą.
10. Darbo tema buvo parašytas ir atspausdintas straipsnis, kuris buvo pristatytas konferencijoje „XVI tarpuniversitetinė magistrantų ir doktorantų konferencija“.

8. LITERATŪRA

1. Ambler, W. S. (2002) *Agile Modeling*. Wiley; 1st edition. 400p. ISBN: 0471202827.
2. Bach, J.S., "DOORS : A Tool Manage Requirements" , [žiūrēta 2010 01 01] prieiga per internetą: <http://www.springerlink.com/content/g763qw2t17366455/>
3. Baniassad E., Clements P., Araújo J., Moreira A., Rashid A., Tekinerdoğan B., "Discovering Early Aspects", Software, IEEE, v.23 n.1, pp.61-70, January 2006
4. Borland® CaliberRM®. Frequently asked questions. [žiūrēta 2010 05 20] Prieiga per internetą: http://www.moonsoft.net/materials/borland_caliber_family_faq.pdf
5. Brackett, J. (1990). Software engineering. Software Engineering Institute, Carnegie Mellon University. TR: SEI-CM-19-1.2, ADA235642. <ftp://ftp.sei.cmu.edu/pub/documents/90.reports/pdf/tr04.90.pdf>
6. CaliberRM Enterprise software requirements management , [žiūrēta 2010 01 11] prieiga internetu http://www.borland.com/downloads/download_caliber.aspx
7. Chung L., Nixon B., Yu E., Mylopoulos J., "Non-Functional Requirements in Software Engineering", The Kluwer international series in software engineering, 1999, ISBN: 0-7923-8666-3.
8. Cysneiros, L. M. , Leite, J. C. S. P., and Neto, J. M. S "A framework for integrating non-functional requirements into conceptual models" Requirements Engineering, 6(2):97–115, 2001. [žiūrēta 2009 12 08] Prieiga per internetą: <http://www.springerlink.com/content/kdd6brnpt0q896ld/?p=8b45f41bb8824254b5ca34c247b73dfa&pi=0>
9. Dimitrov, E., Schmietendorf, E., and Dumke, R., "UML-based performance engineering possibilities and techniques". IEEE Software, 19(1):74{83, January/February 2002. [žiūrēta 2009 12 15] prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00976944&tag=1>
10. Dorfman M., Thayer R. (1990) editors. Standards, Guidelines, and Examples on System and Software Requirements Engineering. IEEE Computer Society Press, Los Alamitos, California.
11. Dromey, R. G. "A model for software product quality". IEEE Transactions on Software Engineering, 21(2):146– 162, 1995.

12. Dusire, S., Flynn, M., Dardenne, N. (2002). Requirements Engineering: Applying Theory to Reality. In 10th Requirements Engineering Conference (RE'02), Essen, Germany, IEEE Computer Society. 300-302.
13. E. Baniassad, P. Clements, J. Araújo, A. Moreira, A. Rashid, B. Tekinerdogan, *Discovering Early Aspects*, Software, IEEE, v.23 n.1, pp.61-70, January 2006
14. Glinz, M., Wieringa R. (2007). "Stakeholders in Requirements Engineering." IEEE Software. IEEE Computer Society. 24(2): 18-21. Prieiga per internetą:
15. How to Write Functional Specifications Documents. [žiūrėta 2010 04 11] prieiga per internetą: <http://www.softwaredevelopmenttemplates.com/functional-requirements/how-to-write-functional-specifications-documents/468/>
<http://citeseerx.ist.psu.edu>
16. IBM Corporation. "Essentials of IBM Rational RequisitePro v7.0". *REQ370 / RR331 – October 2006. Student Workbook. Part No. 800-027250-000* [žiūrėta 2011 02 11] prieiga per internetą:
<ftp://ftp.software.ibm.com/software/websphere/JavaDevTools/Resources/ReqPro-Fundamentals/ReqPro-V7-Fundamentals-Student-Workbook.pdf>
17. IBM Rational RequisitePro by IBM Rational Software [žiūrėta 2009 10 30]. Prieiga per internetą <http://www.componentsource.com/products/ibm-rational-requisitepro/summary.html>
18. IBM Rational Software. Rational RequisitePro [žiūrėta 2009.12.02]. Prieiga per internetą: <http://www-306.ibm.com/software/awdtools/reqpro>
19. Karlsson, J., Ryan K. (1997). "A Cost-Value Approach for Prioritizing Requirements." IEEE Software. IEEE Computer Society. 14(5): 67-74.
20. Klariti. Functional Requirements Template. [žiūrėta 2010 04 28] prieiga per internetą: <http://www.klariti.com/templates/Functional-Requirements-Specification-Template.shtml>
21. Krištapaitis, R., (interaktyvus). Reikalavimų inžinerija. [žiūrėta 2010 02 14] Prieiga per internetą: <http://www.slideshare.net/rollis/pi6paskaita>
22. L. Rosenhainer, *Identifying Crosscutting Concerns in Requirements Specifications*, Proceedings of the Aspect-Oriented Requirements Engineering and Architecture Design Workshop, Vancouver, Canada, October 24-28, 2004

23. Larman C., “Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd edition. Upper Saddle River, NJ: Prentice Hall Inc., 2004.
24. Lawrence, B. (1996). *Unresolved Ambiguity. American Programmer.* 9(5). 17–22p.
25. Le Vie, D., Jr. “Writing Software Requirements Specifications”, [žiūrėta 2009 10 15] prieiga internetu:
<http://www.techwhirl.com/techwhirl/magazine/writing/softwarerequirementspecs.htm>
26. Madigan, M. *Requirements Elicitation.* University of Colorado. [interaktyvus]. [žiūrėta 2010m. kovo 4d.] prieiga per internetą:
<http://ecee.colorado.edu/~swengctf/standalone/presentations/ReqsElctatnWk2.ppt>
27. MagicDraw. Architektūra Made Simple. Version 17. user guide:
<http://www.magicdraw.com/files/manuals/SysML%20Plugin%20UserGuide.pdf>
28. Moreira, A., Araujo, J., Brito, I., “Crosscutting Quality Attributes for Requirements Engineering ” Software Engineering and Knowledge Engineering Conference (SEKE), Ischia, Italy, 15-19 July 2002. [žiūrėta 2010 01 04] prieiga per internetą:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04384217>
29. Nuseibeh, B., Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In 22nd International Conference on Software Engineering (ICSE'00), Limerick, Ireland, ACM Press. 35-46.
30. Ramesh B., Jarke J. (2001). “Toward Reference Models for Requirements Traceability”. *IEEE Transactions on Software Engineering*, 27(1):58–93, Jan.
31. Reikalavimų analizė (skaidrės) [žiūrėta 2010 03 11]. Prieiga internetu:
http://vaidila.vdu.lt/~i5dasi/se2/paskaitos/03_reikalavimai.pdf
32. Requirements Resources. Volere Requirements Specification Template [žiūrėta 2009.11.12]. Prieiga per internetą: <http://www.volere.co.uk/template.htm>
33. Robertson S. and Robertson J. (1999) “Mastering the Requirements Process”, ACM Press books. Addison-Wesley, Harlow et.al..
34. Somerville I., Sawyer P. (1997), *Requirements Engineering - A good practice guide*, Wiley, ISBN 0-471-97444-7.
35. Supakkul, S., Chung, L., “Integrating FRs and NFRs: A Use Case and Goal Driven Approach” International Conference on SERA'04, 2004, p. 30-37.

36. TechnoSolutions. TopTeam™ *Analyst*. [žiūrēta 2010 05 20] Prieiga per internetą: http://www.technosolutions.com/topteam_requirements_management.html
37. User Requirements Specification [žiūrēta 2010 01 10]. Prieiga per internetą: <http://www.canzz.com/2009/08/concept-phase-templates/>
38. Volere requirement specification [žiūrēta 2009 11 25]. Prieiga per internetą <http://www.volere.co.uk/template.htm#anchor828967>
39. Wiegers, K. E. (2003) “*Software Requirements, Second Edition*”. Microsoft Press. Redmond, Washington. 516 p. ISBN:0735618798.
40. Wirfs-Brock, R., Wilkerson, B., Wiener, L. (1999). *Designing Object-Oriented Software*. 0-13629825-7. Prentice-Hall.
41. Young R. R. (2004). *The requirements engineering handbook*. Artech House. Nordwood, MA. 254 p. ISBN: 1580532667
42. Zeiss, B., Vega, D., Schieferdecker, I., Neukirchen, H., Grabowski, J., “Applying the ISO 9126 Quality Model to Test Specifications” [žiūrēta 2009 10 25]. Prieiga per internetą: <http://subs.emis.de/LNI/Proceedings/Proceedings105/gi-proc-105-024.pdf>

9. PRIEDAI

9.1. 1 Priedas. ATM sistemos reikalavimų ir PA detalizavimas

ATM sistemos funkciniai reikalavimai

1. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
1	FR	4
Aprašymas	Vartotojas gali išimti pinigus iš savo sąskaitos	
Pagrindimas	Vartotojas pasirenka sąskaitą nurodo suma ir bankomatas išduotą nurodytą suma, jei sąskaitoje yra pakankama suma	
Šaltinis	Užsakovas	
Tikimo kriterijus	Išduodami vartotojui pinigai, kitu atveju pateikiamas pranešimas, kad nepakankama suma sąskaitoje.	
Priklausomybės	Konfliktai	
Pasirinkti sąskaitą	Nėra	
Papildoma medžiaga		
Istorija	2011 05 03	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Tikrinti kortelę Tikslumo: ATM turi duoti klientui tikslią pinigų sumą,	

	<p>kokią nurodė klientas</p> <p>Vykdomo savybės: Sistema turi išduoti pinigus per 5 sekundes klientui</p> <p>Tik pilnai įvykdytos operacijos saugomos audito istorijoje</p> <p>Saugumo: PIN kodas vedamas 3 kartus</p> <p>Tikslumo: kortelės duomenys turi būti nuskaityti be klaidų</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
2	FR	4
Aprašymas	Sistema tikrina ar suma yra 20 kartotinis	
Pagrindimas	ATM būna tik 20 lt kupiūros arba sumos kurios dalijasi iš 20.	
Šaltinis	Užsakovas	
Tikimo kriterijus	Vartotojo įvesta suma turi dalintis iš 20 be liekanos	
Priklausomybės	Konfliktai	
Pasirinkti sąskaitą Įvesti sumą	Nėra	
Papildoma medžiaga		
Istorija	2011 05 03	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		

Būtinai reikalavimai	<p>Tikslumo : ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas</p> <p>Vykdyimo savybė: Sistema turi išduoti pinigus per 5 sekundes klientui</p> <p>Tikrinti kortelę</p> <p>Tikslumo: kortelės duomenys turi būti nuskaityti be klaidų</p>
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
3	FR	4, 7
Aprašymas	Vartotojas gali pasirinkti sąskaitą iš kurios nori pasiimti pinigus	
Pagrindimas	vartotojas gali turėti banke daugiau nei vieną sąskaitą	
Šaltinis	Užsakovas	
Tikimo kriterijus	Vartotojas nusiima pinigus iš pasirinktos sąskaitos	
Užsakovo patenkinimas	Užsakovo nepatenkinimas	
Pasirinkti sąskaitą Įvesti sumą Tikrinti ar suma yra 20 kartotinis	Nėra	
Papildoma medžiaga		
Istorija	2011 05 80	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	

Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Tikrinti kortelę Tikslumo: kortelės duomenys turi būti nuskaityti be klaidų

4. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
4	FR	4, 5
Aprašymas	Operacijos saugomos audito istorijoje	
Pagrindimas		
Šaltinis	Užsakovas	
Tikimo kriterijus	Vartotojo visos atliktos operacijos saugomos audito istorijoje	
Priklausomybės	Konfliktai	
Vartotojas gali išimti pinigus iš savo sąskaitos Vartotojas gali pervesti pinigus iš savo sąskaitos	Nėra	
Papildoma medžiaga		
Istorija	2011 05 80	
Prioritetų sąrašas		
Klientas	Svarbus	
Operatorius	Svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Saugumo: Tik pilnai įvykdytos operacijos saugomos audito istorijoje Saugumo: PIN vedamas 3 kartus	

5. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
5	FR	13
Aprašymas	Tikrinti PIN kodą	
Pagrindimas	PIN kodas apsauga nuo neteisėto prisijungimo	
Šaltinis	Užsakovas	
Tikimo kriterijus	Teisingas PIN kodas	
Priklausomybės	Konfliktai	
Visi reikalavimai susiję su operacijomis	Nėra	
Papildoma medžiaga		
Istorija	2011 05 80	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Tikrinti kortelę Saugumo: PIN vedamas 3 kartus	

6. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
6	FR	12
Aprašymas	Tikrinti banko kortelės galiojimo datą	

Pagrindimas	Patvirtinti, kad kortelės galiojimo data yra vėlesnė negu šios dienos data
Šaltinis	Užsakovas
Tikimo kriterijus	Data turi būti ne vėlesnė nei šios dienos
Priklausomybės	Konfliktai
Kortelės duomenų nuskaitymas. Klaidos žinutė apie negaliojančią kortelę.	Nėra
Papildoma medžiaga	
Istorija	
Prioritetų sąrašas	
Klientas	Labai svarbus
Operatorius	Labai svarbus
Užsakovas	Labai svarbus
Pirkėjas	Labai svarbus
Kontribucijų sąrašas	
Kontribucija	Nėra
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Tikrinti kortelę

7. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
7	FR	12
Aprašymas	Tikrinti kortelę	
Pagrindimas	Kortelė turi būti autentifikuota, kad vartotojas galėtų naudotis sistema	
Šaltinis	Užsakovas	
Tikimo kriterijus	Data turi būti ne vėlesnė nei šios dienos	
Priklausomybės	Konfliktai	
Kortelės duomenų nuskaitymas. Kortelės galiojimo datos tikrinimas	Nėra	

Papildoma medžiaga	
Istorija	
Prioritetų sąrašas	
Klientas	Labai svarbus
Operatorius	Labai svarbus
Užsakovas	Labai svarbus
Pirkėjas	Labai svarbus
Kontribucijų sąrašas	
Kontribucija	Nėra
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Tikslumas: kortelės duomenys turi būti nuskaityti be klaidų

8. lentelė. Funkcinio reikalavimo detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
8	FR	12
Aprašymas	Kortelės duomenų nuskaitymas	
Pagrindimas	Nuskaityti tik banko kortelių duomenis	
Šaltinis	Užsakovas	
Tikimo kriterijus	Tik banko kortelės	
Priklausomybės	Konfliktai	
Kortelės galiojimo datos tikrinimas	Nėra	
Papildoma medžiaga		
Istorija		
2011 05 03		
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Nėra	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Tikslumas: kortelės duomenys turi būti nuskaityti be	

	klaidų Tikrinti kortelę
--	--------------------------------

Nefunkciniai reikalavimai

Saugumo reikalavimai

9. lentelė. Nefunkcinio reikalavimo „Reikalavimai prieigai“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
9	15	13
Aprašymas	Reikalavimas prieigai	
Pagrindimas	PIN kodas vedamas ne daugiau kaip 3 kartus	
Šaltinis	Užsakovas	
Tikimo kriterijus	PIN kodas suvestas neviršijant 3 kartų	
Priklausomybės	Konfliktai	
PIN kodo tikrinimas Kortelės užblokavimas	Nėra	
Papildoma medžiaga		
Istorija	2011 05 11	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija		
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

10. lentelė. Nefunkcinio reikalavimo „Konfidencialumas“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
--------------	----------------	-------------

10	15	13
Aprašymas	Konfidencialumas	
Pagrindimas	Sistemoje esantys duomenys apsaugoti nuo neteisėtos prieigos, kiekvieno kliento duomenys matomi tik to kliento.	
Šaltinis	Užsakovas	
Tikimo kriterijus	Tikrinama ar vartotojas prisijungęs su kortele	
Priklausomybės	Konfliktai	
Visi reikalavimai susiję su banko kortele	Nėra	
Papildoma medžiaga		
Istorija	2011 05 11	
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija		
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

11. lentelė. Nefunkcinio reikalavimo „Vientisumo“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
11	15	13
Aprašymas	Vientisumo	
Pagrindimas	Sistemos duomenys vienareikšmiškai atitinka šaltinio perduotus (iš jo gautus) duomenis, kartu užtikrinant jų panaudojimo teisėtumą	
Šaltinis	Užsakovas	
Tikimo kriterijus		
Priklausomybės	Konfliktai	

Visi reikalavimai susiję su operacijomis		Nėra
Papildoma medžiaga		
Istorija		2011 05 11
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Labai svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija		
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

Veikimo reikalavimai

12. lentelė. Nefunkcinio reikalavimo „Užduočių vykdymo greitis“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
12	12	
Aprašymas	Užduočių vykdymo greitis	
Pagrindimas	Mašina turi reaguoti laiku, kai klientas naudojami sistema įdėdamas ir išimdamas kortelę	
Šaltinis	Užsakovas	
Tikimo kriterijus	Klientas turi įdėti kortelę į ATM	
Priklausomybės	Konfliktai	
Reaguoti per 2< sekundės kai klientas įdeda kortelę Reaguoti laiku per 2< sekundės, kai klientui gražinama kortelė	Nėra	
Papildoma medžiaga		
Istorija		
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Svarbus	
Užsakovas	Labai svarbus	

Pirkėjas	Labai svarbus
Kontribucijų sąrašas	
Kontribucija	Tikslumas: kortelės duomenys turi būti nuskaityti be klaidų
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Nėra

13. lentelė. Nefunkcinio reikalavimo „Reikalavimas tikslumui“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
13	12	
Aprašymas	Reikalavimas tikslumui	
Pagrindimas	Tikslumas: kortelės duomenys turi būti nuskaityti 100 % tikslumu	
Šaltinis	Užsakovas	
Tikimo kriterijus	Klientas turi įdėti kortelę į ATM	
Priklausomybės	Konfliktai	
Visi reikalavimai susiję su banko kortele	yra	
Papildoma medžiaga		
Istorija		
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija	Vykdomo savybė (-) Vykdomo savybė: Kortelės duomenys turi būti nuskaityti per 2 sekundės	
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

14. lentelė. Nefunkcinio reikalavimo „Reikalavimas tikslumui“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
14	12	
Aprašymas	Reikalavimas tikslumui	
Pagrindimas	Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas	
Šaltinis	Užsakovas	
Tikimo kriterijus	Suma turi būti 20 kartotinis	
Priklausomybės	Konfliktai	
Visi reikalavimai susiję su banko kortele	Nėra	
Papildoma medžiaga		
Istorija		
Prioritetų sąrašas		
Klientas	Labai svarbus	
Operatorius	Svarbus	
Užsakovas	Labai svarbus	
Pirkėjas	Labai svarbus	
Kontribucijų sąrašas		
Kontribucija		
Būtinų reikalavimų sąrašas		
Būtinai reikalavimai	Nėra	

Tikslumo: ATM turi duoti klientui tikslią pinigų sumą, kokią nurodė klientas

Panaudojamumo reikalavimai

15. lentelė. Nefunkcinio reikalavimo „Naudojimosi paprastumas“ detalizavimas

Reikalavimas	Klasifikavimas	Įvykis / PA
15	12	
Aprašymas	Naudojimosi paprastumas	
Pagrindimas	Sistema turi padėti vartotojui išvengti	

	klaidų
Šaltinis	Užsakovas
Tikimo kriterijus	Po vieno mėnesio naudojimosi sistema, klaidų vidurkis mažesnis nei 1 proc.
Priklausomybės	Konfliktai
Visi reikalavimai susiję su banko kortele	Nėra
Papildoma medžiaga	
Istorija	
Prioritetų sąrašas	
Klientas Operatorius Užsakovas Pirkėjas	Labai svarbus Svarbus Labai svarbus Labai svarbus
Kontribucijų sąrašas	
Kontribucija	Tikslumas: kortelės duomenys turi būti nuskaityti be klaidų
Būtinų reikalavimų sąrašas	
Būtinai reikalavimai	Nėra

ATM sistemos PA

16. lentelė. PA „Pasirinkti operaciją“ specifikuojimas

4. PANAUDOJIMO ATVEJIS:	Pasirinkti operaciją
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Kai vartotojas prieina prie ATM, paprašoma, kad asmuo įdėtų kortelę. Įdėjus kortelę, kortelė yra tikrinama, jei kortelė galiojanti tęsiama operacija. Jei PIN kodas įvestas teisingai operacija tęsiama, jei ne operacijos negalimos. Vartotojas gali pasirinkti kokią operaciją jis nori atlikti: patikrinti balansą, išimti ar pervesti pinigus.
Prioritetas	Labai svarbus

Prieš sąlyga:	Vartotojas turi turėti banko kortelę
Pagrindinis scenarijus:	1. Vartotojas įdeda kortelę į ATM arba į kortelių skaitytuvą; 2. ATM tikrina ar kortelė yra galiojanti; 3. Jei sėkmingai suvestas kodas, vartotojas pasirenka operaciją iš operacijų sąrašo.
Po-sąlyga:	Vykdoma pasirinkta operacija
Alternatyvus scenarijus	Jeigu kortelės galiojimas pasibaigęs ar neteisingas PIN kodas, rodomos klaidos žinutės
Susiję PA	Suvesti PIN kodą, Išimti pinigus, Pervesti pinigus, Patikrinti balansą

17. lentelė. PA „Neteisingas PIN“ specifikuojimas

5. PANAUDOJIMO ATVEJIS:	Neteisingas PIN
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Jei įvestas PIN neteisingas operacijos atšaukiamos. Vartotojo paprašoma iš naujo įvesti PIN kodą. Bandyti galima tris kartus.
Prioritetas	Labai svarbus
Prieš sąlyga:	Vartotojas turi turėti banko kortelę
Pagrindinis scenarijus:	1. Vartotojas įdeda kortelę į ATM arba į kortelių skaitytuvą ir suveda PIN kodą; 2. ATM tikrina PIN ir jis neteisingas; 3. Vartotojo paprašome pakartotinai įvesti PIN kodą; 4. Jei PIN kodas suvedamas tris kartus neteisingai, kortelė užblokuojama.
Po-sąlyga:	Jeigu bandant prisijungti PIN kodas suvedamas neteisingai tris kartus, kortelė užblokuojama
Alternatyvus scenarijus	Vartotojas neturėtų suvesti neteisingai PIN kodo daugiau negu tris kartus
Susiję PA	Tikrinti PIN kodą, Tikrinti kortelę

18. lentelė. PA „Išimti pinigus“ specifikuojimas

6. PANAUDOJIMO ATVEJIS:	Išimti pinigus
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas gali išimti iš savo pasirinktos sąskaitos pinigus. Išduodama suma tik 20 lt kupiūromis, ne didesnė nei pasirinktos sąskaitos balansas.

Prioritetas	Labai svarbus
Prieš sąlyga:	<ol style="list-style-type: none"> 1. Įdedama banko kortelė 2. Tikrinama kortelė 3. Įvedamas PIN 4. Patikrinamas PIN 5. Pasirenkama sąskaita 6. Pasirenkama pinigų išėmimo funkcija 7. Įvedama suma 8. Patikrinama ar suma dalijasi iš 20 9. Patikrinama ar yra sąskaitoje pakankamai pinigų 10. Įrašomas žurnalo įrašas 11. Išduodami pinigai
Pagrindinis scenarijus:	Išduodami pinigai
Po-sąlyga:	Spausdinamas čekis Grąžinama kortelė
Alternatyvus scenarijus	Vartotojas pats atšaukė operaciją.
Susiję PA	Spausdinti čekį, Pasirinkti sąskaitą

19. lentelė. PA „Pervesti pinigus“ specifikuojamas

7. PANAUDOJIMO ATVEJIS:	Pervesti pinigus
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas perveda pinigus iš vienos savo sąskaitos į kitą.
Prioritetas	Labai svarbus
Prieš sąlyga:	<ol style="list-style-type: none"> 1. Įdedama banko kortelė 2. Tikrinama banko kortelė 3. Įvedamas PIN kodas 4. Tikrinamas PIN kodas 5. Pasirenkama pinigų pervedimo funkcija 6. Pasirenkama sąskaita iš kurios pervesti pinigai 7. Pasirenkama sąskaita į kuria pervesti pinigai 8. Įvedama suma 9. Patikrinama ar yra sąskaitoje pakankamai pinigų 10. Įrašomas žurnalo įrašas
Pagrindinis scenarijus:	Pervedami pinigai iš vienos sąskaitos į kitą
Po-sąlyga:	Spausdinamas čekis Grąžinama kortelė
Alternatyvus scenarijus	Vartotojas pats atšaukė operaciją.
Susiję PA	Spausdinti čekį, Pasirinkti sąskaitą

20. lentelė PA „Patikrinti balansą“ specifikuojimas

8. PANAUDOJIMO ATVEJIS:	Patikrinti balansą
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas gali peržiūrėti savo sąskaitos balansą
Prioritetas	Svarbus
Prieš sąlyga:	<ol style="list-style-type: none"> 1. Įdedama banko kortelė 2. Tikrinama banko kortelė 3. Įvedamas PIN kodas 4. Tikrinamas PIN kodas 5. Pasirenkama patikrinti balansą funkciją 6. Pasirenkama sąskaita norint peržiūrėti balanso informaciją
Pagrindinis scenarijus:	Patikrinama balanso informacija
Po-sąlyga:	Spausdinamas čekis Grąžinama kortelė
Alternatyvus scenarijus	Vartotojas pats atšaukė operaciją.
Susiję PA	Spausdinti čekį, Pasirinkti sąskaitą

21. lentelė PA „Pasirinkti sąskaitą“ specifikuojimas

9. PANAUDOJIMO ATVEJIS:	Pasirinkti sąskaitą
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas pasirenka sąskaitą su kuria nori atlikti operacijas: išimti pinigus, pervesti pinigus, patikrinti balansą.
Prioritetas	Labai svarbus
Prieš sąlyga:	<ol style="list-style-type: none"> 1. Įdedama banko kortelė 2. Tikrinama banko kortelė 3. Įvedamas PIN kodas 4. Tikrinamas PIN kodas 5. Pasirenkama sąskaita su kuria norima vykdyti operacijas
Pagrindinis scenarijus:	Pasirenkama sąskaita su kuria bus atliekamos operacijos
Po-sąlyga:	Spausdinamas čekis Grąžinama kortelė
Alternatyvus scenarijus	Vartotojas pats atšaukė operaciją.
Susiję PA	Išimti pinigus, Pervesti pinigus, Patikrinti balansą

22. lentelė PA „Spausdinti čekį“ specifikuojimas

9. PANAUDOJIMO ATVEJIS:	Spausdinti čekį
-------------------------	-----------------

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Vartotojas atlikęs operaciją, gali atsispausdinti čekį
Prioritetas	Labai svarbus
Prieš sąlyga:	<ol style="list-style-type: none"> 4. Įdedama banko kortelė 5. Tikrinama banko kortelė 6. Įvedamas PIN kodas 7. Tikrinamas PIN kodas 8. Pasirenkama sąskaita su kuria norima vykdyti operacijas 9. Atlikus operaciją spausdinamas čekis
Pagrindinis scenarijus:	Atlikus operaciją spausdinamas čekis
Po-sąlyga:	Grąžinama kortelė
Alternatyvus scenarijus	Vartotojas pats atšaukė operaciją.
Susiję PA	Išimti pinigus, Pervesti pinigus, Patikrinti balansą

9.2. 2 Priedas. Darbų pasiskirstymo lentelė

Darbų pasiskirstymas		
Magistro darbo turinys	Jolanta Čekanauskaitė	Jūratė Čekanauskaitė
Reikalavimų, reikalavimų integralumo metodų, specifikavimo šablonų ir reikalavimų valdymo įrankių analizė	52 %	48 %
Reikalavimų analizė	70 %	30 %
Reikalavimų gavybos strategijų analizė	70 %	30 %
Reikalavimų specifikavimo šablonų analizė	40 %	60 %
Reikalavimų specifikavimo įrankių analizė	30 %	70 %
FR ir NFR reikalavimų integralumo metodų ir principų analizė	50 %	50 %
Reikalavimų specifikavimo integralumą užtikrinantis modelis	47 %	53 %
Reikalavimų integralumą užtikrinančio metodo procesų modelis	50 %	50 %
Reikalavimų identifikavimo procesas	60 %	40 %
Reikalavimų specifikavimo procesas	30 %	70 %
Reikalavimų komponavimo procesas	50 %	50 %
Reikalavimų integralumą užtikrinantis modelis	60 %	40 %
Reikalavimų integralumą užtikrinančio metodo taisyklės	30 %	70 %
Metodo realizacija Requisite Pro ir MagicDraw įrankiais	55 %	45 %
Šablono kūrimas RequisitePro įrankiu	40 %	60 %
Reikalavimų integralumo metodo atvaizdavimas UML	70 %	30 %
Metodo realizacija pasirinktomis priemonėmis	60 %	40 %
Pasirinktos dalykinės srities aprašas	60 %	40 %
Reikalavimų specifikavimo etapas	50 %	50 %
Reikalavimų rinkinių atvaizdavimas UML	70 %	30 %
Reikalavimų integralumo metodikos eksperimentinis tyrimas	40 %	60 %

9.3. 3 Priedas. Straipsnis

FUNKCINIŲ IR NEFUNKCINIŲ REIKALAVIMŲ SPECIFIKACIJOS INTEGRALUMO UŽTIKRINIMO METODAS

Jūratė Čekanauskaitė¹, Jolanta Čekanauskaitė², Rimantas Butleris³

*Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų g. 50, Kaunas, Lietuva,
1jurate.cekanauskaite@gmail.com, 2jolanta.cekanauskaite@gmail.com, 3rimantas.butleris@ktu.lt*

Santrauka (abstract). Straipsnyje analizuojami funkcinių ir nefunkcinių reikalavimų integravimo metodai, aptariami integruotų reikalavimų privalumai. Sudarytas metodas, užtikrinantis reikalavimų integralumą, naudojant kontribucijas ir kompozicijos taisykles. Metodas grindžiamas reikalavimų rinkiniais, kurie apima besikertančius reikalavimus ir rinkiniams taikomas kompozicijos taisykles. Metodui įgyvendinti naudojamas papildytas *Volere* šablonas funkciniais ir nefunkciniais reikalavimams specifikuoti.

Raktiniai žodžiai: funkciniai ir nefunkciniai reikalavimai, integruoti reikalavimai, reikalavimų specifikacija.

1. Įvadas

Sėkmingo projekto pagrindas - aiškiai suvokiami reikalavimai. Tokie reikalavimai aprašomi ankstyvojoje informacinės sistemos (IS) kūrimo stadijoje ir atsispindi visame jos gyvavimo cikle. Kuriant IS funkciniai reikalavimai (FR) ir nefunkciniai reikalavimai (NFR) atlieka esminį vaidmenį, todėl svarbu juos aiškiai ir tiksliai specifikuoti.

Reikalavimų gausa – pagrindinė problema. Norint užtikrinti reikalavimų specifikacijos kokybę, būtina užtikrinti reikalavimų integralumą. Visoje IS kūrimo projekto eigoje gali atsirasti nauji reikalavimai, o esami - keistis ar tapti nereikalingais. Siekiant užtikrinti reikalavimų integralumą viso projekto metu, yra būtina sieti reikalavimus su kitais reikalavimais, susijusiais panaudojimo atvejais, taip pat kuo anksčiau nustatyti konfliktinius reikalavimus, taikyti kompozicijos taisykles reikalavimų rinkiniams.

2. Reikalavimų integralumo užtikrinimo principų analizė

Reikalavimų integralumas - tai ryšiai tarp reikalavimų, kurie užtikrina jų nuoseklumą ir vientisumą. Integruoti FR ir NFR sudaro vientisus reikalavimų rinkinius, kuriuos lengva išgauti įvairiais pjūviais. Taip pat reikalavimų integralumas palengvina jų atsekamumą bei išgavimą, nes kiekvienas reikalavimas yra susietas tarpusavio ryšiais [6].

Integruotų reikalavimų privalumai:

- Palengvina sistemos kūrimo procesą, didina reikalavimų suprantamumą ir palaikomumą;
- Mažina IS plėtojimo laiką. Sistemos projektuotojams lengviau naudotis tokia reikalavimų specifikacija, nes kiekvienas reikalavimas turi ryšį su vienu ar keliais reikalavimų specifikacijos atributais, kurie gali būti išgauti įvairiais pjūviais pagal projektuotojų poreikius.

Lawrence Chung siūlo integruoti NFR su FR į UML panaudojimo atvejų (PA) modelį [5]. Kai pateikiami tikslūs NFR, jie integruojami į tam tikrus PA diagramos taškus, vadinamus NFR asociacijos taškais. Kadangi reikalavimų inžinieriai naudoja įprastą PA modeliavimą, jie taip pat turi įvertinti ir visus reikalingus NFR. Reikalavimų inžinierius privalo būti tikras ar NFR yra priskirtas visam PA ar tik tam tikrai jo daliai, susiejant NFR su tam tikrais PA modelio elementais: aktoriais, panaudojimo atveju, aktorių panaudojimo atvejo asociacijomis ir sistemos ribomis.

Ana Moreira pasiūlė integruoti NFR su PA [1], išplečiant panaudojimo atvejų modelį ryšiu <<constrains>>, kuris susieja panaudos atvejus su identifikuotais NFR.

3. Reikalavimų integralumą užtikrinantis metodas

Siekiant užtikrinti reikalavimų integralumą, būtina tinkamai specifikuoti reikalavimus bei sudaryti ir palaikyti reikalavimų sąryšius. Siūlomas metodas grindžiamas šiais etapais: 1) Projekto registravimas; 2) Reikalavimų išgavimas/identifikavimas; 3) Reikalavimų specifikuojimas; 4) Reikalavimų komponavimas.

Reikalavimų integralumą užtikrinančios saugyklos modelis pateiktas 1 pav.. Saugykloje registruojami FR ir NFR, remiantis sudarytu šablonu ir dalykinės srities terminų žodynu. Saugykloje išsaugoma informacija apie reikalavimų šaltinius, suinteresuotas šalis, PA, reikalavimų priklausomybes, tuo užtikrinant galimybes analizuoti reikalavimus įvairiais pjūviais. Saugyklos modelyje esančios kompozicijos taisyklės, susiję PA, konfliktiniai reikalavimai užtikrina reikalavimų specifikuojimo integralumą.

Visų pirma registruojamas projektas, kuriam reikalavimų specifikuojimo turi būti sudaryta. Registruojant projektą įvardinamas projekto identifikatorius, pavadinimas, tikslas ir užsakovas. Kadangi modelis nėra griežtai apibrėžtas, galima įtraukti ir papildomus atributus. Toliau sudaromas terminų žodynas, kuriame talpinami terminai, naudojami specifikuojant reikalavimus. Įtraukiant terminą pateikiamas jo glaustas paaiškinimas. Toks žodyno sudarymas leidžia išvengti sąvokų dviprasmybių. Žodynas naudojamas bei pildomas viso projektavimo metu.

Reikalavimų išgavimo procesas susideda iš penkių etapų:

1. Suinteresuotų šalių identifikavimas. Suinteresuotų šalių poreikių nustatymas, kurio metu identifikuojami visi asmenys, organizacijos, kurie turi įtakos reikalavimams.

2. Šaltinių identifikavimas. Tikslas surinkti visus esančius dokumentus, pavyzdžiui, organizacijos struktūros diagramas, procesų modelius, standartus, katalogus.

3. Terminų žodyno pakartotinis panaudojimas. Siekiant palengvinti reikalavimų identifikavimą, siūloma naudoti jau egzistuojančius žodynus, aptartus Chung ir Wieger [5, 6].

4. Reikalavimų klasifikavimas. Reikalavimai klasifikuojami į FR ir NFR. Visų tipų reikalavimai yra svarbūs ir turi būti aprašomi tinkamomis sąvokomis.

5. PA detalizavimas. Kiekvieną sistemos panaudojimo atvejį detalizuojame, išreiškiant PA tekstiniu aprašu, išskiriant PA identifikatorius, aktorius, pavadinimus, prioritetus, išankstines sąlygas, pagrindinius scenarijus, scenarijaus alternatyvas, po sąlygas ir susijusius panaudos atvejus. Tokį PA aprašymą pasiūlė C. Larman [2]. Vienas iš svarbiausių atributų aprašant PA yra *susiję PA*, nes čia sukuriamas ryšys, siejantis vieną PA su kitu. Taigi, kiekvienas FR yra susietas su vienu ar daugiau panaudojimo atvejų. Kartu su FR specifikuojami ir NFR, kurie gali būti susiję su panaudos atvejais ar FR.

FR ir NFR reikalavimai specifikuojami remiantis Volere šablonu [3]. Šablonas papildytas specialiais atributais, kurie užtikrina visapusišką FR ir NFR integralumą (1 pav.). Į šabloną įtraukiamos kompozicijos taisyklės bei kontribucijų nustatymas.

Reikalavimų specifikuojimo procesas susideda iš trijų etapų:

1. Priklausomybių nustatymas. Tai reikalavimų priklausomybių sąrašas, kuriomis reikalavimas pasižymi. Reikalavimai gali būti apibrėžti kaip susietų reikalavimų rinkiniai, kurie nusakomi priklausomybėmis. Reikalavimas, neturintis priklausomybės, parodo, kad kažkas yra praleista arba, kad reikalavimas neturi prasmės, taigi reikia pakartoti reikalavimų identifikavimą.

2. Kontribucijos nustatymas. Kontribucija, tai ryšys tarp dviejų reikalavimų, kuris apibrėžia būdą, kaip vienas reikalavimas veikia kitą. Kontribucijos gali būti bendradarbiaujančios (teigiamai veikiančios ir padedančios įvykdyti reikalavimą) ir jos atvaizduojamos „+“, arba prieštaraujančios (neigiamai veikiančios ir kliudančios įvykdyti reikalavimą) ir jos atvaizduojamos „-“.

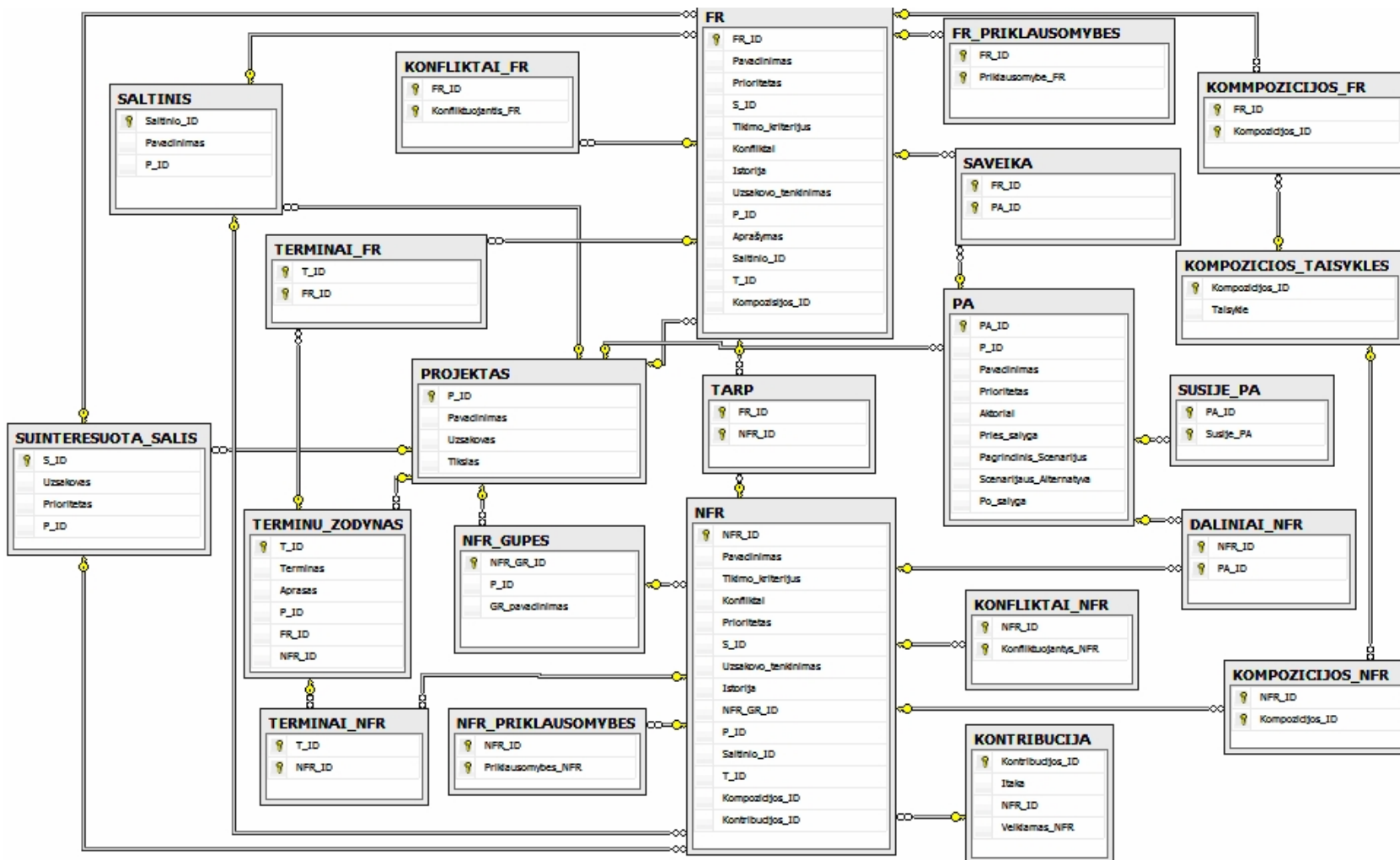
3. Prioritetų nustatymas. Ne visi reikalavimai yra vienodai svarbūs, todėl jiems suteikiami prioritetai. Prioritetas nusako reikalavimo svarbumo laipsnį, kurį nustato užsakovas.

Sekantis procesas, tai reikalavimų komponavimas. Jo tikslas yra sukomponuoti reikalavimus ir suteikti kūrėjams visą sistemos vaizdą bei nustatyti ir valdyti konfliktus tarp reikalavimų. Jį sudaro keturi žingsniai:

1. Sąveikos taškų nustatymas. Nustatomi taškai, kuriuose reikalingos kompozicijos taisyklės. Sąveikos taške vienas ar keli reikalavimai turi būti sukomponuoti kartu. Jis sudaro reikalavimų rinkinį. Vienas iš reikalavimų atlieka pagrindinio reikalavimo vaidmenį, kuris susieja likusius reikalavimus.

2. Besikertančių reikalavimų nustatymas. Reikalavimas yra besikertantis, jei jis būtinas dviems arba keliems kitiems reikalavimams.

3. Konfliktų valdymas. Šio žingsnio tikslas yra nustatyti ir išspręsti konfliktines situacijas tarp reikalavimų. Konfliktines situacijas nustato kūrėjai. Konfliktai gali atsirasti tam tikruose sąveikos taškuose.



1 pav. Reikalavimų integralumą užtikrinančios saugyklos modelis

Atsižvelgiant į tai, konfliktai iškyla bet kuriuo metu tarp dviejų ar daugiau reikalavimų, kurie veikia vienas kitą neigiamai (žr. kontribucijų nustatymas) ir turi tokį pat prioritetą, taip pat priklauso tam pačiam sąveikos taškui. Norint išspręsti konfliktus, reikia suteikti reikalavimams skirtingus prioritetus.

4. Kompozicijos taisyklių nustatymas kiekvienam sąveikos taškui. Kompozicijos taisyklės apibrėžia tvarką, kaip reikalavimai bus vykdomi konkrečiame sąveikos taške. Kompozicijos taisyklės traktuojamos dviem skirtingomis perspektyvomis: 1) tvarka, kaip reikalavimas bus sukomponuotas su pagrindiniu reikalavimu, kuris apibrėžia sąveikos tašką; 2) vykdymu, kaip reikalavimai bus integruoti su pagrindiniu reikalavimu.

Kompozicijos taisyklės naudojamos reikalavimams komponuoti [4]:

1. Seka (žymima $R1 \gg R2$): nurodo seką reikalavimų ir reiškia, kad $R2$ vykdomas tik tada, kai $R1$ reikalavimas įgyvendintas.

2. Lygiagretumas (žymima $R1 \parallel R2$): reiškia lygiagrečius reikalavimų vykdymus, t.y., kad reikalavimas $R1$ turi būti sinchronizuojamas su reikalavimu $R2$.

3. Pasirinkimas (žymimas $R1 \square R2$): išreiškia pasirinkimą ir reiškia, kad tik vienas reikalavimas bus įgyvendintas ($R1$ arba $R2$).

Atlikus pateikto metodo analizę kitų metodų kontekste, pastebėta, kad jis pasižymi reikalavimų išgavimo įvairiais pjūviais ir kompozicijos taisyklių taikymo reikalavimų rinkiniams galimybėmis. Tradiciškai reikalavimai išgaunami tik per PA pjūvį, o pateiktame metode reikalavimai gali būti analizuojami šaltinių, priklausomybių ir suinteresuotų šalių pjūviais. Kompozicijos taisyklės padeda projektuotojams sklandžiau įgyvendinti reikalavimus, nes kompleksiskai nusakomos reikalavimų įgyvendinimo sąlygos ir jų eiliškumas. Lawrence Chung [5] metodikoje konfliktai sprendžiami įvertinant IS tikslus, o pateiktame metode konfliktai sprendžiami atsižvelgiant į reikalavimų prioritetus, kas įneša tvarkos į reikalavimų surinkimo procesą ir padidina specifikacijos kokybę. Pateiktas metodas palaiko reikalavimų susietumą, nes juos specifikuojant sudaromi reikalavimų rinkiniai, kurie užtikrina specifikacijos integralumą.

4. Išvados

Pateiktas metodas suteikia galimybę sudaryti integralią reikalavimų specifikaciją, reikalavimų kompozicijai panaudojant sąveikos taškus ir taisykles. Taisyklių naudojimas reikalavimų rinkiniams sistemos projektuotojams padeda lengviau ir greičiau suvokti bei susisteminti reikalavimus. Metodas padidina klaidų reikalavimų specifikacijoje aptikimo ir pašalinimo efektyvumą. Konfliktai tarp reikalavimų aptinkami ir sprendžiami ankstyvojoje projekto rengimo stadijoje.

Šis metodas išbandytas studentų darbų įkėlimo ir kaupimo sistemos reikalavimams aprašyti, panaudojant RequisitePro įrankį funkciniais ir nefunkciniais reikalavimams specifikuoti remiantis sudarytu reikalavimų specifikacijos saugyklos modeliu.

Literatūros sąrašas

- [1] **Araujo J., Moreira A.** An Aspectual Use Case Driven Approach. *VIII Jornadas de Ingeniería de Software y Bases de Datos (JISBD)*, 2003, p. 12-14.
- [2] **Larman C.** Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design and the Unified Process 3rd Edition. *Upper Saddle River, NJ: Prentice Hall Inc.*, 2004.
- [3] **Requirements Resources.** Volere Requirements Specification Template. [žiūrėta 2011.01.20]. Prieiga per internetą: <http://www.volere.co.uk/template.htm>
- [4] **Sardinha A., Chitchyan R., Weston N., Greenwood P., and Rashid A.** Ea-analyzer: Automating conflict detection in aspect-oriented requirements. *Information Sciences and Technologies Bulletin of the ACM*, 2010, Vol. 2, No. 1, p. 56-59.
- [5] **Supakkul S., Chung L.** Integrating FRs and NFRs: A Use Case and Goal Driven Approach. *International Conference on SERA'04*, 2004, p. 30-37.
- [6] **Wieggers K. E.** Software Requirements 2nd Edition. *Microsoft Press*, 2003.

A method for integrity support of functional and non-functional requirements specification

In this article methods for the functional and non-functional requirements integration are analyzed. Composed method ensures the integrity of requirements on the basis of contributions and compositions rules. The method is based on a set of requirements, which includes crosscutting requirements. For those sets of requirements the composition rules are applied. The method is implemented using the extended Volere template for functional and non-functional requirements specification.