

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

**ALBERTAS MORKEVIČIUS**

**MAGISTRO BAIGIAMASIS DARBAS**

**REIKALAVIMŲ SPECIFIKAVIMO ŠABLONAIŠ MODELIS  
IR JO PROGRAMINIS PROTOTIPAS**

Leidžiama ginti \_\_\_\_\_

Magistrantas \_\_\_\_\_Albertas Morkevičius\_\_\_\_\_

Darbo vadovas \_\_\_\_\_prof. dr. Rimantas Butleris\_\_\_\_\_

\_\_\_\_\_

Darbo įteikimo data\_\_\_\_\_

Registracijos Nr. \_\_\_\_\_

## **Kaunas 2008**

## TURINYS

SANTRUMPŲ SĄRAŠAS .....	3
PAVEIKSLŲ SĄRAŠAS.....	4
LENTELIŲ SĄRAŠAS.....	5
SANTRAUKA .....	6
ĮVADAS.....	7
1.REIKALAVIMŲ SPECIFIKAVIMAS.....	8
1.1. Reikalavimų specifikuojimo samprata .....	8
1.2. Aukštos kokybės reikalavimų proceso nauda .....	9
1.3. Reikalavimų specifikacijos struktūra .....	10
1.4. Būdai specifikuoti programinės įrangos reikalavimus .....	12
2. REIKALAVIMŲ SPECIFIKAVIMAS ŠABLONAIŠ .....	13
2.1. IEEE 830 -1998 standartas .....	15
2.2. Volere šablonas .....	17
2.3. Reikalavimų pasikeitimas .....	31
2.5. Reikalavimų specifikuojimas naudojant UML.....	31
3. REIKALAVIMŲ INŽINERIJOS PROGRAMINIAI ĮRANKIAI.....	33
3.1. Programinių įrankių samprata, problemos, galimi sprendimai .....	33
3.2. Reikalavimų inžinerijos įrankių apžvalga .....	34
3.3. Reikalavimų specifikuojimo įrankių analizė .....	37
4. REIKALAVIMŲ SPECIFIKAVIMO MODELIS.....	42
4.1 Volere šablono modelis .....	42
4.2 Volere reikalavimų surinkimo modelis .....	45
5. MODELIO PROTOTIPAS .....	51
IŠVADOS.....	62
LITERATŪROS SĄRAŠAS.....	63
PRIEDAS NR.1 .....	64
PRIEDAS NR.2.....	65

## **SANTRUMPŲ SĄRAŠAS**

IT - Informacinės technologijos;

IS – Informacinės sistemos;

IEEE – Elektros ir elektronikos inžinerijos institutas (angl. Institute of Electrical and Electronics Engineers, Inc.);

UML – Unifikuota modeliavimo kalba (angl. Unified Modeling Language);

PA – Panaudojimo atvejs;

CASE – Kompiuterizuota informacinių sistemų inžinerija (angl. Computer aided systems engineering);

DXL - Doors išplėtimo kalba.( angl. Doors eXtension Language)

## PAVEIKSLŲ SĄRAŠAS

1 pav. Reikalavimų supratimo reikšmė sistemai .....	8
2 pav. Reikalavimų struktūra tarp kliento iki reikalavimų specifikacijos .....	11
3 pav. Ribos tarp reikalavimų kūrimo ir reikalavimų valdymo procesų .....	11
4 pav. Panaudojimo atvejų surinkimas.....	23
5 pav. Volere reikalavimų surinkimo forma .....	23
6 pav. Doors šablonų biblioteka .....	42
7 pav. Volere šablono Doors įrankyje veiklos diagrama .....	43
8 pav. Volere šablono ir Doors įrankio komponentų diagrama .....	44
9 pav. Modifikuota Volere reikalavimų surinkimo forma .....	46
10 pav. Volere šablono pritaikymo konceptualus modelis .....	47
11 pav. Modelio panaudos atvejų diagrama.....	48
12 pav. Reikalavimų suvedimo modelio veiklos diagrama.....	49
13 pav. Volere šablono realizacijos Doors reikalavimų valdymo pakete modelis.....	50
14 pav. Volere šablono realizacija Doors šablonų bibliotekoje .....	51
15 pav. Volere šablono realizacijos prototipas.....	52
16 pav. Panaudos atvejo įvedimo forma .....	53
17 pav. Panaudojimo atvejų sąrašo fragmentas .....	53
18 pav. Panaudojimo atvejų šablono specifikacija.....	54
19 pav. Volere reikalavimų surinkimo forma .....	54
20 pav. Volere reikalavimų specifikacija .....	56
21 pav. Sąryšiai su priklausomais reikalavimais ir panaudos atvejais .....	56
22 pav. Panaudojimo atvejo sąryšis su reikalavimų moduliu .....	57
24 pav. Sąryšių tarp reikalavimų ir aktorių pjūvis .....	58
25 pav. Sąryšių tarp reikalavimų ir panaudojimo atvejų pjūvis .....	58
26 pav. Neautomatizuoto reikalavimų surinkimo veiklos.....	59
27 pav. Automatizuoto Volere reikalavimų surinkimo veiklos .....	60

## **LENTELIŲ SĄRAŠAS**

Lentelė nr. 1 Reikalavimų pakitimų ir versijų valdymo šablonas .....	17
Lentelė nr. 2 Reikalavimų inžinerijos įrankių analizė.....	37

## **SANTRAUKA**

MORKEVIČIUS, Albertas (2008) *Template Based Model for Requirements Specification and its Prototype*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 56 p.

## **S U M M A R Y**

The main object of this thesis is to analyze requirements specification templates and requirements management tools and realize model prototype using requirements specification templates.

There are analyze of requirements templates and requirements managing tools results in considering about a gain of high quality requirements process. To introduce a conception witch supports clearly requirement specification process – the model of Volere requirements specification template adopting Telelogic Doors requirements managing tool. Model realized in Doors using DXL - Doors eXtension Language. Created prototype reduce 30% of users activities and lets user to get more efficiency and productivity in requirements specification process.

Number of tables: 2.

Number of pictures: 27.

## IVADAS

Darbe pristatomas reikalavimų specifikavimo šablonais modelis. Aptariama aukštos kokybės reikalavimų proceso nauda kuriamoms sistemoms. Pateikiami išanalizuotų reikalavimų specifikacijos dokumento standartų ir reikalavimų inžinerijos programinių įrankių detalios analizės rezultatai.

Pristatoma koncepcija geriau užtikrinanti reikalavimų identifikavimo ir specifikavimo kokybę - sumodeliuotas reikalavimų specifikacijos modelis, panaudojantis Volere reikalavimų surinkimo šabloną, pritaikant reikalavimų specifikavimo programinės įrangos įrankį Telelogic Doors. Sukurto prototipo pagalba, sistema automatizuoja reikalavimų surinkimo procesą, sumažindama 30 % vartotojo veiksmų, taip sudarydama didesnę vartotojo darbo našumą ir efektyvumą reikalavimų specifikavimo procese.

**Darbo tikslas** - išanalizuoti reikalavimų specifikavimo standartus – šablonus, programinę įrangą, bei suprojektuoti ir realizuoti modelio programinį prototipą.

### **Uždaviniai:**

- apžvelgti ir išanalizuoti reikalavimų specifikavimo standartus;
- atlikti reikalavimų inžinerijos įrankių analizę;
- sukurti reikalavimų specifikavimo šablonais sistemos modelį;
- realizuoti reikalavimų specifikavimo modelio programinį prototipą.



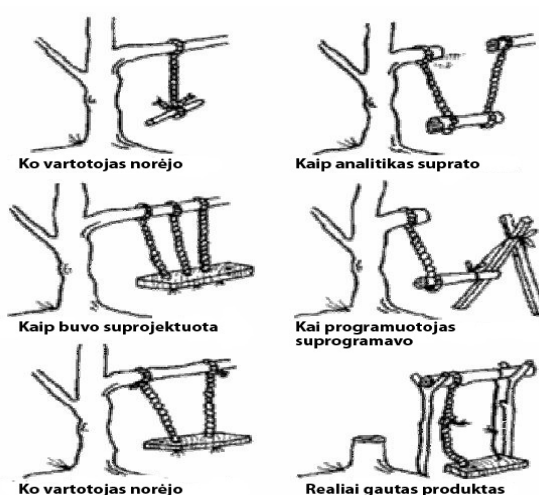
# 1. REIKALAVIMŲ SPECIFIKAVIMAS

## 1.1. Reikalavimų specifikuojimo samprata

Pasaulyje egzistuoja įvairių metodų ir priemonių, skirtų kompiuterizuotoms informacinėms sistemoms (IS) kurti. Vieni apima visą sistemos gyvavimo ciklą etapą, o kiti – skirti atskiroms sistemos kūrimo fazėms kompiuterizuoti. Palyginus su projektavimo ir programavimo kūrimo fazėmis, egzistuoja daug daugiau metodų bei priemonių, negu pradinėse IS kūrimo fazėse.

Bet kurio produkto ar IS kūrimo stadijose yra neapšėinama be reikalavimų aspekto. Reikalavimų dokumentas – oficialus pareiškimas, ko reikalaujama iš sistemos kūrėjų. Reikalavimai reikalingi nustatyti – išsiaiškinti užsakovo poreikius kuriamam produktui, produkto būsimas galimybes, specifikas, ką turi daryti ir atlikti sukurtas produktas. Šis procesas yra labai svarbus, nuo jo priklauso tolimesnės gaminamos sistemos kūrimo (projektavimo, modeliavimo, realizavimo) stadijos, fazės. Kuo geriau bus išsiaiškinta ką produktas turi daryti, atlikti, turėti, tuo tiksliau bus įgyvendintas produktas, tuo mažiau reikės laiko ir darbo sąnaudų jo tobulinimui netiksliai specifikuotų reikalavimų atveju. Taigi, reikalavimai – reikšmingas veiksnys sistemos kūrimo procese, kadangi reikalavimų rezultatų kokybė įtakoja vėlesnes sistemos kūrimo fazes.

Reikalavimų procesas nustato funkcionalumus, kurių klientai reikalauja iš sistemos, ir taikomus sistemos veikimo ir kūrimo apribojimus. Reikalavimai yra sistemos atliekamų funkcijų ir apribojimų aprašymai. Reikalavimai nurodo KĄ sistema turi daryti, bet ne KAIP sistema turi atlikti užduotis.



Šaltinis: autoriaus parengta pagal Šilingas, D. (2005) Programinės įrangos kūrimo metodologija.

1 pav. Reikalavimų supratimo reikšmė sistemai.

Paveikslėlyje 1 pav. akivaizdžiai parodyta nesuprastų, neteisingai nustatytų ir specifikuotų reikalavimų paradoksas. Vartotojas norėjo vieno, bet prastai atliktas reikalavimų fazės etapas, turėjo poveikio ir įtakos visos sistemos kūrimo etapams, bei jos tolimesnei raidai, ir gavosi visai kitas projektas, kuriam dar labai toli iki to projekto kokio norėjo vartotojas.

Taigi, reikalavimų inžinerijos procesas yra labai svarbus, nuo jo priklauso tolimesnės projekto kūrimo (projektavimo, modeliavimo, realizavimo) stadijos, fazės. Kuo geriau bus išsiaiškinta KAS, KA ir KAIP turi daryti, atlikti, turėti, tuo tiksliau bus įgyvendintas produktas. Vadinasi mažiau reikės laiko ir darbo sąnaudų jo tobulinimui netiksliai specifikuotų reikalavimų atveju. Taigi, reikalavimai – reikšmingas veiksnys sistemos kūrimo procese, kadangi reikalavimų rezultatų kokybė įtakoja vėlesnės sistemos kūrimo fazes.

## **1.2. Aukštos kokybės reikalavimų proceso nauda**

Organizacijos, efektyviai vykdančios reikalavimų inžinerijos procesus įgauna daug privalumų ir naudų. Didelė nauda gaunama iš sumažinamo nereikalingo perdirbimo vėlesniuose sistemos kūrimo, bei palaikymo etapuose. Šis reikalavimų kokybės padarinys nėra akivaizdus, ir daug žmonių klaidingai tiki, kad laikas sugaištas reikalavimų analizei, atitinkamai tiek pat uždelsia projekto galutinę sukūrimo datą [11].

Reikalavimų specifikuojimas diskusijų, interviu būdu, pabrėžiamas bendradarbiavimas įtraukiant užsakovus, vartotojus, ir visus suinteresuotus asmenis į projekto kūrimą. Surinkti reikalavimai bet kokio projekto kūrėjams leidžia geriau suprasti vartotojų bendruomenę ar rinką, bei kritinės sėkmes faktorius. Tai daug pigesnis būdas pasiekti šį supratimą prieš sukuriant produktą, nei po produkto atidavimo naudoti vartotojams [11].

Įtraukiant vartotojus į reikalavimų specifikuojimą, padidėja vartotojų entuziazmas bei lojalumas produktui. Koncentruojantis daugiau prie vartotojo uždavinių, nei prie paviršutiniškos produkto analizės, kūrėjai gali išvengti bereikalingai atliekamų darbų, kurie galbūt niekad nebus panaudoti. Vartotojo įtraukimas sumažina atstumą tarp laukiamo produkto kurio tikisi vartotojas, ir realaus produkto kuris bus sukurtas. Reikalavimų specifikacija sunaudoja laiko sąnaudas, bet jos tikrai yra mažesnės, nei taisant klaidas ir sprendžiant problemas testavimo ar palaikymo etapuose, dėl neteisingai identifikuotų ir atliktų darbų [11].

Nepaisant jau paminėtų reikalavimų privalumų, efektyvus pakitimų valdymas sumažina nepalankių veiksnių atsiradimą vykdant reikalavimų pakitimus. Dokumentuoti, vienareikšmiški

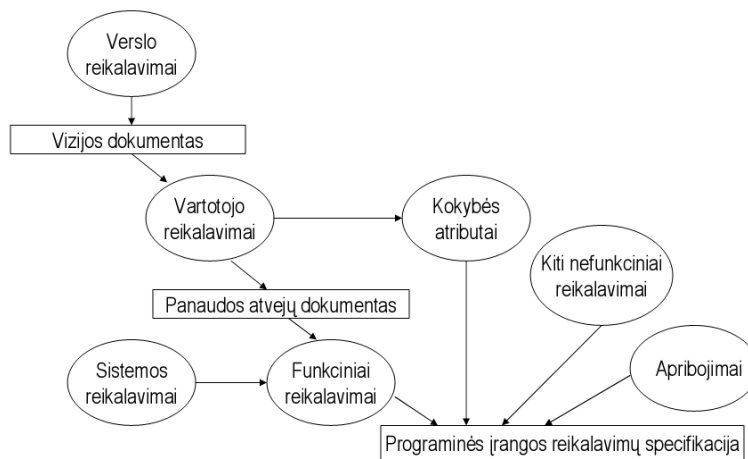
reikalavimai žymiai palengvina sistemos testavimą, kuriame padidėja tikimybė aukštos kokybės produktą, kuriuo bus patenkinti visi su sistema susiję asmenys (užsakovai, rėmėjai, vartotojai ir t.t.).

Niekas negali tiksliai nusakyti geresnių reikalavimų teikiamą naudą. Analitiškai peržvelgus būsimus procesus reikėtų numatyti kuriose vietose bus naudinga specifiškai reikalavimus. Pirmiausia nuspręsti kainą, į kurią įeina praktika, kuriant naujas procedūras ir dokumentų šablonus, darbuotojų komandos apmokymas, knygos įrankiai ir išorinių konsultantų paslaugos. Didžiausia investicija yra laikas, kurį darbuotojų komanda sugaišta rinkdama, dokumentuodama, peržiūrėdama ir valdydama reikalavimus. Toliau rekomenduojama apmastyti galimas naudas ir privalumus, ir kiek tai gali sutaupyti projekto lėšų. Yra sudėtinga nustatyti ar išmatuoti visas galimas naudas, teikiamas naudojant reikalavimų inžineriją, tačiau nauda tikrai yra, ir ji yra reali:

- Sumažėjęs netikslių reikalavimų kiekis;
- Sumažintas produkto kūrimo perdirbimas, taisymas;
- Sumažintos nereikalingos produkto ypatybės ir požymiai;
- Mažesni modifikavimo ir tobulinimo kaštai;
- Greitesnis produkto sukūrimas;
- Mažiau nesusipratimų ir dviprasmybių;
- Realistiškesni užmojai ir galimybės;
- Sumažintas chaosas komandoje ir projekte;
- Tiksliesni sistemos testavimo įvertinimai ir apskaičiavimai;
- Didesnis užsakovų ir kūrėjų pasitenkinimas projektu [2, 10,11].

### **1.3. Reikalavimų specifikacijos struktūra**

Reikalavimų specifikacijos bendrinė struktūra – tai kelias, nuo žemiausio lygmens - verslo reikalavimų iki reikalavimų specifikacijos dokumento, pavaizduotas 2 pav.. Verslo reikalavimai surenkami, ir gaunamas vizijos dokumentas. Tuomet iškelti vartotojo reikalavimai aprašomi panaudos atvejais, bei kokybės atributais. Taigi, kaip matyti ir paveikslėlio 2 pav., reikalavimų specifikacijos dokumentas aprašomas funkciniais sistemos ir vartotojų reikalavimais, apribojimais, kokybės atributais, bei nefunkciniais reikalavimais.

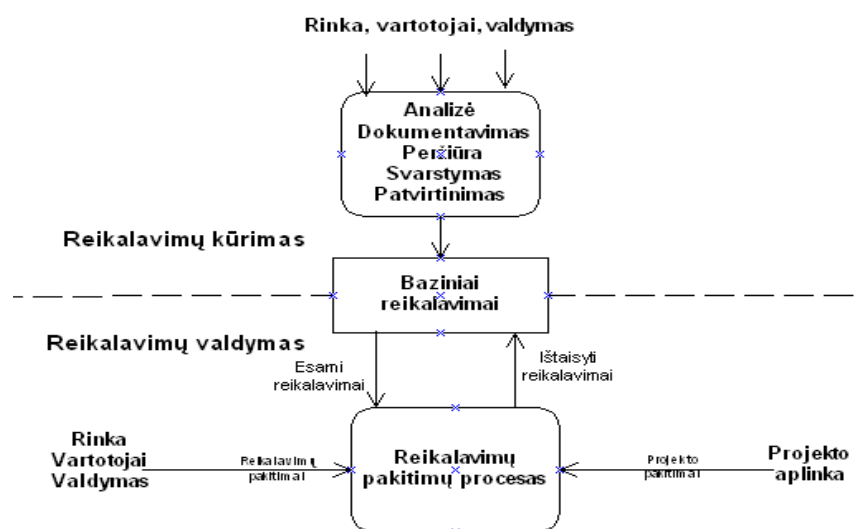


Šaltinis: Šilingas, D. (2005) Programinės įrangos kūrimo metodologija.

2 pav. Reikalavimų struktūra tarp kliento iki reikalavimų specifikacijos [11].

Reikalavimų specifikacijos charakteristikos:

- Vartotojai dažnai pirmąkart pamato sistemą tik tuomet, kai ji jau yra sukurta;
- Pilnumas - neturi būti trūkstamos informacijos;
- Suderinamumas - reikalavimai turi būti tarpusavyje suderinti ir neprieštarauti vienas kitam arba aukštesnio lygio sistemos reikalavimams;
- Keičiamumas - turi būti galimybė keisti reikalavimus ir sekti pakeitimų istoriją
- Atsekamumas - turi būti galimybė susieti reikalavimus tarpusavyje bei su projektavimo, realizavimo, testavimo planų elementais [3].



Šaltinis: autoriaus parengta pagal Weigers, K., E. (2003) Software Requirements, Second Edition.

3 pav. Ribos tarp reikalavimų kūrimo ir reikalavimų valdymo procesų.

Paveiksle 3 pav. pateikiamas skirtingas požiūris tarp reikalavimų vystymo ir reikalavimų valdymo. Iš čia matyti, kad reikalavimų vystymo procese yra analizuojami, surenkami, peržiūrimi ir aptariami reikalavimai iš vartotojų, užsakovų ar verslo srities. Suformuluoti reikalavimai, pereina į reikalavimų valdymo etapą, kuriame vykdomi jų pakitimai, atsiradę dėl projekto aplinkos pasikeitimo ir/ar reikalavimų pakitimų iš vartotojų, užsakovų, verslo srities (rinkos) pusės.

#### **1.4. Būdai specifikuoti programinės įrangos reikalavimus**

- Tekstiniai dokumentai, kurie naudoja aiškios struktūros ir tiksliai parašytą natūralią kalbą;
- Grafiniai modeliai, iliustruojantys pasikeitimų procesus, sistemos būsenas ir pasikeitimus tarp jų, duomenų sąryšius, logines veiksmų sekas, objektų klases ir jų ryšius;
- Formalios specifikacijos, apibrėžiančios reikalavimus naudojant tikslias formalias kalbas [1].

Programinės įrangos reikalavimų specifikacija tiksliai nustato funkcijas ir galimybes, kurias programinės įrangos sistema turi palaikyti, ir apribojimus, kuriuos turi atitikti. Specifikacija yra pagrindas visam tolesniam projekto planavimui, projektavimui, realizacijai, testavimui ir vartotojo dokumentacijos ruošimui, todėl ji turi aprašyti kiek įmanoma tiksliai išorinį, vartotojui matomą, sistemos funkcionalumą. Joje neturėtų būti projektavimo, realizacijos, testavimo ar projekto valdymo detalių, išskyrus numatytus projektavimo ir realizacijos apribojimus

## 2. REIKALAVIMŲ SPECIFIKAVIMAS ŠABLONAIŠ

Projektų reikalavimų specifیکavimui išskiriamos tokios reikalavimų specifیکavimo šablonų klasės:

- Universalūs reikalavimų specifیکavimo šablonai - turintys platų funkcionalumą, pritaikyti įvairioms sritims.
- Specializuoti reikalavimų specifیکavimo šablonai - turintys ribotą funkcionalumą, pritaikyti konkrečioms taikymo sritims.

Paprastai bent kiek sudėtingesnė programų sistema turi tenkinti kelis šimtus reikalavimų. Žmogus nėra pajėgus juos visus prisiminti ir operuoti jais savo galvoje. Be to, su reikalavimais dirba daugelis žmonių ir, nedokumentavus reikalavimų, negalima garantuoti, kad visi asmenys operuoja tuo pačiu reikalavimų rinkiniu. Tačiau ir užrašius reikalavimus raštu, išsispredžia toli gražu ne visos reikalavimų valdymo problemos. Visų pirma toks aprašas užima keliasdešimt puslapių. Todėl gali būti gana sudėtinga rasti jame reikiamą informaciją. Pavyzdžiui, kuomet norima pasitikslinti kaip skamba vienas ar kitas reikalavimas arba norint tą reikalavimą pakeisti. Būtų visiškai neprotinga reikalauti, kad reikiamos informacijos būtų ieškoma skaitant visą tekstą nuo pradžios iki galo. Be to, išmetus kokį nors reikalavimą visada, o kartais ir pridėjus naują reikalavimą, tekstą reikia perrašyti. Tai gali būti gana didelis darbas. Saugant reikalavimų specifیکaciją kompiuteriniu formatu (pvz. rtf), teksto perrašymo problema atkrita, nes tekstą galima keisti lokaliai. Tačiau reikiamos informacijos paieškos problema išlieka, nes tekstų redagavimo sistemos turi gana ribotas informacijos paieškos priemones. Be to, daugelis žmonių pageidauja dirbti su išspausdintu tekstu, o ne skaityti jį kompiuterio ekrane. Dar viena problema susijusi su reikalavimų specifیکacijos pakeitimais yra ta, kad po kurio laiko atsiranda kelios reikalavimų specifیکacijos versijos ir prireikia aiškintis, kuri specifیکacija yra naujesnė už kitą ir kuo jos skiriasi viena nuo kitos. Dėl šių ir daugelio kitų priežasčių reikalavimus reikia organizuoti specialiu būdu, t.y. taip, kad būtų patogu juos tvarkyti ir su jais dirbti visiems, kam to prireikia. Taigi, reikalavimų dokumentavimo problemos nėra tokios paprastos, kaip gali pasirodyti neįsigilinus į jų esmę [12].

Bendrasis sistemos aprašas kartais yra vadinamas vartotojo reikalavimais. Šis dokumentas aprašo bendrus sistemos reikalavimus, suformuluotus iš dalykinės srities specialistų požiūrio taško. Šį dokumentą skaitys žmonės, kurie ketina pirkti sistemą, ir žmonės, kurie ja naudosis. Todėl dokumentas turi būti parašytas bendra kalba, kuria kalba dalykinės srities specialistai, t.y. vartojant dalykinės srities

terminologiją. Šiame dokumente funkciniai sistemos reikalavimai aprašomi kartu su jos paskirtimi, jos naudojimo aplinka ir jos nefunkciniais reikalavimais. Dokumente gali būti pateikti sistemos veikimo kontekstą iliustruojantys konceptiniai modeliai, jos panaudojimo scenarijai, įvesties ir išvesties duomenų, principinių esybių bei darbo srautų aprašai [12].

Programinės įrangos inžinerijos standartai nustato praktikas, kurias laikytis turi programinės įrangos procesai. Standartai apskritai apibrėžia praktikas išreikštas apribojimais dokumentams. Standartas gali būti suprantamas kaip sutarimu parengtas dokumentas, kuris nustato bendram ir daugkartiniam naudojimui tinkančias taisykles, bendruosius principus ar charakteristikas ir yra skirtas įvesti optimalią tvarką tam tikroje srityje.

Yra daug skirtingų reikalavimų standartų, nustatančių reikalavimų specifikacijos struktūrą: JAV nacionaliniai standartai (American National Standards); IEEE standartas STD 830 - 1998; (EIA)/IEEE standartas 12207; nebegaliojantys standartai MIL-STD-2167A ir MILSTD-498.

Nors tie standartai nustato skirtingą reikalavimų specifikacijos struktūrą, tačiau visi jie reikalauja, kad specifikacijoje būtų pateikti tie patys pagrindiniai dalykai:

- sistemos vizija;
- sistemos sąsajų reikalavimai;
- funkciniai sistemos reikalavimai;
- nefunkciniai sistemos reikalavimai;
- našumo reikalavimai;
- projektiniai ribojimai;
- reikalaujamos sistemos kokybės atributų reikšmės [2, 4, 8, 10].

Kokį konkretų standartą yra geriau pasirinkti, iš esmės priklauso nuo to, kokie yra sistemos gyvavimo ciklo modelio reikalavimai ir koks konkretus programų sistemos kūrimo procesas yra pasirinktas tam modeliui įgyvendinti. Jeigu, tarkime, yra naudojama reikalavimų duomenų bazė ir visas dokumentas arba jo dalys yra generuojamos automatiškai, tai ir reikalavimų specifikacijos standartas turi atitikti generatoriaus galimybes. Pasirinkus bet kurį standartą, jį vis tiek dar reikia pritaikyti konkrečiau projekto poreikiams. Yra ir kitas požiūris, teigiantis, kad reikalavimų specifikacijos proceso

apskritai nereikia atlikti. Jį turėtų pakeisti bendras sistemos aprašas arba vartotojo vadovas. Šis požiūris pateisinamas atvejais, kada svarbiau yra kuo greičiau pateikti sistemą rinkai arba užsakovui, negu užtikrinti jos ilgalaikes jos priežiūros galimybes.

IEEE sukūrė IEEE 830-1998 standartą reikalavimų specifikacijos dokumentui. Kitas standartas yra Volere. Šie standartai apibrėžia reikalavimų specifikacijos dokumento išdėstymą. Šio etapo rezultatas – dokumentas, turintis visus reikiamus reikalavimus kuriamai sistemai. Viena svarbiausių veiklų yra reikalavimų užrašymas, ir reikalavimų specifikacijos dokumento sukūrimas. Panagrinėsiu detaliau kiekvieną iš minėtų reikalavimų specifikacijos standartų.

## **2.1. IEEE 830 -1998 standartas**

### 1. Projekto pristatymas

- 1.1. Tikslas
- 1.2. Dokumento formalumai
- 1.3. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys
- 1.4. Produkto uždaviniai
- 1.5. Nuorodos

### 2. Projekto apibūdinimas

- 2.1. Perspektyvos
- 2.2. Funkcijos
- 2.3. Vartotojų klasės ir jų charakteristikos
- 2.4. Eksploatavimo aplinka
- 2.5. Projektavimo ir diegimo apribojimai
- 2.6. Vartotojų dokumentacija
- 2.7. Prielaidos ir priklausomybės

### 3. Išorinės sąsajos reikalavimai

- 3.1. Vartotojo sąsajos
- 3.2. Techninės sąsajos
- 3.3. Programinės įrangos sąsajos
- 3.4. Bendradarbiavimo sąsajos



#### 4. Sistemos savybės

##### 4.1. Savybės apibūdinimas Nr.1

4.1.1. Aprašymas ir prioritetai

4.1.2. Įtakos elgesio sekos

4.1.3. Funkciniai reikalavimai

Reikalavimas Nr.1:

Reikalavimas Nr.2:

...

##### 4.2. Savybės apibūdinimas Nr.2

4.1.1. Aprašymas ir prioritetai

4.1.2. Įtakos elgesio sekos

4.1.3. Funkciniai reikalavimai

Reikalavimas Nr.1:

Reikalavimas Nr.2:

...

##### 4.3. Savybės apibūdinimas Nr.2 (ir t.t.)

#### 5. Nefunkciniai reikalavimai

5.1. Našumo reikalavimai

5.2. Saugumo reikalavimai

5.3. Apsaugos reikalavimai

5.4. Programinės įrangos kokybės atributai

5.5. Verslo taisyklės

#### 6. Kiti reikalavimai

Priedas A – Žodynas

Priedas B – Analizės modeliai

Priedas C – Baigiamųjų sąlygų sąrašas

Lentelėje (Lentelė Nr.1) pateikiamas IEEE standarto reikalavimų pakitimų ir versijų valdymo šablonas. Šio šablono pagalba vykdomas reikalavimų pakitimo kontrolė, sekamos pakitimų versijos.

Lentelė Nr. 1 Reikalavimų pakitimų ir versijų valdymo šablonas.

Pavadinimas	Data	Priežastys pakitimui	Versija

Šaltinis: autoriaus parengta pagal Requirements Specifications, IEEE Std 830-1998 (1998)

## 2.2. Volere šablonas

Kaip jau minėta anksčiau, daug lengviau ir patogiau surinkti reikalavimus, kai yra naudojami standartai, šablonai ar tam tikri įrankiai. Volere Requirements Specification yra šablonas, kuris apima pilną produkto funkcionalumo ir galimybių aprašymą. Pagrindinės Volere šablono dalys yra :

- Projekto varovai,
- Projektiniai apribojimai,
- Funkciniai reikalavimai,
- Nefunkciniai reikalavimai,
- Projektiniai klausimai

**Projekto varovai** – pagrindžiamas projekto kūrimo tikslas, apibrėžiama sistemos paskirtis ir tikslai, nustatomi suinteresuoti asmenys, bei numatomi būsimi sistemos vartotojai.

**Projektiniai apribojimai** identifikuoja, kaip galimas produktas turi atitikti realią aplinką (apribojimai sprendimui, diegimo aplinka, bendradarbiaujančios sistemos, svarbūs faktai ir prielaidos).

**Funkciniai reikalavimai** yra fundamentalūs veiksniai, įtakojantys sistemą. Šie veiksniai yra išmatuoti ir turi konkrečias reikšmės, įvertinimus, sprendimų priėmimo logiką ar algoritmus. Kitaip tariant aprašomi sistemos funkciniai reikalavimai nurodo kokias funkcijas turi atlikti kuriama sistema, ir kaip šios funkcijos turi būti atliekamos.

**Nefunkciniai reikalavimai** yra elgesio ypatybės, kurias turi turėti apibrėžtos funkcijos. Tai – paslaugų arba funkcijų apribojimai, kokybės atributai (patogumas, patikimumas, greitis, palaikomumas, saugumas), juridiniai bei kontrolės reikalavimai, palaikomos operacinės sistemos, standartai suderinamumas ir kt.

**Projektiniai klausimai** apibrėžia sąlygas, prie kurių projektas bus įgyvendintas. Jos įtraukiamos į reikalavimų specifikaciją, kad pateikti bendrą vaizdą, visų faktorių kurie gali įtakoti projekto pasisekimo ar nesėkmės atveju.

Šablonas suskirstytas į skyrius pagal reikalavimų tipą. Toliau pateiksiu detalią Volere šablono struktūrą, kurią ir aprašo minėtos Volere šablono dalys:

## **Projekto varovai (Project Drivers)**

### **1. Sistemos paskirtis**

Trumpas būsimos sistemos aprašymas, jos paskirtis.

#### **1.1. Projekto kūrimo pagrindas (pagrindimas)**

Pagrindžiamas poreikis būsimai sistemai, nusakomas jos reikalingumas.

#### **1.2. Sistemos tikslai**

Aprašoma pagrindinė esmė ko iš sistemos norima, ką ji turi atlikti. Tai realios priežastys dėl ko sistema bus kuriama.

### **2. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys**

Yra keletas asmenų tipų, kurie yra susiję su sistema. Šie asmenys identifikuojami šiame skyriuje.

#### **2.1 Užsakovas**

Užsakovas – tai sistemos savininkas, kuris užsako sistemą jos sukūrimui, apmoka jos kūrimo išlaidas. Jis priima galutinį produktą. Jeigu sistema kuriama asmeniniams tikslams, tuomet užsakovas ir pirkėjas dažniausiai būna tas pats asmuo. Jeigu sistema kuriama rinkai (išorės vartotojams), tuomet užsakovu gali būti marketingo skyrius, kurį atstovauja šio skyriaus darbuotojas

#### **2.2 Pirkėjas**

Pirkėjas – sistemą perkantis asmuo ar organizacija. Pirkėjo rolė yra nuspręsti ar perkama sistema yra reikalinga, verta ir atitinka jų išsikeltus poreikius.

#### **2.3 Kiti asmenys**

Tai kiti sistemos kūrimą įtakojantys asmenys, kurie gali būti:

- Vartotojai (detaliau apie juos 3 vartotojų dalyje)
- Sponsorai
- Techniniai specialistai
- Projektuotojai

- Marketingo specialistai
- Projekto vadovai
- Testuotojai
- Inspektoriai
- Teisininkai
- Panaudojimo specialistai

### **3. Vartotojai**

#### **3.1 Produkto vartotojai**

Tai potencialių asmenų sąrašas, kurie naudosis ir dirbs su sistema.

Kiekvienai vartotojų kategorijai reikia nurodyti:

- Vartotojo kategorija, pav., mokinys, inžinierius, projekto vadovas;
- Vartotojo tipas – tai dažniausiai priskiriamas vartotojų grupių tipas, pvz moksleivis, inžinierius, projektuotojas, projekto vadovas.
- Vartotojo rolė – aprašomos vartotojo pareigos ir priklausomybės.
- Technologinė patirtis – vartotojo patirtis dalykinėje srityje, pvz naujokas, pažengęs, patyręs, ekspertas.
- Kitos charakteristikos – aprašo vartotojo charakteristikas, kurios gali turėti įtaką projekto kūrimo fazėse.
  - fizinės galimybės / trūkumai
  - intelektualinės galimybės / trūkumai
  - požiūris į darbą
  - požiūris į technologijas
  - išsilavinimas
  - lingvistiniai įgūdžiai
  - amžiaus grupė
  - lytis.

#### **3.2 Vartotojų prioritetai.**

Rekomenduojama vartotojų grupėms suteikti prioritetus:

- Esminiai vartotojai – vartotojai, kritiškai įtakojantys tolimesnės sistemos kūrimo sėkmę. Rekomenduojama suteikti didesnę prioritetą reikalavimų sritį koordinuojantiems vartotojams.

- Antraeiliai vartotojai – produkto vartotojai, kurių nuomonė nėra svarbi ilgalaikio sistemos kūrimo atžvilgiu.
- Nesvarbūs vartotojai – šiai vartotojų kategorijai suteikiamas žemiausias svarbumo lygmuo, kuriam priklauso neautorizuoti, neturintys įgūdžių asmenys.

## **Projekto Apribojimai**

Tai išankstiniai apribojimai sistemos kūrimo eigai ar charakteristikoms. Jie turi nurodymo (griežto reikalavimo) pobūdį. Juos reikia tiksliai aprašyti, nepamirštant numatyti, kaip galima “išmatuoti” (testuoti) jų laikymąsi sukurtoje sistemoje. Čia galima apibūdinti ir tokio apribojimo atsiradimo priežastis.

Šiuos apribojimus dažniausiai pateikia užsakovas, pirkėjas arba vartotojas dėl įvairiausių priežasčių. Neįvertinus šių apribojimų, sistema gali būti nepriimta. Šie apribojimai apibrėžia tam tikrą galimų sprendimų kontekstą, todėl juos reikia registruoti itin kruopščiai. Negalima apriboti laisvės priimti geresnį sprendimą. Čia gali būti pateikiami tik neginčytini apribojimai.

### **4. Įpareigojantys apribojimai**

Šioje dalyje apibūdinami apribojantys reikalavimai galutiniam produktui.

#### **4.1. Sprendimų apribojimai**

Išankstiniai apribojimai sistemos kūrimo eigai ar charakteristikoms. Tai yra įpareigojantys apribojimai. Neįvertinus šių apribojimų sistema gali būti nepriimta.

#### **4.2. Diegimo aplinkos apribojimai**

Charakterizuojama technologinė ir fizinė aplinka, kurioje bus diegiama sistema. Šioje dalyje įtraukiami automatizavimo, mechanizavimo, organizaciniai ir kitokie įrenginiai, kurie atspindi su kuriama sistema susijusias sistemas. Sistemos įtraukiamos nevertinant žmogaus kriterijaus.

#### **4.3. Bendradarbiaujančios sistemos**

Aprašomos sistemos (ne kuriamos sistemos dalimis), su kuriomis kuriama sistema bendradarbiaus. Tai posistemės, taikomosios programos, komerciniai specializuoti programų paketai ar specialiai šiai sistemai sukurtos posistemės. Apibrėžiami integravimo uždaviniai ir problemos.

#### **4.4. Komerciniai specializuoti programų paketai**

Aprašomi komerciniai specializuoti programų paketai, reikalingi sistemos funkcionalumui užtikrinti. Reikia įvertinti šių paketų specifikas, sąsajas, prieštaravimus, kadangi jie turi įtaką formuojant reikalavimus kuriamai sistemai.

#### **4.5. Numatoma darbo vietos aplinka**

Aprašoma sistemos aplinka, kurioje dirbs vartotojas, identifikuojamos aplinkos charakteristikos, galinčios turėti įtaką sistemos kūrimui bei jos reikalavimams.

#### **4.6. Sistemos kūrimo terminai**

Identifikuojami laiko apribojimai, terminai. Reikia numatyti konkrečias datas ir terminus atskiroms sistemos dalims parengti ir užbaigti. Naudinga identifikuoti veiksmus kurie įvyks, jei nebus laiku atlikti darbai.

#### **4.7. Sistemos kūrimo biudžetas**

Sistemos kūrimo biudžetas išreikštas pinigais ar kitais galimais resursais. Reikalavimai negali viršyti biudžeto, todėl dalį reikalavimų gali tekti atmesti. Įvertinus numatomą biudžetą, galima identifikuoti, ar sistema tokį biudžetą atitinkanti tikrai bus reikalinga.

### **5. Terminų žodynas ir apibrėžimai**

Pateikiamas naudojamų reikalavimus specifikuoti terminų žodynas, Jame surašytos dalykinėje srityje naudojamos sąvokos. Kiekvienai sąvokai pateikiamas išsamus paaiškinimas. Sąvokos turi būti priimtinos dalykinės srities atstovams. Šio punkto tikslas išvengti nesusipratimų ir dviprasmybių, taip sutaupomas laikas ir projekto kūrimo kaštai.

Žodynas naudojamas ir vis pildomas visose sistemos kūrimo fazėse.

### **6. Svarbūs faktai ir prielaidos**

#### **6.1. Išoriniai faktai**

Faktai aprašantys dalykinę sritį, sistemas, veiklas ir veiksmus, kurie gali įtakoti produktą. Į šį punktą įtraukiami faktai, kurie nepatenka į kitus šablono punktus, bet yra numanoma, kad jie gali turėti įtaką reikalavimų specifikacijai.

#### **6.2. Prielaidos**

Prielaidų sąrašas, įtakojančios sistemos kūrimo aspektus. Sistemos kūrėjai pateikia prielaidas, kaip faktų priešingybes, kurios gali įvykti. Tai teisiniai ar politiniai sprendimai ir įstatymai, galintys prieštarauti kuriamoms sistemoms veikloms ar dalims.

## **Funkciniai reikalavimai**

### **7. Veiklos sudėtis / darbo apimtis**

#### **7.1. Veiklos kontekstas (pateikiama konteksto diagrama)**

Nagrinėjama veiklos sričiai apibrėžti naudojama konteksto diagrama, kuri identifikuoja darbus, kurie turi būti atlikti. Nepaisant to, kontekstas turi platesnes ribas, nei pati sistema, kadangi kuriama sistema bus kontekste, bei įtakojama aplinkos konteksto informacinių srautų.

## **7.2. Veiklos paskirstymas**

Identifikuojamas įvykių sąrašas, apimantis visus veiklos įvykius, su kuriais įvykiai susiję. Reakciją į kiekvieną veiklos įvykį atvaizduoja veiklos dalis, priklausanti pilnam veiklos funkcionalumui.

Įvykių sąrašas susideda:

- Įvykio pavadinimas;
- Įeinantys ir išeinantys informaciniai srautai..

Sudarant veiklos įvykių sąrašą yra būdas ištestuoti veiklos kontekstą. Taip išaiškinami kilę sistemos ar projekto netikslumai ir nesusipratimai, bei padeda nustatyti tikslus ryšius ir komunikacijas.

## **8. Sistemos sudėtis**

### **8.1. Sistemos ribos**

Sistemos riboms nustatyti naudojama panaudojimo atvejų diagrama. Jos pagalba nustatomos ribos tarp vartotojų ir sistemos.

### **8.2. Panaudojimo atvejų sąrašas**

Panaudojimo atvejų diagrama grafiniu būdu susistemina visus sistemos apimamus panaudojimo atvejus. Jeigu jų yra virš 15-20, tuomet geriau sudaryti panaudojimo atvejų sąrašą ir modeliuoti atskirais fragmentais. Kiekvienam sąrašo panaudojimo atvejui pateikiamas :

- panaudojimo atvejo numeris;
- vartotojo / aktoriaus pavadinimas;
- panaudojimo atvejo aprašymas;
- panaudojimo atvejo tikimo kriterijus.

Pateikiamas panaudojimo atvejo specializuota surinkimo forma, naudojama panaudojimo atvejų aprašui sudaryti 4 Pav..

PANAUDOJIMO ATVEJIS Nr. \_\_: Panaudojimo atvejo pavadinimas

**Vartotojas/Aktorius:** *Identifikuojamas aktorius*

**Aprašas:** *Aprašomas procesas, jo apimtys, ribos*

**Prieš sąlyga:** *Išvardijamos sąlygos, esančios prieš įvyksiant panaudojimo atvejui*

**Sužadinimo sąlyga:** *Aprašoma kas įvyks, sužadinus panaudojimo atvejį*

**Po-sąlyga:** *Įvykę pakeitimai sistemoje, po įvykdyto panaudojimo atvejo*

Šaltinis: autoriaus parengta pagal Robertson S., Robertson J. Volere Requirements Specification template (2006)

#### 4 Pav. Panaudojimo atvejų surinkimas.

## 9. Funkciniai reikalavimai ir reikalavimai duomenims

### 9.1. Funkciniai reikalavimai

Funkciniams ir nefunkciniams reikalavimams surinkti naudojama specializuota apklausos forma. Ši forma atstoja vieningą reikalavimo specifikavimo struktūrą ir gali būti pateikiama atskiroje kortelėje ar lape. Vėliau taip užregistruota reikalavimų specifikacija gali būti apdorojama ir kompiuterizuotu būdu. Kaip matyti iš paveikslėlio (5 pav.) Volere surinkimo formos struktūra išsamiai numato identifikuoti būsimo projekto kūrimui reikalingus reikalavimus.

<u>Reikalavimas #:</u>	...	<u>Reikalavimo tipas:</u>	...	<u>Įvykis/panaudojimo atvejis #:</u>	..
<u>Aprašymas:</u>		<i>Trumpas reikalavimo aprašymas</i>			.
<u>Pagrindimas:</u>		<i>Reikalavimo pagrindimas, nurodomos jo atsiradimo priežastys</i>			
<u>Šaltinis:</u>		<i>Reikalavimo autorius</i>			
<u>Tikimo kriterijus:</u>		<i>Reikalavimo matavimas, toks kad, yra galima ištestuoti, jei sprendimas atitinka originalų reikalavimą.</i>			
<u>Užsakovo tenkinimas:</u>	<i>Skaičius nuo [1-5]</i>		<u>Užsakovo netenkinimas:</u>	<i>[1-5]</i>	
<u>Priklausomybės:</u>		<i>Reikalavimai, turintys įtaką nagrinėjamam reikalavimui.</i>	<u>Konfliktai:</u>	<i>Kitų reikalavimų vykdymas negalimas jei egzistuos šis reikalavimas</i>	
<u>Papildoma medžiaga:</u>		<i>Nuoroda į dokumentą, kuris paaiškina, iliustruoja šį reikalavimą</i>			
<u>Istorija:</u>		<i>Užregistravimo, pakeitimo, ištrynimo data.</i>			



Šaltinis: autoriaus parengta pagal Robertson S., Robertson J. Volere Requirements Specification template (2006)

### 5 pav. Volere reikalavimų surinkimo forma.

Detaliau panagrinėsiu kiekvieną reikalavimų surinkimo formos elementą.

**Reikalavimo identifikavimas** - Reikalavimas identifikuojamas trimis parametrais:

- numeriu;
- tipu;
- įvykiu arba panaudojimo atvejo numeriu.

Reikalavimo numeravimas turi būti unikalus.

Suklasifikavus reikalavimus pagal tipus yra lengviau:

- nustatyti, prieštaraujančius reikalavimus;
- numatyti reikiamą tikimo kriterijų;
- identifikuoti reikalavimų pasikartojimą bei jų nepakankamumą

Reikalavimui priskiriami panaudojimo atvejai, nurodant panaudojimo atvejo numerį.

**Aprašymas** – tekstinis trumpas reikalavimo aprašymas, siekiant nustatyti reikalavimo paskirtį.

**Pagrindimas** – nusakomas reikalavimo pagrindimas, nurodant jo reikalingumą. Aprašomos reikalavimo atsiradimo priežastys, jų svarba ar įtaka sistemai.

**Šaltinis** - tai reikalavimo autorius (asmuo).

**Tinkamumo kriterijus** – toks reikalavimo išmatuojamumas, kad yra galima ištestuoti jei sprendimas tenkina ir atitinka reikalavimą.

**Užsakovo tenkinimas ir netenkinimas** – matuojamas įvertinimu - skaičiumi intervale [1-5]. Įvertis nurodo užsakovo patenkinimą reikalavimu, tuo atveju jei reikalavimas bus įgyvendintas sistemoje. Jeigu tenkinimas įvertintas 5 balų įverčiu – užsakovas bus labai patenkintas reikalavimo įgyvendinimo atveju, jeigu įvertis – 1, tuomet reikš kad užsakovas mažai susidomėjęs reikalavimo įgyvendinimu. Jei užsakovo netenkinimo įvertis 5 – jis bus labai nepatenkintas jeigu reikalavimas nebus įgyvendintas, ir jeigu 1 – užsakovas yra mažai susidomėjęs.

**Priklausomybės** – reikalavimo priklausomybė nuo kitų reikalavimų. Jeigu pasikeičia reikalavimas, tuomet keičiasi ir priklausomi reikalavimai, ir atvirkščiai.

**Konfliktai** – prieštaraujantys reikalavimai nagrinėjamam reikalavimui, kurių egzistavimas sukelia konfliktines situacijas ir prieštaravimus.

**Papildoma medžiaga** – pateikiama papildoma, betarpiškai susijusi su nagrinėjamu reikalavimu medžiaga.

**Istorija** – nustatomos su reikalavimu susijusios datos:

- įvedimo,
- modifikavimo,
- pašalinimo.

## **9.2. Reikalavimai duomenims**

Susijusių su sistema veiklos objektų, esybių, klasių specifikacija. Tai pradinis duomenų, objektų ar srities modelis, dažnai pateikiamas esybių – ryšių schema. Adekvačiai susijęs su terminų žodynu (5 punktas).

## **Nefunkciniai reikalavimai**

### **10. Reikalavimai sistemos išvaizdai (Look and feel)**

Šiais reikalavimais specifikuojama užsakovo produkto išvaizdos vizija.

#### **10.1 Vartotojo sąsaja**

Nustatomi reikalavimai vartotojo sąsajai ir jos ypatybėms. Užsakovas gali pateikti tam tikrus nurodymus, tokius kaip korporacijos ženklas/etiketė, stilius, spalvų gamos ir t.t. Šis skyrius užtikrina kad produkto išvaizda tenkintų užsakovus..

#### **10.2 Stilius**

Produkto stilius, nusakomas tuo, kokiai auditorijai yra planuojamas skirti. Jeigu tai verslo klasei, tuomet produkto stilius ir išvaizda turi būti konservatyvaus dizaino. Tačiau jei sistemos vartotojai bus vaikai, tuomet stilius bus spalvingas, judrus, žaismingas.

Reikalavimai nustatyti šiame etape, bus kaip gairės produkto dizaineriams, kuriant bendrą sistemos išvaizdą.

### **11. Reikalavimai panaudojamumui (Usability)**

#### **11.1 Panaudojimo paprastumas**

Apibūdina kliento norus kaip lengvai bus galima naudotis būsima sistema. Panaudojimo reikalavimai turėtų apimti:

- Naudojimo efektyvumas – kaip greitai ir tiksliai vartotojas gali atlikti užduotį;
- Lengva prisiminti – per kiek laiko atsitiktinis vartotojas gali įsiminti apie produkto naudojimąsi

- Klaidų įvertinimas – kai kuriems produktams yra labai svarbu kad vartotojas padarytu labai mažai klaidų, arba išvis jų nepadarytų.
- Bendras pasitenkinimas naudojant produktą – ypač svarbus faktorius komerciniams, interaktyviems produktams turintiems didelį konkurencingumą rinkoje.

## **11.2 Pritaikymas asmeniui ir nacionaliniai reikalavimai**

Aprašomi būdai, kurių pagalba produktas gali būti pakeistas ar konfigūruojamas pagal vartotojo norus. Norima užtikrinti, kad vartotojai neturėtų patirti diskomforto dėl kalbinių specifikų. Tai tokie reikalavimai, apimantys aspektus:

- Kalba, kalbinės idiomos;
- Valiutos ir jų simboliai;
- Asmeninės konfigūracijos pasirinkimas.

## **12. Našumo reikalavimai (Performance)**

### **12.1. Greičio ir vėlavimų reikalavimai**

Apibrėžia laiko kiekį, reikalingą įvykdyti apibrėžtas užduotis. Šie reikalavimai dažnai siejasi su atsakymo (responded) laiku. Taip pat siejami su produkto gebėjimu dirbti greičiu, tinkamu numatytai aplinkai.

### **12.2. Saugumo kritiniai reikalavimai**

Nustatomi pavojai žmonėms, turtui, aplinkai. Tikslas identifikuoti potencialius pavojus, įvyksiančius naudojantis produktu. Skirtingos šalys ar regionai, turi skirtingus standartus, todėl turi būti tiksliai apibrėžta kokius standartus turi atitikti produktas.

### **12.3. Tikslumo reikalavimai**

Nustatomi sistemos duomenų, rezultatų tikslumai., įvertinamas skaitmenų skaičius einantis po kablelio, bei tam tikrų matavimo prietaisų parodymai.

### **12.4. Patikimumo ir tinkamumo naudoti reikalavimai**

Nustatomas būtino produkto patikimumo kiekis. Patikimumas yra paprastai reiškiamas kaip leistinas laikas tarp nesėkmių, arba leistinas bendras nesėkmių skaičius. Tai taip pat nustatomas norimas tinkamumas - galimybė panaudoti, pvz. sistema turi būti tinkama naudoti 24val. per parą, 365 dienas per metus.

### **12.5. Stiprumo ir klaidos tolerancijos reikalavimai**

Nurodomas produkto veikimas atsitikus avarinei, neplanuotai situacijai aplinkoje. Pavyzdžiui sistema turi atlikti operacijas vietinėje aplinkoje, kuomet prarandamas ryšys su centriniu serveriu.

### **12.6. Talpumo reikalavimai**

Nustatomos apimtys, su kuriomis sistema turi gebėti dirbti. Tai ir duomenų laikmenų apimtys, taip pat vartotojų maksimalus kiekis, atliekamų operacijų kiekis ir pan.

## **13. Reikalavimai veikimo sąlygoms (Operational)**

### **13.1. Fizinės aplinkos reikalavimai**

Apibrėžiama fizinė aplinka, kurioje produktas veiks. Šie reikalavimai garantuoja, kad produktas yra tinkamas naudoti numatytoje aplinkoje.

### **13.2. Reikalavimai sąveikai su gretimomis sistemomis**

Reikalavimai sąveikai su gretimomis sistemomis (aplikacijomis, įrenginiais) kuriuos produktas turi sėkmingai funkcionuoti. Šie reikalavimai dažnai yra neaiškūs iki produkto diegimo fazės, todėl juos būtina identifikuoti anksčiau, kad sumažinti reikalavimų pakitimo tikimybę.

### **13.3. Produkcijos (productization) reikalavimai**

Identifikuojami bet kokie reikalavimai, kurie yra būtini siekiant sukurti perkamą, paklausų produktą. Numatomos vartotojo sąnaudos (laikas, pinigai, pastangos) įdiegiant produktą. Taip pat nustatoma produkto laikmena (CD, DVD ir t.t.). Taip pat tinkama apibūdinti operacijas, kurios bus įvykdytos, kad turėtų programinės įrangos produktą, sėkmingai įdiegtą.

### **13.4. Išleidimų (releases) reikalavimai**

Produkto išleidimų ciklas, ir jo forma. Numatoma kaip dažnai bus išleidžiami produkto patobulinimai.

## **14. Reikalavimai sistemos priežiūrai (Maintainability and portability)**

### **14.1. Palaikymo reikalavimai**

Nustatomas apibrėžtų produkto pakitimų atlikimo laikas.

### **14.2. Palaikymų (supportability) reikalavimai**

Apibrėžiami produkto palaikymo aspektai.

### **14.3. Pristaikymo reikalavimai**

Platformų ir aplinkų identifikavimas, kurioms bus pritaikytas kuriamas produktas.

## **15. Reikalavimai saugumui**

### **15.1. Naudojimosi teisės**

Nustatomi konfidencialumo aspektai. Nurodoma kas ir kokias teises turi naudojantis produktu. Kokiomis sąlygomis teisės yra suteikiamos, ir kurios produkto dalys yra leidžiamos naudotis.

### **15.2. Vientisumas**

Reikalaujamo duomenų bazės ir kitų produkto failų vientisumo specifikacija.

### **15.3. Privatumas**

Apibrėžiama kaip užtikrinti vartotojų duomenų atitinkantį teisinius įstatymus.

### **15.4 Imunitetas**

Apibrėžiamas kaip produktas turi būti apsaugotas nuo virusų ar išorinių įsibrovimų.

## **16. Kultūriniai-politiniai reikalavimai**

Apibrėžiami įtakojantys kultūriniai, socialiniai, politiniai faktoriai, galintys įtakoti produkto priimtinumą. Ypač svarbūs reikalavimai platinant įvairiuose šalyse ir regionuose, kuriuose išryškėja tokie faktoriai kaip religija, politinė ideologija, papročiai, kultūra ir kalbos ypatumai.

## **17. Teisiniai reikalavimai**

Apibrėžiami apribojimai susiję su įstatymais ir jų laikymosi. Apribojime nurodomas konkretus įstatymas ar standartas, kurio turi būti laikomasi.

## **Projektiniai klausimai**

### **18. Atviri klausimai (problemos)**

Svarstomos problemos, kurios buvo pakeltos ir dar neturi išvados. Tai faktoriai, kurie yra abejojantys ir galėtų turėti reikšmingą įtaką produktui.

## **19. Egzistuojantys sprendimai (Off-the-Shelf Solutions)**

### **19.1. Pagamintos sistemos, kurios gali būti nupirktos**

Sąrašas egzistuojančių produktų, kurie turi būti panaudoti kaip potencialūs sprendimai.

### **19.2. Pagaminti komponentai, kurie gali būti panaudoti**

Komponentų sąrašas, kurie gali būti panaudojami. Pateikiami ir pagaminti kompanijoje, ir nupirkti iš išorės komponentai.

### **19.3. Galimas pakartotinas panaudojimas**

Sąrašas kitų panašių produktų ar dalys produktų, kuriuos jūs galite teisiškai nukopijuoti ar lengvai modifikuoti savo poreikiams.

## **20. Naujos problemos**

### **20.1. Poveikis egzistuojančiai aplinkai**

Apibūdinamas poveikis egzistuojančiai aplinkai. Taip pat šiame skyriuje turi būti nurodoma kokio poveikio aplinka produktas neturi daryti.

### **20.2. Įtaka jau įdiegtoms sistemoms**

Apibūdinama įtaka, konfliktai su egzistuojančiomis įdiegtomis sistemomis.

### **20.3. Potencialių vartotojų reakcija**

Aprašoma neigiama vartotojų reakcija esamai sistemai. Gali būti, kad vartotojai patiria diskomfortą naudojantis tam tikromis produkto funkcijomis. Turi būti numatyti šių problemų išsprendimo būdai.

### **20.4. Kliudantys diegimo aplinkos apribojimai**

Aprašomos potencialios problemos, susijusios su naujų technologų diegimu ar naujais struktūros pakeitimais organizacijoje.

### **20.5. Tolimesnės problemos**

Situacijų identifikavimas, su kuriomis nebus įmanoma susidoroti. Siekiama apsaugoti prieš tokias situacijas, kurios atneštų produkto žlugimą.

## **21. Uždaviniai**

### **21.1. Projekto planavimas**

Aprašomi sistemos gyvavimo ciklo etapai, kuriais remiantis bus kuriamas produktas. Pateikiama aukšto lygio procesų diagrama uždaviniams bei jų sąsajoms nustatyti.

### **21.2. Planavimas ir kūrimo fazės**

Pateikiamas visų kūrimo fazių ir veikimo aplinkos komponentų aprašymas. Identifikuojamos būtinos naujos produkto veikimo aplinkos fazės, kad kūrimo procesą būtų galima valdyti. Naujo produkto fazių kūrimui identifikuojami reikalingi įrenginiai ir kitos sistemos.

## **22. Produkto pritaikymas**

### **22.1. Reikalavimai produkto pritaikymui**

Aprašomas pritaikymo veiksmų sąrašas, diegimo tvarkaraštis. Identifikuojami pritaikymo uždaviniai – kaip projekto planavimo proceso įėjimai.

### **22.2. Duomenų modifikavimas ir transformavimas naujai sistemai**

Aprašomas duomenų modifikavimo ir transformavimo sąrašas. Apibūdinamos esamos duomenų laikmenų technologijos. Aprašomos būsimos naujos duomenų laikmenų technologijos. Aprašomi uždaviniai duomenų modifikavimui ir transformavimui atlikti, bei šių veiksnių galimos problemos.

### **23. Rizikos**

Numatomas galimų iškilti rizikų sąrašas. Aprašomi įmanomi rizikų išvengimo būdai, bei identifikuojamas svarbiausių, turinčių didžiausią įtaką rizikų sąrašas, pavyzdžiui netiksli metrika, neadekvatūs skaičiavimai, įvairūs piktnaudžiavimai, netikslus kaštų apskaičiavimas ir įvertinimas, prasta produkto kokybė, darbo neproduktyvumas.

### **24. Kaina**

Kaina - pinigų ar pastangų kiekis, kurį turite išleisti produkto kūrimo. Kai tik reikalavimų specifikacija yra užbaigta, galima panaudoti vieną iš įvertinimo metodų, kad įvertinti kainą, išreiškiant rezultatą piniginiiais kiekiais ar laiku skirtu projektui sukurti. Skaičiavimai turi remtis materialiais ir suskaičiuojamais faktoriais. Volere šablono specifikacija numato daug išmatuojamų faktorių:

- Įėjimo/išėjimo srautų skaičius veiklos kontekste;
- Veiklos įvykiu skaičius;
- Panaudojimo atvejų skaičius;
- Funkcinių reikalavimų skaičius;
- Nefuncinių reikalavimų skaičius;
- Reikalavimų apribojimų skaičius.

### **25. Vartotojo dokumentacija ir apmokymai**

#### **25.1. Vartotojo dokumentacija**

Nustatomas vartotojų dokumentacijos sąrašas, kuri būtų tiekama kaip produkto sudedamoji dalis. Numatomi asmenys, kurie bus atsakingi už dokumentacijos kūrimą. Kuriant dokumentaciją reikia atsižvelgti dokumentacijos būsimus vartotojus - skaitytojus.

#### **25.2. Mokymai**

Reikalingų produkto vartotojams mokymų apibūdinimas. Apsvarstyti koks mokymas bus būtinas, kas suprojektuos mokymą, kas rūpinsis mokymo vykdymu.

### **26. Būsiami reikalavimai**

Aprašomi reikalavimai, kurių nebus šio leidimo versijoje, tačiau ateityje jie gali būti įtraukti būsimose produkto versijose.

## **27. Sprendimų idėjos.**

Aprašomos bet kokio sprendimo idėjos, kurios gali būti naudingos ateityje [2].

### **2.3. Reikalavimų pasikeitimas**

Turint omenyje šiuolaikinių IT projektų apimtis ir sudėtingumą, beprasmiška tikėtis, kad visi reikalavimai sistemai gali būti tiksliai apibrėžiami iš anksto. Reikalavimai kinta projektų eigoje, ir į tai būtina atsižvelgti. Kiek reikalavimai kinta priklauso nuo to, kaip gerai sistemos analitikai išsiaiškino reikalavimus prieš pradėdami realizuoti sistemą – jei gerai atliko reikalavimų rinkimo ir analizės darbus, tie pakitimai neturėtų būti dideli ir smarkiai įtakojantys sistemos realizavimą. Didelių reikalavimų pakitimų priežastys dažniausiai būna šios:

- Vartotojai dažnai pirmą kartą pamato sistemą tik tuomet, kai ji jau yra sukurta;
- Kai kurių sistemos dalių veikimas būna pagrįstas daugiau programuotojų fantazija nei jos vartotojų įsivaizdavimu.

Reikalavimų pasikeitimą galima sumažinti taikant UML (Unified Modeling Language) programinės įrangos kūrimo modeliavimo kalbą. Jau nagrinėtu Volere šablonu identifikuotus metodus, specifikuoti naudoti UML [1].

### **2.4. Reikalavimų specifikavimas naudojant UML**

Reikalavimų specifikavimas naudojant UML (Unified Modeling Language – Vieninga modeliavimo kalba) teikia daug galimybių sistemos projektavime ir užtikrina išsamų reikalavimų specifikacijos sudarymą, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus. Paketas sudarytas iš specifikavimo diagramų:

- Veiklos diagrama - modeliuoja dinaminę sistemos elgseną (vaizduojami veiksmai);
- Panaudos atvejų diagrama - apibūdina funkcinį sistemos veikimą vartotojo požiūriu;
- Sekos diagrama - apibūdina dinaminę veikėjų (aktorių), sistemos objektų ir sistemos sąveiką;



- Bendradarbiavimo diagrama - apibūdina pranešimus, siunčiamus tarp komponentų;
- Klasijų diagrama - apibūdina statinę sistemos struktūrą: objektus, atributus, asociacijas;
- Būsenų diagrama - apibūdina vieno sistemos objekto dinaminį elgesį kaip būsenų kaitą;
- Komponentų diagrama - aprašo sistemoje naudojamus komponentus;
- Išdėstymo diagrama - aprašo fizinį sistemos diegimą [1, 10, 12].

Siekiant tiksliau specifikuoti surinktus reikalavimus šiomis UML išvardintomis diagramomis galima aprašyti įvairiausias dalykinės srities charakteristikas bei vartotojo reikalavimų niuansus. UML kalbos priemonėmis galima adekvačiai specifikuoti daugumą dalykinės srities semantinių aspektų. Pateikti grafiniai vaizdai (diagramos) duoda daug didesnę naudą analizuojant, specifikuojant reikalavimus, mažesnė tikimybė susidaryti dviprasmybėms.

### **3. REIKALAVIMŲ INŽINERIJOS PROGRAMINIAI ĮRANKIAI**

#### **3.1. Programinių įrankių samprata, problemos, galimi sprendimai**

Reikalavimų inžinerijos įrankiai yra programiniai įrankiai, automatizuojantys procesus reikalavimų inžinerijos etapuose. Programinis įrankis turi palaikyti visą reikalavimų inžinerijos procesą: identifikaciją, analizę, aptarimą ir patvirtinimą. Reikalavimų valdymas yra dalis reikalavimų inžinerijos proceso, tad dažniausiai reikalavimų inžinerijos įrankiai yra vadinami reikalavimų valdymo įrankiais. Funkcionalumas šių įrankių persidengia su reikalavimų inžinerijos veiklomis, tai ne tik projekto pakitimų valdymas, bet ir reikalavimų išsiaiškinimas, analizė, aptarimas ir patvirtinimas.

Šiuolaikinės informacinės sistemos negali būti įgyvendinamos be minėtų įrankių pagalbos. Daugelis įrankių yra aprašomi kaip CASE įrankiai, iš kurių laukiamas darbo palengvėjimas, efektyvumo ir našumo padidėjimas projekto analitinėje, projektavimo ir diegimo fazėse.

Vienas iš pagrindinių reikalavimų inžinerijos įrankių minusų yra vienišumas, kai netiekiamas joks galimas bendras darbas. Problema yra bendradarbiavimo įrankių stygius. Nei vienas iš įrankių nėra pilnai tinkamas daugelyje disciplinų įdiegtose sistemose, paskirtose komandose, kur suinteresuoti asmenys turi įvairiapusių įgūdžius. Bendradarbiavimo darbas yra ypatingai svarbus projekto pirminėse fazėse - reikalavimų valdyme. Geografiškai pasiskirsčiusios komandos darbui galimybės sutaupyti laiką ir finansinius resursus. Reikalavimų įrankiuose jaučiasi trūkumas pakartotinio panaudojimo galimybių ir funkcionalumo. Kai kuriais atvejais įrankiai turi galimybę sukurti asocijacias tarp skirtingų projektų, bet iš bendro praktinio pakartotino panaudojimo požiūrio, tai atliekama pasinaudojus „copy-paste“ funkcijomis. Pakartotinio panaudojimo problemos sprendimas galėtų būti bendra duomenų bazė, naudojama tos pačios šeimos produktų siekiant pagerinti reikalavimų inžinerijos procesą. Bendra duomenų saugykla reikalavimams ir jų sąryšiams būtų naudingas ir nuoseklus pakitimų integravimasis į produkto veiklos sritį.

Reikalavimų specifikuojimo įrankiai yra sukurti kvalifikuotų specialistų, todėl sudėtingas įrankio funkcionalumas yra nepatogus projekto užsakovams, kurie nėra kvalifikuoti IT specialistai.

### **3.2. Reikalavimų inžinerijos įrankių apžvalga**

Apžvelgiami reikalavimų inžinerijos įrankių esminiai privalumai ir funkcionalumas, remiantis įrankių dokumentacijomis, ir apžvalgomis. Daugiausiai naudota Volere, Incose ( International Council on Systems Engineering) ir PRQA (Program Research) apžvalgų medžiaga.

#### **Analyst Pro (Goda Software, Inc.)**

Analyst Pro – reikalavimų valdymo, sąryšių nustatymo ir analizės įrankis. Šis įrankis naudoja konfigūracijos valdymo metodologijas, leidžiančias projektą kuriančiam personalui analizuoti reikalavimų pakitimų įtaką bei komponentų vertybes. Įrankyje įtrauktos reikalavimų importavimo, reikalavimų dalyvavimo įtakos, pakitimų valdymo, priskyrimo ir reikalavimų diagramų funkcijas.

#### **CaliberRM (Borland Software Inc)**

CaliberRM palengvina bendradarbiavimą tarp projekto komandų tiekdamas centralizuotų reikalavimų duomenis komandos nariams, sudarydamas sąlygas dokumentuoti diskusijas apie reikalavimus, taip pat projekto komandoms pilnai apibrėžti, valdyti ir komunikuoti keičiantis sistemos reikalavimams. Reikalavimų pakitimai yra įrašomi į įrankio centrinę saugyklą. Caliber RM informuoja komandos narius, automatiškai pranešdamas atsakingiems asmenims apie atliktus reikalavimų pakitimus. Įrankis taip pat suteikia galimybę komandos nariams identifikuoti potencialias reikalavimų problemas akcentuojant dviprasmiškus ir dažniausiai vartojamus terminus, aprašytus žodyne.

#### **Catalyze (SteelTrace)**

Catalyze įrankis apima struktūrizuotą požiūrį skirstant reikalavimus į funkcinis ir nefunkcinis reikalavimus. Catalyze automatiškai generuoja srautų diagramas iš teksto. SteelTrace kompanija siūlo Catalyze modelių integraciją su UML modeliavimo įrankiais, tokiais kaip Rational Rose ir Borland kompanijos produktais Together Solo ir Control Centre. Sąryšis su Microsoft Word profiliu suteikia galimybę ta patį Catalyze projektą generuoti į skirtingus duomenų formatus.

### **CORE (Vitech Corporation)**

CORE įrankis suteikia galimybę ištraukti reikalavimus iš šaltinių dokumentacijos, analizuoti jų pilnumą, nuoseklumą ir sąryšius, ir surišti kiekvieną reikalavimą į elgsenos modelį, kuris aprašo sąveikas ir procesų sekas. Sistemos elgsena yra vaizduojama vartotojo-atrinkimo grafiniais modeliais, kurie fiksuoja sistemos kontrolės logiką ir duomenų srautus į integruotas elgsenas. Vartotojas paskirsto sistemos funkcinis modelius į sistemos fizinę architektūrą. CORE turi tikrinimo ir validavimo galimybes, kur vykdomi ir testuojami modeliai, siekiant atlikti sistemos charakteristikas ir resursų naudojimą.

### **Cradle (3SL - Structured Software Systems Ltd.)**

Cradle teikia reikalavimų surinkimo galimybes, kas nustato vartotojų reikalavimus ir išplečia reikalavimus, prielaidas ir/ar srities žinias, kuriant mišrias nuorodas į pradinį dokumentą. Kai naujosios versijos tokio dokumento yra užregistruojamos, Cradle palygina ir suranda skirtumus bei apskaičiuoja poveikio įvertinimą. Cradle pilnai integruotas su Word, Excel, PDF bei kitais tekstiniais formatais. Reikalavimai gali būti susieti su plačią įvairove UML standarto panaudos atvejo, funkciniu, elgsenos, dinaminio ir architektūriniu modeliais. Tai pat reikalavimus galima suklasifikuoti į skirtingų modelių sritis. Reikalavimus galima priskirti panaudos atvejams, funkcijoms, verslo procesams, vykdymo sekoms, visa tai paskirstoma funkcijoms, klasėms ir t.t. su komponentais daugialypėse architektūrose. Jose pilnai palaikomas atlikimo įvertinimas, biudžeto grupavimas ir paskirstymas, taip pat priemonės naudoti šiuos modelius aparatūrinės ir programinės įrangos kūrimo, įtraukiant kodo generavimą ir atgalinę inžineriją (reverse engineering).

### **DOORS (Telelogic Inc.)**

DOORS reikalavimus traktuoja kaip askirus objektus. Kiekvienas reikalavimas gali būti susietas su neribojamu bruožų kiekiu, leidžiantis lengvą reikalavimų pogrupių atrinkimą. DOORS įrankis turi on-line režimu veikiančią pakitimų pasiūlymų ir peržiūros sistemą, kuri suteikia galimybę vartotojams patvirtinti siūlomus reikalavimų pakitimus, įskaitant ir pateisinančias aplinkybes (justification). DOORS siūlo neribotus ryšius tarp visų projekto objektų pilnam daugialypio lygmens (multi-level) atsekamumui (traceability). Visose projekto gyvenimo ciklo fazėse yra prieinamos poveikio ir atsekamumo ataskaitos, taip pat kaip ir pranešimai apie identifikuotus trūkstamus sąryšius.

## **EasyRM (Cybernetic Intelligence GmbH)**

EasyRM – CASE įrankis apimantis pirmines programinės įrangos projektų kūrimo fazes, turintis reikalavimų žodyną bei reikalavimų ir nurodymų dokumentavimo valdymą. EasyRM suteikia vartotojams šias galimybes: reikalavimų kūrimas, apibūdinimas, modifikacija ir progreso stebėjimas, reikalavimų klasifikavimas, jų tarpusavio ryšių specifikuojimas, semantinių sąsajų iš reikalavimų žodyną peržiūra, atsekamumo ryšių tarp reikalavimų ir pradinių duomenų šaltinių, iš kurių kilo reikalavimas, palaikymas.

### **Focal Point (Focal Point )**

Focal Point – internetu paremta platforma produktų kūrimui. Įrankio stiprioji pusė – bendradarbiavimas tarp kūrėjų ir užsakovų siekiant surinkti reikalavimus. Įrankis palaiko spragų analizę, konkurencijos analizę su rinkos tikslais ir apibrėžimais. Taip pat įrankyje realizuotas reikalavimams prioritetų priskyrimas, galutinio projekto pridavimo ir resursų planavimo funkcijos.

### **Gatherspace (GatherSpace)**

On-line režimu veikiantis reikalavimų valdymo ir panaudos atvejų kūrimo įrankis centralizuotiems, modeliuojamiems ir bendrai naudojamiems reikalavimams valdyti.

### **IRqA (TCP Sistemas & Ingenieria)**

IRqA – programinės įrangos kūrimo ankstyvosiose gyvenimo ciklo fazėse pagalbininkas. IRqA turi reikalavimų išsiaiškinimą, analizę, specifikuojimą ir valdymą. Reikalavimai gali būti klasifikuojami remiantis kriterijais tokias kaip prioritetas, tipas, būseną, ar kiti valdymo aspektų kriterijai. Reikalavimų analizė ir tobulinimas yra atliekami kuriant probleminės srities modelį, kuris leidžia vartotojams pristatyti sąvokas, bendras idėjas ir koncepcijas tam tikroje srityje. Palaikomi dviejų rūšių modeliai: objektiškai orientuotus ir veiklos sąryšio modelius.

### **Rational RequisitePro (IBM)**

Rational RequisitePro integruotas su Microsoft Word ir reikalavimų duomenų baze. Programinės įrangos kūrėjai gali surinkti, įvesti ir valdyti reikalavimus duomenų bazėje ar savo

dokumentuose. Automatizuotas atsekamumas suskirsto reikalavimus ir jų pakitimus realizacijos ir testavimo etapuose. Susiję reikalavimai gali būti susieti kartu, kad įvykus vieno reikalavimo pakitimams, vartotojui būtų nesudėtinga nustatyti poveikį kitiems susijusiems reikalavimams. RequisitePro turi galimybes organizuoti, prioritetizuoti ir nubraižyti ryšius tarp reikalavimų.

### **Vital Link (Compliance Automation, Inc.)**

Vital Link – reikalavimų valdymo įrankis su integruotu word procesoriumi ir reliacine duomenų baze. Vital Link naudoja Adobe FrameMaker, kuris suteikia galimybes pakartotiniam panaudojimui, grafikų ir lentelių kūrimui, sudėtingų matematinių formulių rašymui. Į duomenų bazę galima importuoti egzistuojančius dokumentus iš didelės įvairovės word procesorių, ir automatiškai nagrinėjamas dokumentas, kuris kaip mat bus paruoštas redaguoti vartotojui, susieti veiklas, pridėti atributus ar generuoti ataskaitas. Įrankis turi atsekamumo tarp reikalavimų skirtinguose specifikacijos lygmenyse funkciją.

Apžvelgti reikalavimų inžinerijos programiniai įrankiai ir aptartos kiekvieno įrankio pagrindinės funkcijos, privalumai, išskirtinumas. Toliau detaliau panagrinėsiu, analizuosiu ir tarpusavyje palyginsiu reikalavimų inžinerijos įrankius ir jų specifikas.

### **3.3. Reikalavimų specifikavimo įrankių analizė**

Toliau lentelėje (Lentelė Nr.2) pateikiama detali įrankių analizė. Pasirinkti vieni didžiausių ir geriausių reikalavimų specifikavimo Requisite Pro, IRqA, Gatherspace, Doors ir Caliber RM, įrankiai. Pasirinkimo kriterijai buvo pradinių reikalavimų specifikacijos fazių, tokių kaip išsiaiškinimas, analizė, aptarimas ir patvirtinimas, egzistavimas. Lentelėje pateikti šių reikalavimo specifikavimo įrankių įvertinimai, atrinkti iš įrankių gamintojų dokumentacijų, bei IncoSe, PRQA ir Volere apžvalgų.

**Lentelė Nr. 2 Reikalavimų inžinerijos įrankių analizė**

	<b>Requisite Pro</b>	<b>IRqA</b>	<b>Gatherspace</b>	<b>Doors</b>	<b>Caliber RM</b>
1. Užfiksuoti reikalavimai/ identifikacija	+	+	+	+	+
1.1. Įvesties duomenų dokumento analizė	+	+	+	+	+
1.1.1. Įvesties duomenų dokumento pakitimai/lyginamoji analizė	+	+	/	+	/

	Requisite Pro	IRqA	Gatherspace	Doors	Caliber RM
1.2. Automatinė reikalavimų analizė	+	+	+	+	+
1.3. Interaktyvi/pusiau automatizuota reikalavimų indentifikacija	+	+	/	+	+
1.4. Reikalavimų indentifikacija rankiniu būdu	+	+	+	+	+
1.5. Reikalavimų indentifikavimas iš įrankio išorės	+	+	+	+	+
1.6. Reikalavimų klasifikavimas	+	+	+	+	+
2. Sistemos elementų struktūros nustatymas (sistemos elementų tam, kad paskirstymas ir ryšiai būtų nustatyti)	+	+	+	+	+
2.1. Grafinis sistemos struktūros nustatymas	+	+	+	+	+
2.2. Tekstinis sistemos struktūros nustatymas	+	+	-	+	+
3. Reikalavimų nutekėjimas/paskirstymas sistemos elementams	+	+	/	+	+
3.1. Reikalavimų kilmės nustatymas (galimybė sukurti papildomus reikalavimus ir sąryšį tarp jų kaip) reikalavimas – reikalavimas, reikalavimas – analizė/ tekstas)	+	+	+	+	+
3.2. Reikalavimų veiksmų paskirstymas sistemos elementams (svoris, kaštai, rizika)	+	+	+	+	+
3.3. Dvikryptis reikalavimas susietas su sistemos elementais	+	+	+	+	+
3.4. Atskaitomybės, testavimo, patvirtinimo, kritinių būsenų nustatymas. Jei yra, kaip ir ką mechanizmas atlieka.	+	+	-	+	+
4. Atsekamumo analizė	+	+	+	+	+
4.1. Identifikuoti nenuoseklumą (nesusietus reikalavimus ar elementus)	+	+	+	+	+
4.2. Ryšių matomumas (nuo šaltinių iki realizavimo) t.y. sekti sąsajas	+	+	+	+	+
4.3. Reikalavimų patikrinimas (kas ir kaip buvo padaryta)	+	+	-	+	+
4.4. Reikalavimų įvykdymo patikrinimas iš sistemos elementų (išvystymo iš faktų iki sistemos elementų) pusės	/	/	+	+	+

	Requisite Pro	IRqA	Gatherspace	Doors	Caliber RM
5. Konfigurācijas pakītimai	+	+	-	+	+
5.1. Reikalavimū pakītimū istorija (kas, kā, kada, kur kodēl, kaip)	+	+	/	+	+
5.2. Išeities taškū / versijū valdymas	+	+	-	+	+
5.3. Priējimo valdymas ir teisiū nustatymas	/	+		+	+
6. Dokumentai ir kiti išvesties duomenys	+	+	+	+	+
6.1. Palaikomi išvesties specifikācijas standartai (jei yra, kokio tipo)	/	+	-	+	+
6.2. Kokybēs ir klaidū tikrinimas (rašybos ir gramatikos tikrinimas, duomenū žodynai, ... )	/	/	-	/	+
6.3. Prezentacijū generavimas, kokybēs grafikai, lentelēs.	/	+	-	+	+
6.4. Pasirinktū požymiū ar įvertinimū išvedimas (apibrēzimo lentelēs, saugumo ženklai)	+	+	-	+	/
6.5. WYSIWYG peržiūra galutinio išėjimo formatu	+	/	-	+	+
6.6. Esamos padēties ataskaitos	+	+	-	+	+
6.6.1. Techninė vykdymo būklē	+	+	+	+	+
6.6.2. Reikalavimū progreso būsenos ataskaita	+	+	-	+	+
6.6.3. Kiti specialūs klausimai	+	+	-	+	+
7. Grupinio darbo galimybēs	+	+	-	+	/
7.1. Reikalavimū ir vykdymo peržiūra, pažymėjimas ir komentavimas	+	+	-	+	+
7.2. Skirtingū vartotojo sąsajū palaikymas, prieigū ribojimas	+	+	+	+	+
8. Sąsaja su kitais įrankiais	+	+	-	+	+
8.1. Komunikavimas su kitais įrankiais	+	+	-	+	+
8.1.1. Su kokiais įrankiais komunikuojasi	+	+	-	+	+
8.1.2. Komunikavimasis su išorine programine įranga ar duomenū bazėmis	+	+	-	+	+
8.1.3. Palaiko atviros duomenū bazēs sistemą (standartinē užklausū kalba, formatai)	+	+	+	+	/
8.1.4. Duomenū importavimas iš įvairiū failū formatū	+	+	-	+	+
8.1.5. Palaiko duomenū apsikeitimo standartus (AP-233, XML,...)	/	+	+	/	/
8.2. Vidinės įrankio komunikacijos	+	+	-	+	/



	Requisite Pro	IRqA	Gatherspace	Doors	Caliber RM
8.2.1. Informacijos keitimasis tarp įrankio skirtingų instaliacijų ar duomenų bazių.	+	+	-	+	/
8.2.2. Suderinamumas tarp įrankio duomenų rinkinių	+	+	-	+	+
9. Sistemos aplinka	+	+	-	+	+
9.1. Vieno/daug vartotojų darbas sistemoje	+	+	-	+	+
9.2. Platformos ir operacinės sistemos	/	+	+	+	+
9.3. Komeracinė ar firminė duomenų bazė	+	+	+	+	+
9.4. Resursų reikalavimai (programinei/techninei įrangai)	+	+	-	+	+
9.4.1 Reikalavimai atminčiai	+	+	+	+	+
9.4.2. Reikalavimai procesoriui	+	+	/	+	+
9.4.3. Reikalavimai talpai diske	+	+	-	+	+
10. Vartotojo sąsajos	+	+	+	+	+
10.1. Vienu metu atlikti kelis veiksmus	+	+	-	+	/
10.2. Atnaujinimas visuose atidarytuose languose	+	/	-	+	/
10.3. Grafinių duomenų įėjimas ir kontrolė	+	+	-	+	+
10.4. Palaikomas langų standartas	+	+	-	+	+
10.5 Vykdomieji skriptai ir makrokomandos	/	+	-	/	+
10.6 Naršyklės sąsaja	+	+	+	+	+
10.7. Redagavimo, atšaukimo (Undo) funkcijų palaikymas	/	/	-	/	+
11. Kompanijos naudojami standartai	+	+	+	+	+
12. Pagalba ir palaikymas	+	+	-	+	+
12.1. Garantija	+	+	/	+	-
12.2. Licencijavimo politika	+	+	+	+	+
12.3. Palaikymas ir atnaujinimai	+	+	+	+	+
12.4. Pagalba on-line režimu	+	+	-	+	+
12.5. Prieiga internetu	+	+	/	+	+
12.6. Pagalba telefonu	+	+	+	+	+
12.7. Vartotojų grupių parama	+	+	+	+	+
13. Apmokymai	+	+	-	+	+
13.1. Įrankio specifikos apmokymo kursai	+	+	-	+	+

	<b>Requisite Pro</b>	<b>IRqA</b>	<b>Gatherspace</b>	<b>Doors</b>	<b>Caliber RM</b>
13.2. Apmokymas kliento valdose	+	+	/	+	+
13.3. Rekomenduojamas apmokymų laikas	+	+	-	+	+
13.4. Programinės įrangos instaliavimas tik su baziniu apmokymu	+	+	-	+	+

Lentelėje atvaizduotos reikšmės:

„+“ pilnai išpildomas kriterijus,

„/“, iš dalies išpildomas kriterijus,

„-“, šis kriterijus programiniame įrankyje nėra įtrauktas.

Atlikus detalią reikalavimo inžinerijos įrankių analizę paaiškėjo, kad pasirinkti produktai ne taip stipriai skiriasi tarpusavyje, išskyrus Gatherspace, kuris į atranką buvo įtrauktas tik todėl, kad vykdomas On-line režimu. Reikalavimų inžinerijai skirti IBM Rational RequisitePro ir Telelogic Doors, TCP Sistemas & Ingenieria IRqA ir Borland CaliberRM produktai yra geri ir funkcionalūs, bei kaip matosi iš analizės lentelės, skiriasi nedideliu kriterijų skaičiumi. Stipriausias iš analizuotų įrankių yra Telelogic kompanijos produktas Doors, kuriame įgyvendinta daugiausiai analizėje pateiktų kriterijų. Analizės lentelėje duomenys pateikiami tik iš teorinių duomenų, pateikiamų gamintojų interneto svetainėse ar produktų dokumentacijose ir iš egzistuojančių reikalavimų inžinerijos įrankių apžvalgų Incose, Volere, PRQA. Apmaudu, kad ne visus šiuos programinius įrankius teko išbandyti iš praktinės pusės, arba buvo galimybė išbandyti tik nepilną įrankio bandomąją (trial / demo) versiją.

Yra gana sudėtinga įvertinti programinius įrankius atsižvelgiant į jų poveikį organizacijos procesams, kadangi vis dar retai reikalavimų inžinerijos įrankiai yra naudojami praktikoje. Kita vertus yra sudėtinga išegzaminuoti ir įvertinti įrankius eksperimentais, kadangi sunku susekti gamintojo suteiktų įrankiui galimybių įvairovę tam tikro projekto kontekste, net nekalbant apie įrankių analizę tik iš gamintojo pateikiamų dokumentacijų, ir egzistuojančių reikalavimų inžinerijos įrankių apžvalgų ir riboto funkcionalumo (demo/trail) programinių įrankių versijomis, kaip kad ir buvo atlikta šiame darbe.

Nepaisant šių reikalavimų inžinerijos programinių įrankių įvertinimo sudėtingumo, be abejonės galima teigti, jog dideliuose projektuose reikalavimų specifikavimo įrankiai suteikia daug našesnę ir produktyvesnę darbą su reikalavimais. Taip yra todėl kad, sunkiau numatyti iš anksto reikalavimus projektui, dažnai keičiasi užsakovų pateikti reikalavimai.

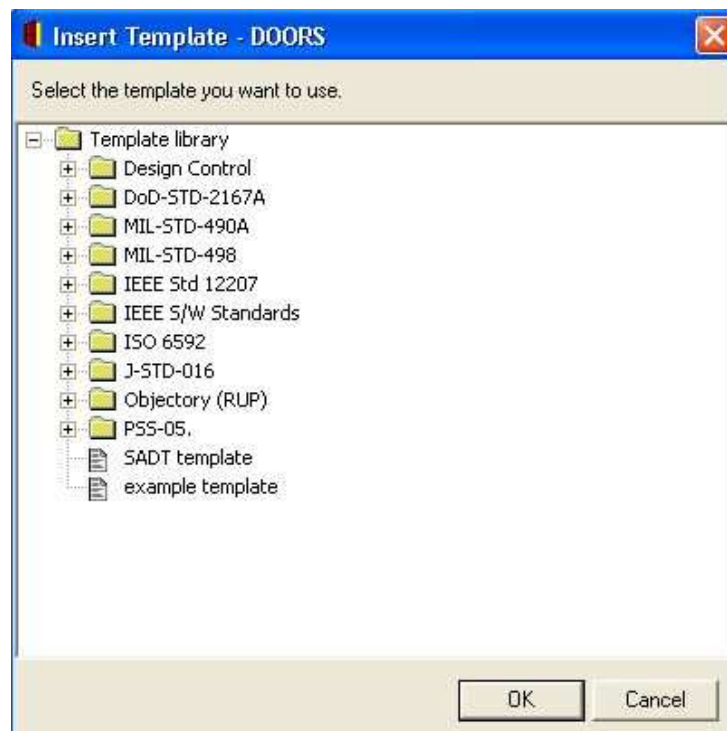
Pakankamai sudėtinga sukurti ir atlikti testus įrankiams, kad gauti pilnai realistiškus rezultatus. Be to, įrankių įvertinimo išlaidos gali būti nenaudingos vidutinio ar mažo dydžio kompanijoms, kurios labiau ieško pigesnio produkto.

## 4. REIKALAVIMŲ SPECIFIKAVIMO MODELIS

### 4.1 Volere šablono modelis

Remiantis Volere reikalavimo specifikavimo standartu, ir reikalavimų inžinerijos įrankių analizės rezultatais, aprašytais ankstesnėse darbo dalyse, pasirinktam reikalavimų inžinerijos įrankiui Doors sumodeliuosime Volere reikalavimų specifikavimo šablono pritaikymą.

Šablonai padeda išlaikyti kuriamų modulių nuoseklumą, skatina vartotojus naudoti standartinę struktūrą. Remiantis reikalavimų specifikavimo analizės rezultatais, pasirinktas Volere šablonas reikalavimams aprašyti, kuriame vadovaujamosi susisteminto proceso apribojimais, nusakančiais, kokio etapo metu ir kokie reikalavimai turi būti aprašyti.



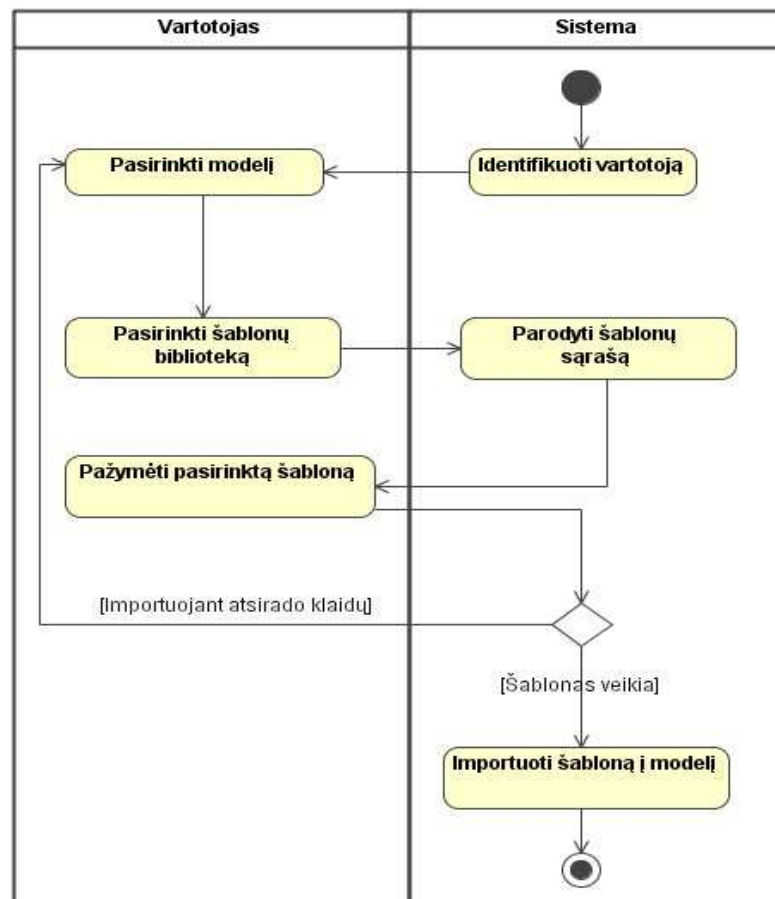
Šaltinis: Morkevičius, A.(2008)

**6 pav. Doors šablonų biblioteka.**

Doors įrankis turi šablonų biblioteką, kurios pagalba vartotojui yra sukuriama galimybė pasinaudoti jau esančiais šablonais (Projekto valdymo, DoD-STD, MIL-STD, IEEE, ISO, J-STD, PSS). Naudojant šablonus, pagreitinamas reikalavimų analitikų darbas kuriant reikalavimų specifikaciją, taip pat išlaikomas reikalavimų specifikacijos nuoseklumas, vieninga struktūra ir antraštės. Vartojant vieningą standartą kompanijoje, yra daug paprasčiau susikalbėti, suprasti neaiškumus ar dviprasmybes

ne tik to pačio padalinio darbuotojams, bet ir darbuotojams visoje kompanijoje. Taip pat naudojant vieningą standartą yra iš anksto žinomos reikalavimo specifikacijos dalys, jų funkcijos ir specifika. Iš anksto nusistačius naudojamą standartą, sumažinama projekto sužlugimo, kaštų viršijimo tikimybė, ir stabdantys projektų eigą veiksniai. Minėti aspektai yra akivaizdūs vieningos struktūros naudojimo privalumai.

Projektuojame Volere šablono integravimą į programinių reikalavimų valdymo įrankį Doors. Veiklos diagramos schema su sričių padalinimais, pavaizduota 7 pav., parodo veiklų eigą tarp vartotojo ir sistemos. Vartotojo nesudėtingas šablono įtraukimas į kuriamą reikalavimų specifikavimo modelį, yra nesudėtingas ir paprastas. Vartotojo pagrindinis darbas yra išsikviesti šablonų biblioteką ir pasirinkti šabloną, kai tuo tarpu sistema importuos Volere šabloną į pasirinktąjį modelį. Įvykus nesklandumams ar kitiems trukdžiams dėl atsiradusių klaidų, šablonas neimportuojamas, ir vartotojas grąžinamas į pirminį pasirinktą modelį.



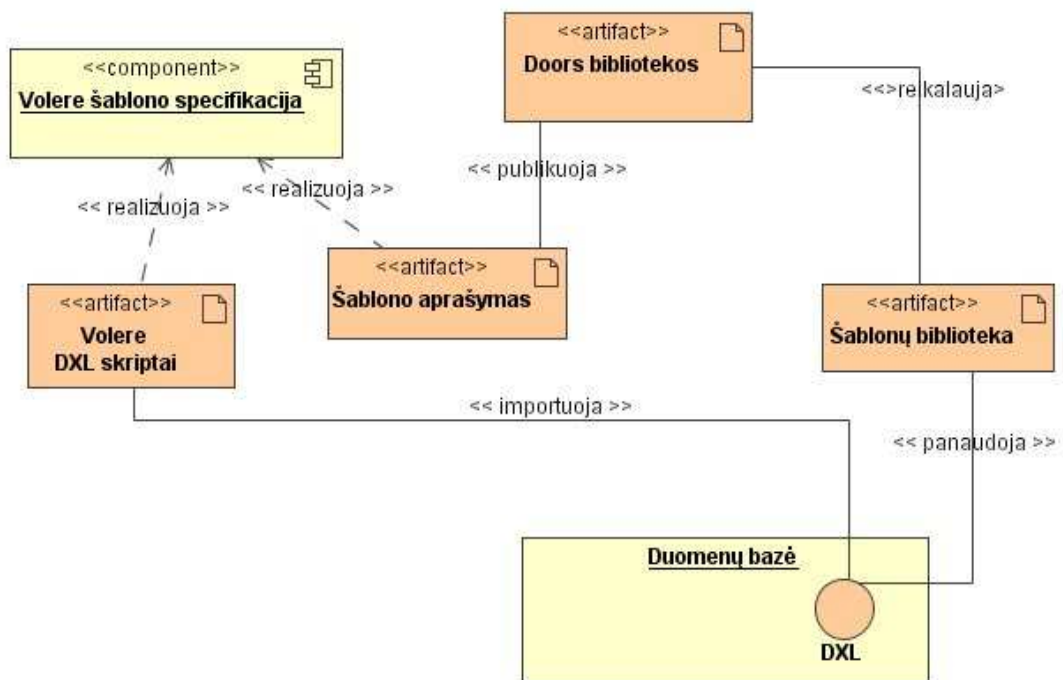
Šaltinis: Morkevičius, A.(2008)

7 pav. Volere šablono Doors įrankyje veiklos diagrama.

Doors programinė įranga turi skriptų kalbą Doors eXtension Language (DXL). Jos dėka, suteikiama galimybė kontroliuoti ir išplėsti Doors funkcionalumą. Panaudojant DXL skriptų kalbą galima išplėsti nuo Doors duomenų bazės konfigūravimo iki reikalavimų valdymo proceso patobulinimo integruojant papildomus įrankius ir funkcionalumus.

Volere reikalavimų specifikavimo šablono realizavimui Doors aplinkoje panaudoti DXL (Doors eXtension Language) programiniai skriptai.

Bendras projektuojamos sistemos vaizdas pateikiamas 8 pav. pavaizduotos modulių integravimo schemas - komponentų diagramos. Volere šablono specifikacija realizuojama dviem artefaktais, komponentais: Volere DXL skriptais, ir šablono aprašymas. Programiniai Volere DXL skriptai įtraukiami į šablonų bibliotekos duomenų bazę.



Šaltinis: Morkevičius, A.(2008)

**8 pav. Volere šablono ir Doors įrankio komponentų diagrama.**

Volere šablono specifikacijos aprašymas publikuojamas Doors bibliotekoje, kuri reikalauja duomenų šablonų bibliotekos panaudojant šablonų bibliotekos duomenų bazės sukauptais duomenimis. Šitoks sistemos modelis skirtas Volere šablono integracijai į reikalavimų specifikavimo programinį įrankį Doors. Nurodomas pasirinktas šablonas automatiškai iš duomenų bazės, kurioje patalpintas Volere šablono aprašantys DXL skriptai, automatiškai užkrauna šablono į vartotojo turimą Doors

programos modulį, taip suteikdami vartotojui aiškius reikalavimų specifikacijos kontūrus, vieningą struktūrą ir antraštes, kurių pagalba pagreitinamas vartotojo darbas ir efektyvumas kuriant reikalavimų specifikaciją.

#### 4.2 Volere reikalavimų surinkimo modelis

Kaip jau buvo minėta ankstesniuose skyriuose Volere yra ne tik reikalavimų šablonas, turintis specifinę reikalavimų struktūrą, standartą. Ši reikalavimų specifikacija praturtinta papildomomis formomis, padedančiomis pirminėse reikalavimų specifikacijos fazėse – reikalavimų surinkime. Prie Doors reikalavimų programiniame įrankyje Volere reikalavimų šablono suprojektuosiu minėtą Volere reikalavimų surinkimo formą .

Volere reikalavimų surinkimo formos pagrindu galima aprašyti pagrindinius funkcinius bei nefunkcinius reikalavimus, sudarančius bendrą vartotojo reikalavimų specifikacijos modelį **P**:

$$\mathbf{P} = \langle \mathbf{F}, \mathbf{N} \rangle.$$

Funkcinių reikalavimų aibė **F** aprašo kompiuterizuojamos veiklos aspektu tiesioginius vartotojo pageidavimus, susijusius su sistemos funkcionalumu.

$$\mathbf{F} = \langle \mathbf{L}, \mathbf{V}, \mathbf{O}, \mathbf{R}, \mathbf{E}, \mathbf{A}, \mathbf{S} \rangle;$$

*L* - paslaugos (funkcijos), kurias sistema turi užtikrinti;

*V* - sistemos atliekami veiksmai (duomenų tikrinimas, įrašymas, nuskaitymas, skaičiavimai);

*O* - sistemos įėjimo ir išėjimo informacijos srautai;

*R* - sistemos reakcija į įėjimus;

*E* - sistemos elgsena konkrečioje situacijoje;

*A* - sistemos elgsenos apribojimai;


*S* - vartotojo sąsajos reikalavimai [5].

Nefunkcinių reikalavimų aibė **N** aprašo kokybines sistemos ypatybes, nusakančias sistemos patikimumą, veikimo greitį, adaptyvumą ir kitus faktorius.

$$\mathbf{P} = \langle \mathbf{I}, \mathbf{U}, \mathbf{N}, \mathbf{G}, \mathbf{M}, \mathbf{K}, \mathbf{C}, \mathbf{T} \rangle$$

- I* – reikalavimai sistemos išvaizdai ir dizainui;
- U* – panaudojimo paprastumo, sistemos pritaikymo konkrečiam asmeniui nurodymas;
- N* – sistemos našumo įvertinimas (greitis, vėlavimai, tikslumas, talpa);
- G* – sistemos veikimo sąlygos, fizinė aplinka, sąveika su kitomis sistemomis;
- M* – sistemos priežiūra, palaikymas ir prisitaikymas;
- K* – saugumo, teisių, privatumo ir konfidencialumo aspektai;
- C* – kultūriniai, socialiniai, politiniai apribojimai;
- T* – teisiniai (įstatymų) apribojimai.

Specifikacijos modelyje **P** reikalavimams surinkti yra naudojama specializuota apklausos - surinkimo forma 4 pav. Šios forma padeda surinkti funkcinius ir nefunkcinius reikalavimus, juos analizuoti ir apibendrinti. Kaip matoma volere šablono struktūra išsamiai numato identifikuoti būsimo projekto kūrimui reikalingus reikalavimus. Pateikiama modifikuota Volere reikalavimų surinkimo forma, kurioje panaikinti reikalavimo įvedimo laukai, automatiškai įvedami Doors reikalavimų įrankio modulyje.

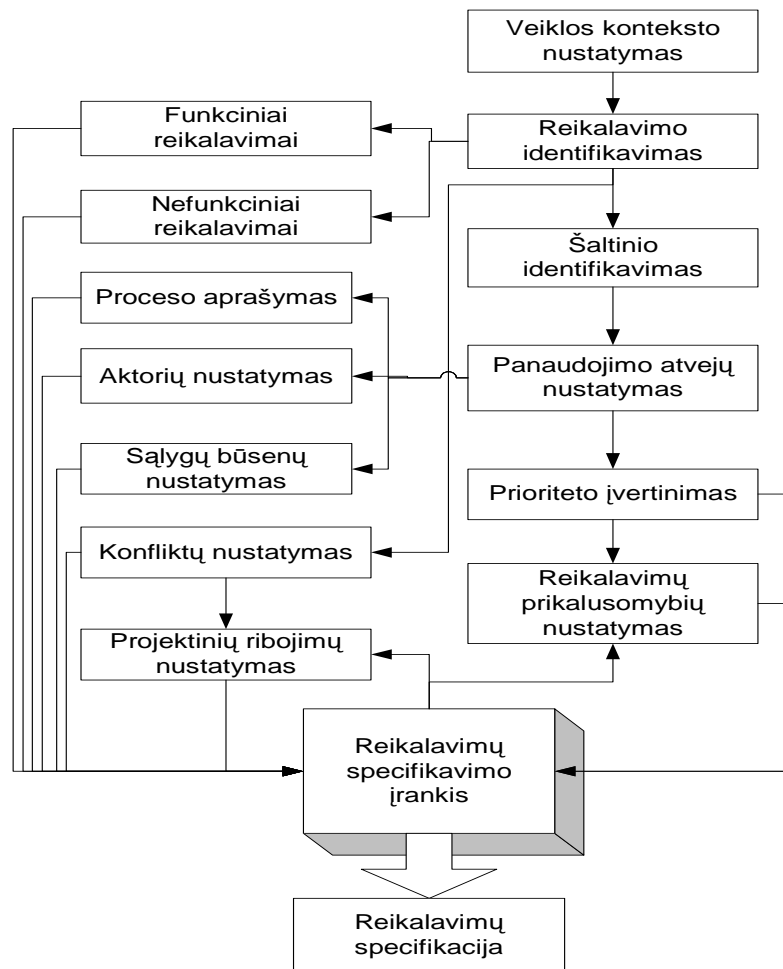
 <small>Copyright © Atlantic Systems Guild</small>	<u>Reikalavimo tipas:</u>	...	<u>Ivykis/panaudojimo atvejis #:</u>	..
<b><u>Aprašymas:</u></b>	<i>Trumpas reikalavimo aprašymas</i>			
<b><u>Pagrindimas:</u></b>	<i>Reikalavimo pagrindimas, nurodomos jo atsiradimo priežastys</i>			
<b><u>Šaltinis:</u></b>	<i>Reikalavimo autorius</i>			
<b><u>Tikimo kriterijus:</u></b>	<i>Reikalavimo matavimas, toks kad, yra galima ištestuoti, jei sprendimas atitinka originalų reikalavimą.</i>			
<b><u>Užsakovo tenkinimas:</u></b>	<i>Skaičius nuo [1-5]</i>	<b><u>Užsakovo netenkinimas:</u></b>	<i>[1-5]</i>	
<b><u>Priklausomybės:</u></b>	<i>Reikalavimai, turintys įtaką nagrinėjamam reikalavimui.</i>	<b><u>Konfliktai:</u></b>	<i>Kitų reikalavimų vykdymas negalimas jei egzistuos šis reikalavimas</i>	

Šaltinis: autoriaus parengta pagal Robertson S., Robertson J. Volere Requirements Specification template (2006)

**9 pav. Modifikuota Volere reikalavimų surinkimo forma.**

Volere adaptuotos reikalavimų surinkimo formos pagalba, surinkti reikalavimai bus valdomi, susiejami sąryšiais, priklausomybėmis, konfliktais, panaudos atvejais. Panaudos atvejų surinkimą atliksime su pačiame įrankyje įdiegta panaudos atvejų surinkimo ir modifikavimo forma. Jos pagalba surinktus panaudojimo atvejus panaudosime tik reikalavimų surinkimo formoje nurodant ryšius.

Panaudojant Volere reikalavimų surinkimo formą, sukurtas Volere surinkimo formos pritaikymo konceptualusis modelis 10 pav.



Šaltinis: Morkevičius, A.(2008)

**10 pav. Volere šablono pritaikymo konceptualus modelis.**

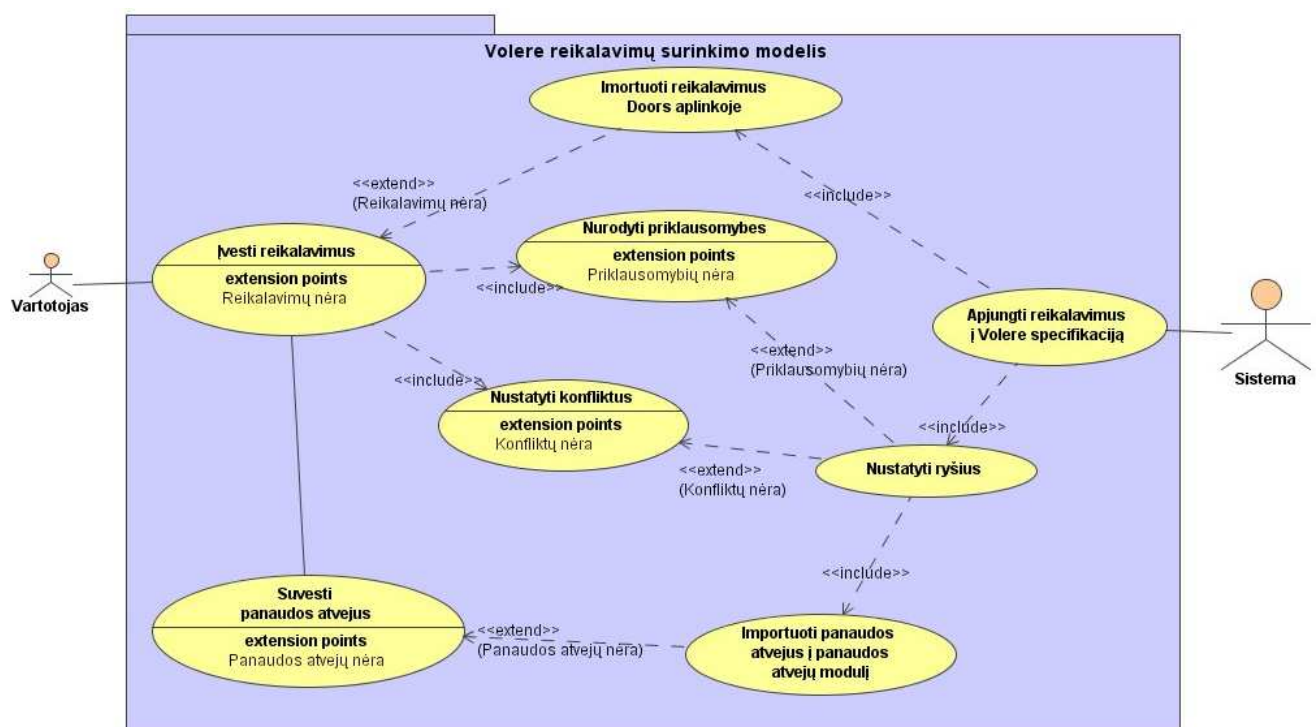
Reikalavimų specifikavimas pradamas nustatant veiklos kontekstą. Toliau identifikuojamas reikalavimas išskirstomas į funkcinių ir nefunkcinių reikalavimų kategorijas, bei nustatomi galimi konfliktai, kuriuose nurodomas kitų reikalavimų neveiknumas egzistuojant identifikuotam reikalavimui. Suformuojamas projektinių ribojimų nustatymas, nurodantis kuriamo projekto ribas. Kita



specifikacijos komponentė – tai panaudojimo atvejų išskyrimas, kurie nustato dalykinės srities vartotojus, funkcijas ir jų sistematiką. Toliau identifikuojami reikalavimų prioritetai, tarpusavio priklausomybės. Pritaikant minėtas modelio komponentes, sudarytas Volere šablonu paremtas reikalavimų specifikavimo konceptualus modelis.

Remiantis reikalavimų specifikavimo įrankių analize, atlikta ankstesniame skyriuje, pasirinktas Doors reikalavimų specifikavimo įrankis, kuriam projektuojamas reikalavimų surinkimo forma Volere šablono pagrindu. Modelio pavaizduotu paveiksle 10 pav.

Nurodomos būsimos vartotojo funkcijos reikalingos reikalavimų įvedimui, ryšių tarp reikalavimų nustatymui, bei panaudos atvejų identifikavimui nustatyti. Sukurto modelio funkcija – įdiegti Volere šablonu paremtą specifikacijos reikalavimų surinkimo formą į Doors įrankį. Sukurto modelio pagalba, sistema dalį vartotojo funkcijų automatizuos, taip sudarydama didesnę vartotojo darbo našumą ir efektyvumą reikalavimų specifikavimo procese.



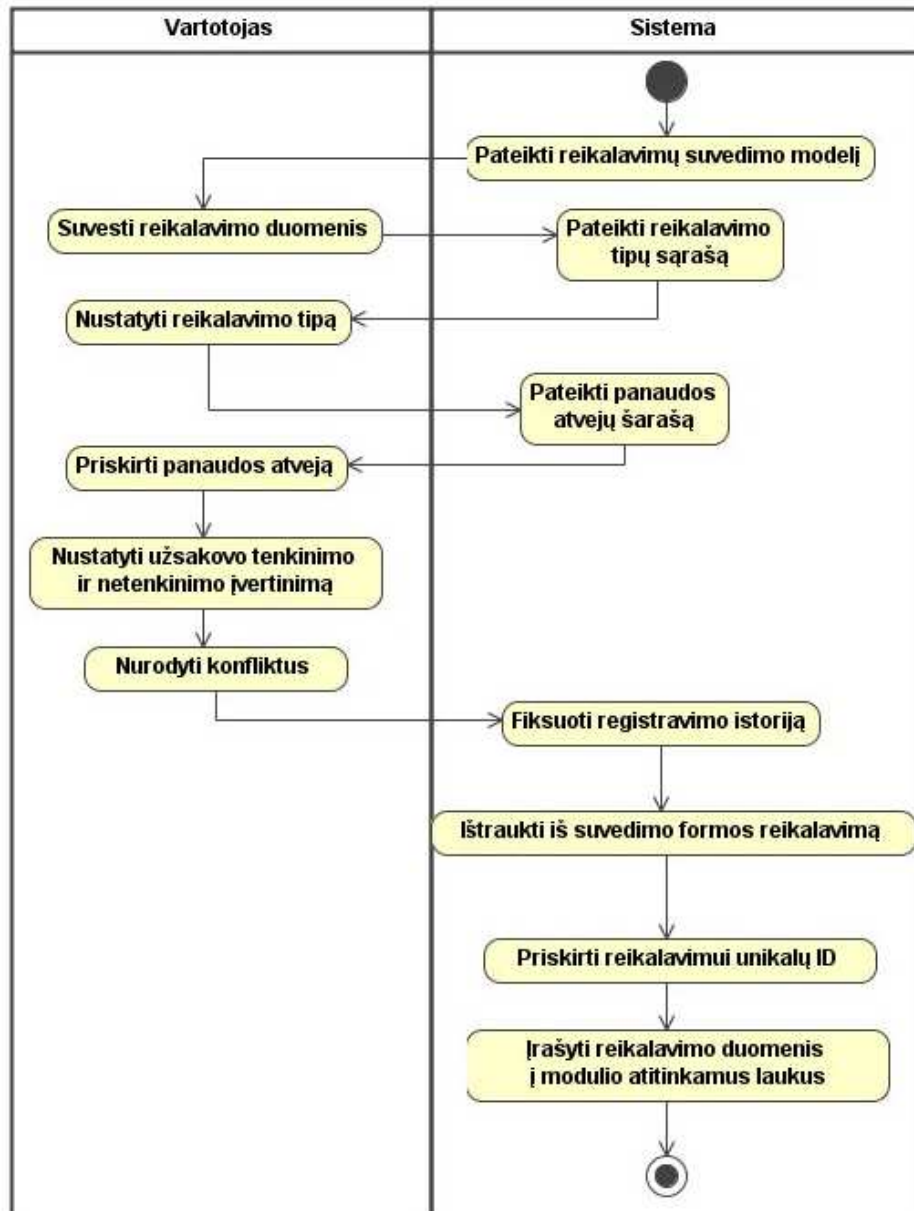
Šaltinis: Morkevičius, A.(2008)

11 pav. Modelio panaudos atvejų diagrama

Kaip matosi iš 11 pav. sistemos vartotojo funkcijos įvesti reikalavimus, nurodyti reikalavimų priklausomybes, nustatyti konfliktus. Sistemos vaidmuo yra suvestus reikalavimų duomenis importuoti į Doors reikalavimų modulį. Taip pat nurodytoms priklausomybėms, konfliktams bei panaudojimo

atvejams nustatyti ryšius su reikalavimais. Tokiu būdu reikalavimų surinkimas Doors įrankyje yra apjungiamas į Volere specifikaciją pavaizduotą 8 pav.

Paveiksle 12 pav. pateikiama reikalavimų specifikavimo modelio veiklos diagrama:

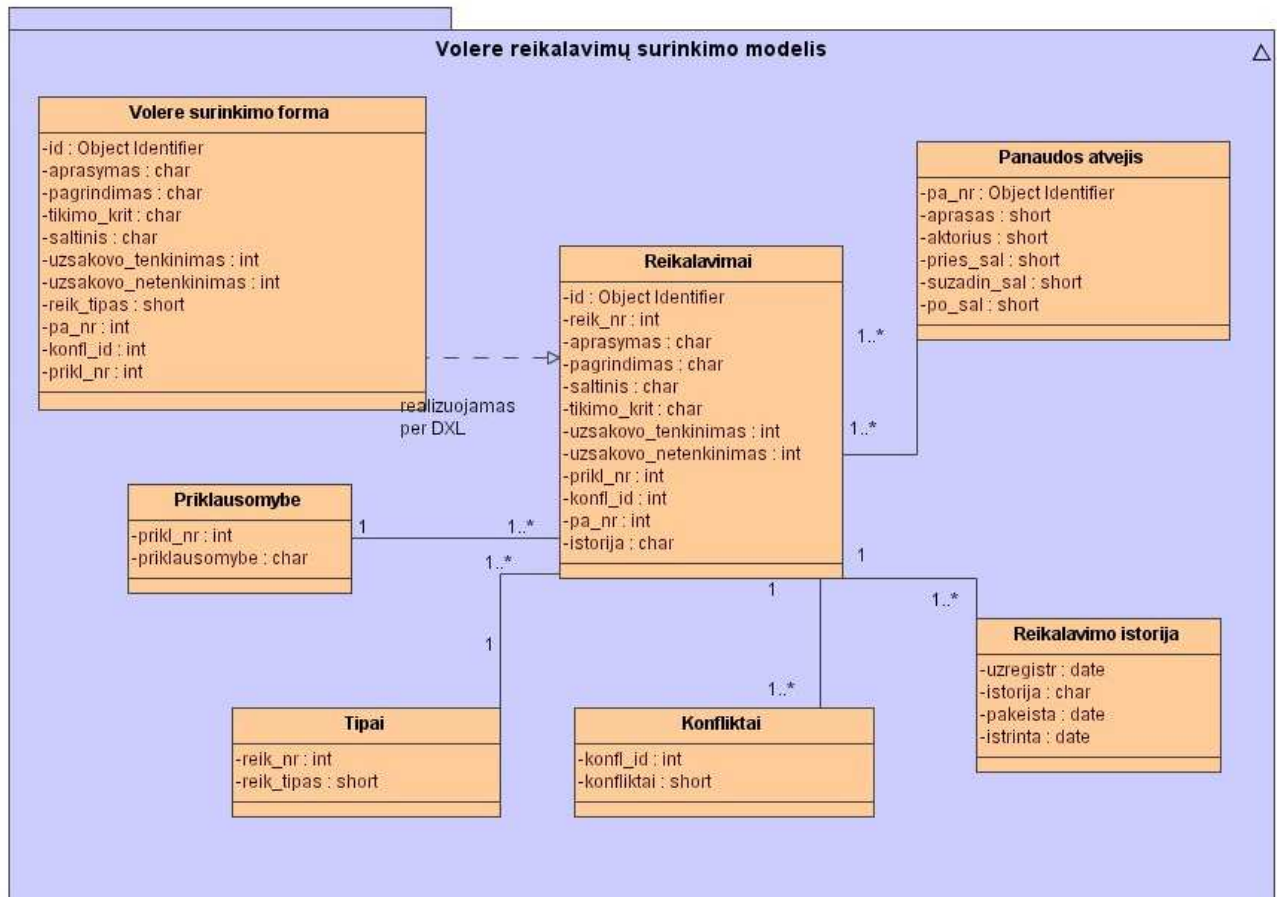


Šaltinis: Morkevičius, A.(2008)

12 pav. Reikalavimų suvedimo modelio veiklos diagrama.

Kaip matyti iš 12 pav. didžiąją dalį vartotojo funkcijų sistema automatizuoja, todėl yra sumažinamos vartotojo pastangos, skirtos reikalavimui specifikuoti. Vadinasi vartotojo darbas specifikuojant reikalavimus tampa našesnis ir efektyvesnis.

Paveiksle 13 pav. pavaizduojamas Volere šablono realizacijos Doors reikalavimų valdymo pakete modelis.



Šaltinis Morkevičius, A.(2008)

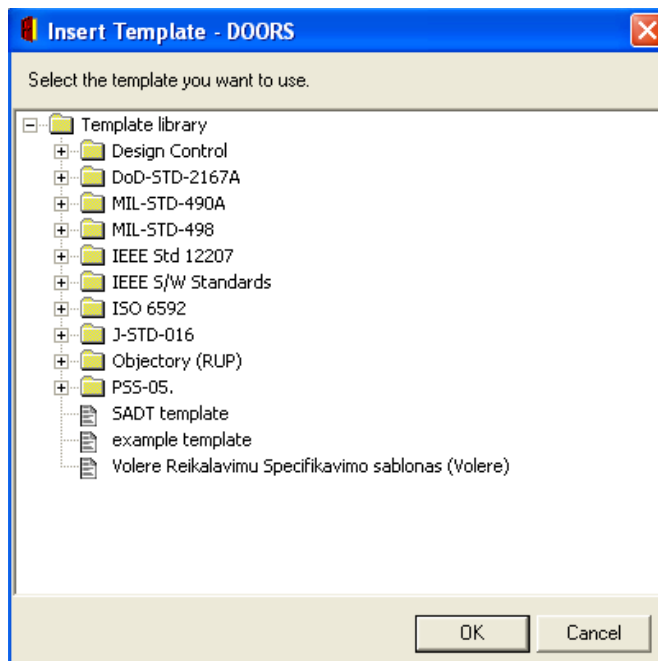
13 pav. Volere šablono realizacijos Doors reikalavimų valdymo pakete modelis.

Volere surinkimo forma, realizuojama Doors DXL išplėtimo kalba. Formoje esantys reikalavimo duomenys panaudojant DXL iš Volere surinkimo formos reikalavimams perkelia į Volere reikalavimų specifikacijos modulį.

## 5. MODELIO PROTOTIPAS

Remiantis 4-iaame skyriuje aprašytu sistemos modeliu, sukurta Volere reikalavimų specifikavimo šablono realizacija reikalavimų inžinerijos įrankyje Doors.

Šablonai padeda išlaikyti nuoseklumą ir skatina vartotojus naudoti standartinę struktūrą. Realizuojamas Volere reikalavimų specifikacijos šablonas Doors reikalavimų inžinerijos įrankio šablonų bibliotekoje (14 pav.) kartu su kitais prieš tai buvusiais šablonais.

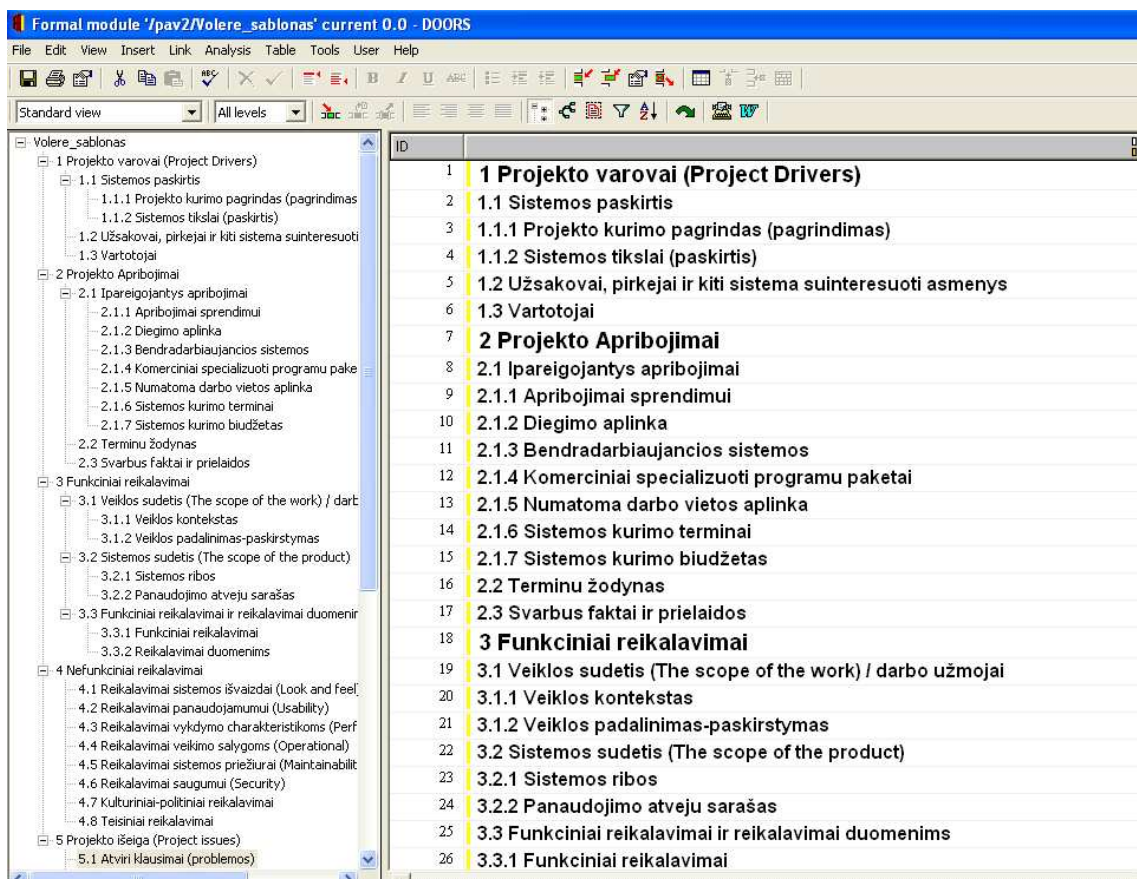


Šaltinis: Morkevičius, A.(2008)

**14 pav. Volere šablono realizacija Doors šablonų bibliotekoje.**

Šablonas realizuojamas remiantis praeitame skyriuje 7 pav. ir 8 pav. aprašytais schemomis, panaudojant Doors DXL išplėtimo kalbą. Volere standarto struktūros formavimo programinis kodas pateikiamas priede Priedas Nr. 1.

Iškvietus 14 pav. minėtą šablona, pateikiu realizuotą Doors modulyje šablono struktūrą (15 pav.).



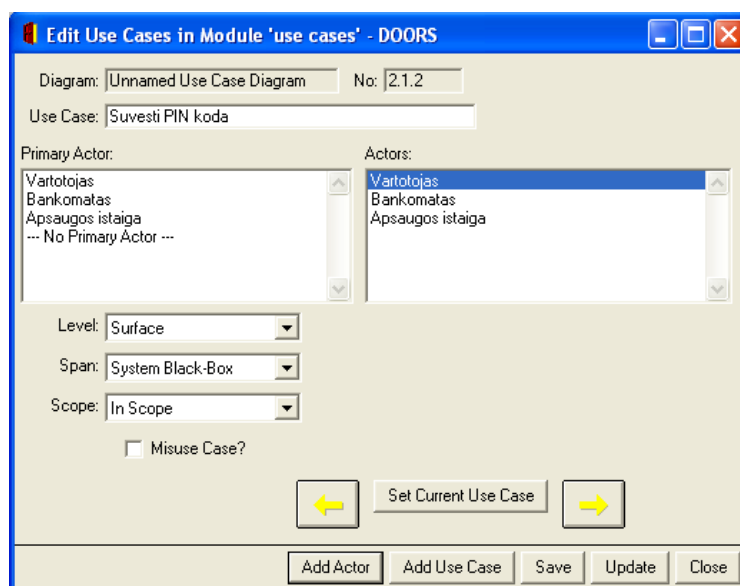
Šaltinis: Morkevičius, A.(2008)

15 pav. Volere šablono realizacijos prototipas.

Iš šablonų bibliotekos išsirinkus Volere reikalavimų specifikacijos šabloną, Doors modulis adaptuojamas visa Volere reikalavimų specifikacijos struktūra (išskyrus panaudojimo atvejus).

Sukurtam šablonui, sukurta ir praeitame skyriuje aprašyta reikalavimų surinkimo forma. Gautas formos prototipas, kurios grafinė vartotojo sąsaja pateikiama 19 pav.

Sukurtam prototipui pavaizduoti, pasirinktas konkretus uždavinys – eksperimentas. Per kurio prizmę pavaizduosiu Volere šablono pritaikymo funkcionalumą Telelogic Doors reikalavimų specifikavimo įrankyje. Pasinaudojant sukurtu Volere reikalavimų specifikavimo šablonu, bei Volere reikalavimų surinkimo firma, specifikuosiu banko „Bankas“ bankomatų sistemai reikalavimus.



Šaltinis: Morkevičius, A.(2008)

**16 pav. Panaudos atvejo įvedimo forma.**

Norimame projekte susikuriame modulį panaudojimo atvejams talpinti. Tuomet pasinaudojant panaudojimo atvejų surinkimo šabloną (16 pav.) įdiegtą Doors įrankyje, suvesti panaudojimo atvejai, jiems priklausantys aktoriai, ir visa likusi informacija, reikalinga panaudojimo atvejams indentifikuoti. Paveiksle 17 pav. pavaizduotas panaudojimo atvejų fragmentas

ID	Use Case Level	Use Cases	Actors
6	Surface	2.1.1 Pateikti kortelę	Vartotojas
38	Surface	2.1.2 Suvesti PIN koda	Vartotojas
65	Overview	2.1.3 Patikrinti PIN duomenų bazeje	Bankomatas
93	High	2.1.4 Paziureti saskaitos likuti	Vartotojas
120	Surface	2.1.5 Nusiimti pinigų	Vartotojas
147	Surface	2.1.6 Atspausdinti čekius	Bankomatas
174	Surface	2.1.7 Išmokėti pinigus	Bankomatas
201	Surface	2.1.8 Gražinti kortelę	Bankomatas

Šaltinis: Morkevičius, A.(2008)

**17 pav. Panaudojimo atvejų sąrašo fragmentas**

Panaudojimo atvejų šablono specifikacijos fragmentas pateikiamas 18 pav. Jis pateikiamas detaliu panaudojimo atvejo išplėstiniu sąrašu.

ID	Use Cases	Actors	Manual Step
200	---		
201	2.1.8 Grazinti kortele	Bankomatas	
203	2.1.8.1 Context Vartotojui grazinama kortele		
204	2.1.8.2 Description Istumiama kortele is bankomato		
202	2.1.8.3 Primary Scenario		
205	Vartotojui nurodzius kad noni uzbaigti darba ir nepageidatija daugiau atlikti tranzakciju, kortele yra grazinama		
206	2.1.8.4 Variations		
207	2.1.8.4.1 Unnamed Variation		
208	Undefined Step		
209	2.1.8.5 Exceptions		
210	2.1.8.5.1 Istumiama kortele		
211	Kortele bus grazinta vartotojui		
212	2.1.8.6 Trigger		
213	---		
214	2.1.8.7 Preconditions		
215	Ivykdziaus tranzakcija (-as) aktyvuojamas darbo baigimo funkcija - "Baigti darba"		
216	2.1.8.8 Stakeholders and Interests		
217	---		
224	2.1.8.9 Business Rules		
225	---		
226	2.1.8.10 Process Changes		
227	---		

Šaltinis: Morkevičius, A.(2008)

### 18 pav. Panaudojimo atvejų šablono specifikacija

Kai sistemoje jau yra sukurtas panaudojimo atvejų modulis, su bent vienu panaudojimo atveju, galima pradėti suvedinėti reikalavimus Volere reikalavimų surinkimo formos (19 pav.) pagalba. Realizuotos formos programinis kodas pateikiamas priede Priedas Nr.2.

Šaltinis: Morkevičius, A.(2008)

### 19 pav. Volere reikalavimų surinkimo forma.

Reikalavimų surinkimas vykdomas pagal Volere šablono struktūrą ir praeitame skyriuje aprašytą reikalavimų surinkimo modelį  $P = \langle F, N \rangle$ , kur

$$F = \langle L, V, O, R, E, A, S \rangle$$

$$P = \langle I, U, N, G, M, K, C, T \rangle.$$

Taigi nustatius reikalavimo tipą ir jo sub - tipą, įvedus reikalavimo duomenis, patenkinimo/netenkinimo įverčius, suvedami panaudojimo atvejų, priklausomybių ir konfliktų sąryšiai. Šie reikalavimų sąryšiai įvedami identifikaciniu numeriu, kelis sąryšio ID atskiriant kableliu. Minėti sąryšiai yra skirti:

- Identifikuoti funkcinių reikalavimų sąsają su nefunkciniais reikalavimais;
- Identifikuoti susijusius tarpusavyje - priklausomus reikalavimus;
- Nustatyti įvykį / panaudojimo atvejį;
- Susieti reikalavimą su konfliktinėmis situacijomis, kurios įvyks jei egzistuos šis reikalavimas.

Kadangi priklausomybės ir konfliktai – tai reikalavimai, susiję su nagrinėjamu reikalavimu, todėl priklausomybių ir konfliktų sąryšiai užmezgami tame pačiame reikalavimų modulyje. Panaudojimo atvejai, talpinami atskirame - panaudojimo atvejų modulyje, todėl sąryšiai tarp nagrinėjamų reikalavimų ir panaudojimo atvejų sudaromi modulių lygmenyje. Dėl ribojamo Telelogic Doors versijos funkcionalumo, Volere reikalavimų surinkimo formos nustatomi sąryšiai nėra pilnai funkcionalūs, kadangi ryšiai užmezgami tarpinėje Doors atmintyje.

Naujame Doors modulyje, iš šablonų bibliotekos išsirinkus Volere reikalavimo specifikavimo šabloną (15 pav.), bei suvedus reikalavimus 19 pav. pavaizduota reikalavimų surinkimo forma, gaunamas 20 pav. pateiktas bankomatų reikalavimų specifikacijos fragmentas.



Aprašymas	Pagrindimas	Sahris	Tikimo kriterijus	Uzzakovo tenkinim	Uzzakovo netenkinim	Reikalavimu tipai
<b>3.1.1 Veiklos kontekstas</b>						
<b>3.1.2 Veiklos paskirstymas</b>						
<b>3.2 Sistemos sudėtis (The scope of the product)</b>						
<b>3.2.1 Sistemos ribos</b>						
<b>3.3 Funkciniai reikalavimai ir reikalavimai duomenims</b>						
<b>3.3.1 Funkciniai reikalavimai</b>						
<b>3.3.1.1</b> Bankomatas turi būti pritaikytas diegti uzdarose patalpose ir ju sieniu lauko pusese.	Galimybė bankomata imonuoti ir laukinese ir vidinese sieniu pusese	Albertas M.		5	5	Funkcinis
<b>3.3.1.2</b> Bankomatas turi aptarnauti tik banko "Bankas" vartotojus		Albertas M.		4	5	Funkcinis
<b>3.3.1.3</b> Bankomatas turi priimti tik "Bankas" galiojancias korteles		Albertas M.			5	Funkcinis
<b>3.3.1.4</b> Bankomatas turi aptarnauti tik viena vartotoja vienu metu		Albertas M.		3	5	Funkcinis
<b>3.3.1.5</b> Bankomatas turi isduoti pinigus ir ekrane rodyti informacija		Albertas M.		5	5	Funkcinis
<b>3.3.1.6</b> Bankomato laukimo busenoje, turi buti parodytas pranesimas - banko atributika	Bendros informacijos ar reikamos rodymo galimybė.	Albertas M.		3	5	Funkcinis
<b>3.3.1.7</b> Bankomate esantys mygtukai turi islikti neaktyvus kol neidedama "Bankas" kortele	Uzkirsti kelia bet kokiems ne pagal paskirti bankomato panaudojimams	Albertas M.		5	5	Funkcinis
<b>3.3.1.8</b> Vartotojui idejus kortele, bankomato korteliu skaitytuvas ja turi itraukti				3	5	Funkcinis
<b>3.3.1.9</b> Jei korteles neimanoma perskaityti, grazinti kortele vartotojui ir pranesi kad nepavyko nuskaityti korteles.	Informuoti klienta apie problema	Albertas M.		5	5	Funkcinis
<b>3.3.1.10</b> Bankomatas turi itraukti nepaimtas korteles	Patikras korteles itraukti, kad nepasinaudotu treti asmenys			2	4	Funkcinis
<b>3.3.1.11</b> Idejus kortele prasyti vartotojo suvesti identifikacijos numeri - PIN koda	Itraukus kortele reikia identifikuoti vartotoja.	Albertas M.		4	4	Funkcinis
<b>3.3.1.12</b> Lauke statomuose bankomatuose turi buti idiegta klimato kontrole	Galimi lauko temperaturs ir santykinio drėgnumo slankiniai neturi itakoti bankomato veidai	Albertas M.		3	4	Funkcinis
<b>3.3.1.13</b> Bankomate turi veikti Standartinis asmeninis kompiuteris su Pentium IV procesorumi		Albertas M.		4	5	Funkcinis
<b>3.3.1.14</b> Bankomato seifas turi atitikti saugumo standartus	Seifas privalo atitikti standartus.	Albertas M.		4	5	Funkcinis

Šaltinis: Morkevičius, A.(2008)

## 20 pav. Volere reikalavimų specifikacija.

Paveiksle 20 pav. paryškintomis raidėmis matosi Volere standarto struktūrinės dalys, o smulkesnio šrifto raidėmis (pradedant punktu 3.3.1.1.) matomi suvesti funkcinų reikalavimų fragmentas.

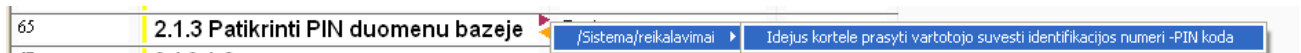
Reikalavimų suvedimo forma (19 pav.) įvedant reikalavimą ir nurodant priklausomybes, konfliktus, panaudojimo atvejus, yra užmezgami sąryšiai tarp reikalavimų modulio ir panaudojimo atvejų modulio. Reikalavimų sąryšis, šiuo atveju priklausomybės ir panaudojimo atvejai, pavaizduojamas paveiksluose 21 pav ir 22 pav. Kur reikalavimas „3.3.1.11 Įdėjus kortelę prašyti suvesti identifikacijos numerį – PIN kodą“ susietas su panaudojimo atvejais „Suvesti PIN kodą“ ir „Patikrinti PIN duomenų bazėje“, bei su priklausomu nefunkciniu reikalavimu „Pin kodą turi sudaryti 4 skaitmenys“.

117	<b>3.3.1.11</b> Idejus kortele prasyti vartotojo suvesti identifikacijos numeri -PIN koda	Itraukus kortele reikia /Sistema/reikalavimai	Albertas M.
114	<b>3.3.1.12</b> Lauke statomuose bankomatuose turi buti idiegta klimato kontrole	/Sistema/use cases ir santykinio drėgnumo	Suvesti PIN koda Patikrinti PIN duomenų bazėje

Šaltinis: Morkevičius, A.(2008)

## 21 pav. Sąryšiai su priklausomais reikalavimais ir panaudos atvejais

Analogiškai panaudojimo atvejų modulyje užmezgamas sąryšis tarp panaudojimo atvejo ir nagrinėjamo reikalavimo „3.3.1.11 Įdėjus kortelę prašyti suvesti identifikacijos numerį – PIN kodą“.



Šaltinis: Morkevičius, A. (2008)

**22 pav. Panaudojimo atvejo sąryšis su reikalavimų modulių**

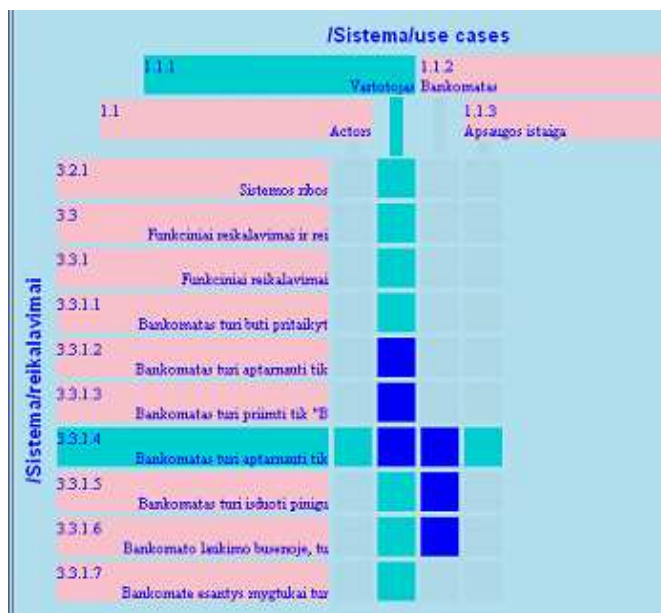
Paveiksle pavaizduojamas užmegztų reikalavimų priklausomybių, konfliktų ir panaudojimo atvejų peržiūros pjūvis.

ID		Priskirti panaudos atvejai	Priskirti konfliktai	Priskirti priklausomi reikal.
43	<b>4.3.1 Greicio ir velavimu reikalavimai</b>			
44	<b>4.3.2 Saugumo kritiniai reikalavimai</b>			
137	4.3.2.1 Grazinti kortele dingus elektrai	201		115,140
45	<b>4.3.3 Tikslumo reikalavimai</b>			
121	4.3.3.1 Pin koda turi sudaryti 4 skaitmenys	38,65		117
123	4.3.3.2 Maziausiai 50% kupiuru turi buti 20 Lt. ir 50 Lt nominalo kupiuros			
46	<b>4.3.4 Patikimumo ir tinkamumo naudoti reikalavimai</b>			
125	4.3.4.1 Sekmingas pinigų išdavimas turi buti 999 is 1000 kartu	174		
126	4.3.4.2 Sistema turi veikti 24val per para ir 365 dienas per metus			
140	4.3.4.3 Turi buti užtikrintas nepertraukiamas elektros energijos tiekimas			92,137
141	4.3.4.4 Korteles idejimo anga turi buti apsaugota nuo temperatūros pokyciu			104,105, 114
47	<b>4.3.5 Stiprumo ir klaidos tolerancijos reikalavimai</b>			
48	<b>4.3.6 Talpumo reikalavimai</b>			
49	<b>4.4 Reikalavimai veikimo salygoms (Operational)</b>			
50	<b>4.4.1 Fazines alpinkos reikalavimai</b>			
91	4.4.1.1 Galimybė naudotis prie ivairaus apsvietimo			
116	4.4.1.2 Montuojamas i ne plonesne nei 360 mm siena			
93	4.4.1.3 Atsparus saules spinduliams ekranas			
92	4.4.1.4 Taupus energijos panaudojimas			97,100,140
104	4.4.1.5 Temperatura: nuo -40° C iki + 55° C			114,141
105	4.4.1.6 Santykine oro dregme nuo 5 % iki 85 %			114,141
107	4.4.1.7 Nerudijancio plieno klaviatura			
108	4.4.1.8 Klaviaturos apsauga nuo pasalinii akiu			
109	4.4.1.9 Klaviatura zymejimas Brailio rastu			111
112	4.4.1.10 Prie bankomato privaziavimas invalido vezimeliu			109
113	4.4.1.11 Bankomato klaviatura turi buti ne zemiau kaip 1,25m. ir neauksčiau 1.5 m. aukscio			111

Šaltinis: Morkevičius, A.(2008)

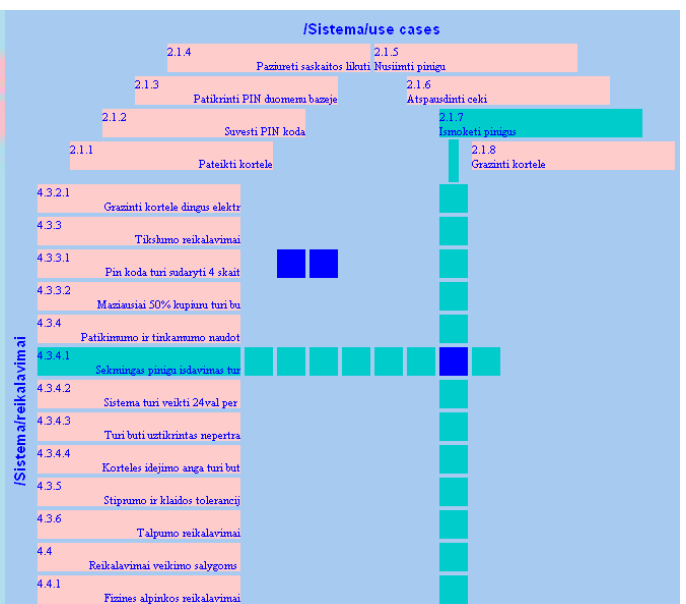
**23 pav. Reikalavimų priklausomybių ir sąryšių peržiūros pjūvis**

Automatizuotas reikalavimų sąryšių užmezgimas, suteikia galimybę įvairiems peržiūros pjūviams atvaizduoti grafiniu būdu. Tai labai vaizdus, aiškiai suprantamas sąryšių atvaizdavimas, bei Doors įrankio teikiama galimybė ateityje modifikuoti ar papildyti naujais sąryšiais ir priklausomybėmis. Paveiksle 24 pav. pavaizduotas reikalavimų ir aktorių sąryšių grafinis pjūvis, o paveiksle 25 pav. sistemos reikalavimų ir panaudojimo atvejų sąryšių pjūvis.



Šaltinis: Morkevičius, A.(2008)

24 pav. Sąryšių tarp reikalavimų ir aktorių pjūvis



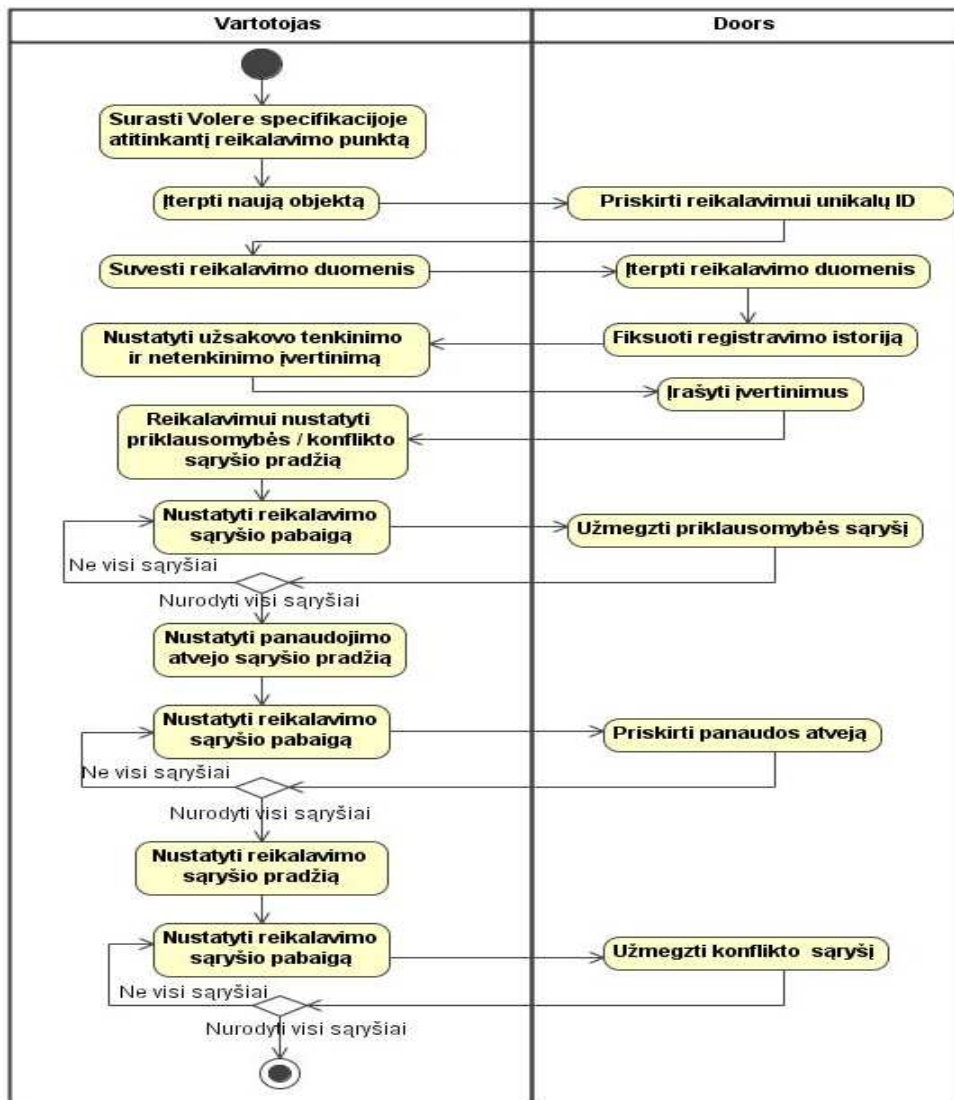
Šaltinis: Morkevičius, A.(2008)

25 pav. Sąryšių tarp reikalavimų ir panaudojimo atvejų pjūvis

Kaip matyti iš prototipo eksperimento, Volere šablono automatizuotas reikalavimų surinkimas bei automatizuotas sąryšių sudarymas Doors įrankyje teikia akivaizdžią naudą ir panaudojimo galimybes tolimesnėse sistemos kūrimo gyvavimo ciklo etapuose. Yra gan sudėtinga apskaičiuoti realią automatizuoto reikalavimų surinkimo naudą laiko, našumo, efektyvumo kriterijų atžvilgiu. Tačiau tai galime parodyti sumažėjusiu vartotojui reikiamų atlikti veiklų skaičiumi. Tam pačiam procesui sumažinamas veiksmų skaičius sąlygoja padidėjusį vartotojo darbo našumą ir sumažėjusį laiko kiekį skirtą reikalavimui identifikuoti.

Paveiksluose 26 pav. ir 27 pav pateikiamas neautomatizuoto ir automatizuoto Volere reikalavimų surinkimo procesų veiklos diagramos. Paveiksle 26 pav. pateikta neautomatizuoto Volere reikalavimų surinkimo proceso veiklos diagrama, o paveiksle 27 pav. – automatizuoto proceso.

26 pav. pavaizduotos visos veiklos, kurios turi būti atliktos norint specifikuoti reikalavimą Telelogic Doors įrankiu, nenaudojant Volere reikalavimų surinkimo formos, tačiau remiantis Volere reikalavimų specifikuojimo standarto struktūra.



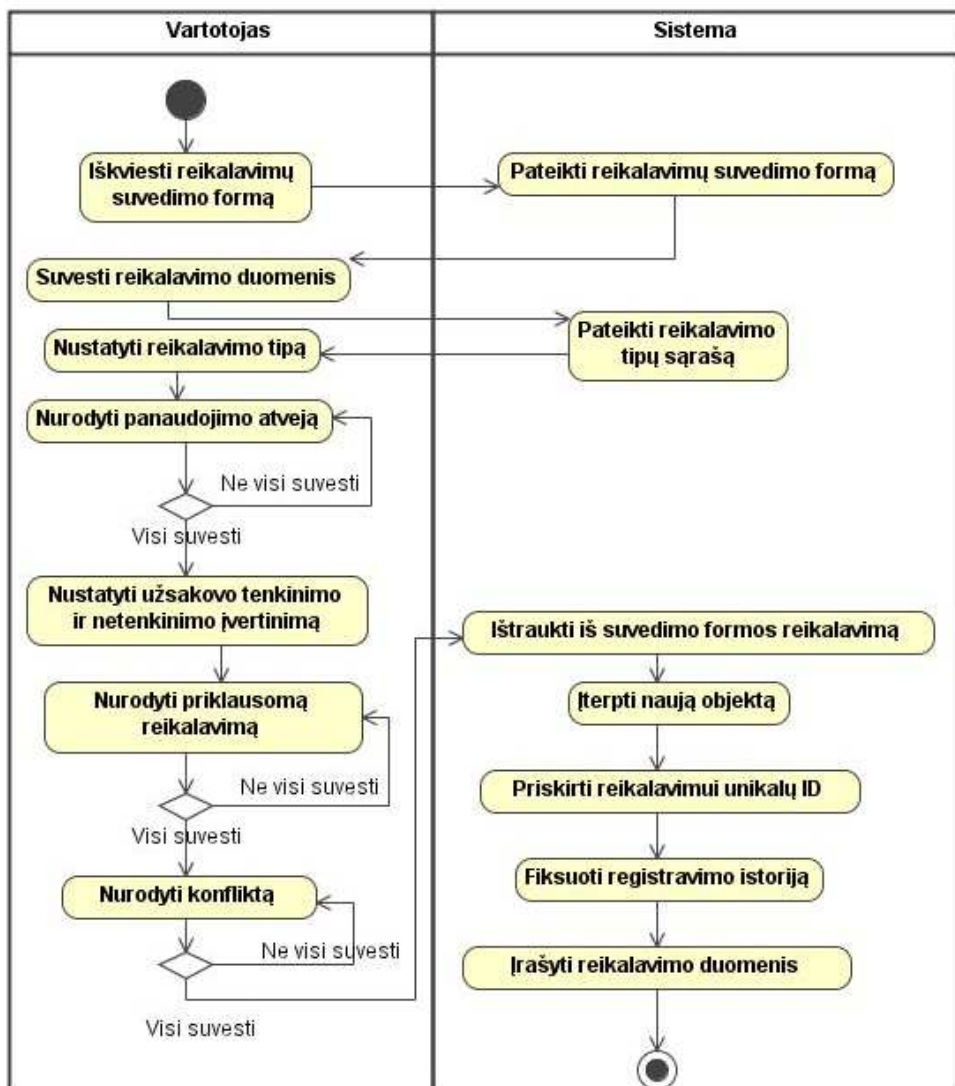
Šaltinis: Morkevičius, A.(2008)

26 pav. Neautomatizuoto reikalavimų surinkimo veiklos

Sistemos (Doors) dalies veiklos yra tik įvykdyti – realizuoti vartotojo atliekamus uždavinius: priskirti, įterpti, fiksuoti, įrašyti, užmegzti. Vartotojo dalyje apibrėžiamos visos veiklos, skirtos mechaniniu – rankiniu būdu specifiuoti reikalavimą. Vartotojas rankiniu būdu surasti Volere standarto struktūroje reikalavimo talpinimo vietą – pagal reikalavimo tipą. Toliau įterpti naują objektą – reikalavimą, suvesti jo duomenis, nustatyti užsakovo tenkinimo ir netenkinimo įverčius, nustatyti ir susieti priklausomybių ryšius tarp reikalavimų (pradedant ryšį nustatant ryšio pradžią, ir užbaigiant ryšį nustatant susiejamo reikalavimo pabaigą). Tokiu pačiu principu nustatomi ir konfliktai, bei

panaudojimo atvejai, kurie talpinami atskirame panaudos atvejų modulyje. Tik įvykdžius šias minėtas veiklas, bus atliktas reikalavimų specifikavimas.

Toliau panagrinėsiu 27 pav. pavaizduotą automatizuoto Volere reikalavimų surinkimo proceso veiklas. Reikėtų atkreipti dėmesį į tai, jog pavaizduota veiklos diagrama yra atliekamo reikalavimų specifikavimo, t.y. lygiai toks pat procesas, kaip ir prieš tai nagrinėtoje veiklos diagramoje (26 pav.) Automatizuoto Volere reikalavimų surinkimo veiklos diagramoje akivaizdžiai matosi sumažėjęs vartotojui skirtų veiklų kiekis. Didžioji „darbo“ dalis perkeliama į sistemos atliekamų veiklų dalį.



Šaltinis: Morkevičius, A.(2008)

27 pav. Automatizuoto Volere reikalavimų surinkimo veiklos

Vartotojo atsakomybėje paliekamos šios reikalavimo specifikavimo veiklos:

- Iškviesti reikalavimų suvedimo formą;
- Suvesti reikalavimo duomenis;
- Nustatyti reikalavimo tipą;
- Nurodyti panaudojimo atveją (-us);
- Nustatyti užsakovo tenkinimo ir netenkinimo įverčius;
- Nurodyti priklausomą (-us) reikalavimą (-us);
- Nurodyti konfliktą (-us).

Atlikus reikalavimo specifikavimo proceso veiklų palyginimus automatizuotu ir neautomatizuotu reikalavimų surinkimo būdu paaiškėjo, kad sukurto prototipo pagalba sumažinamas vartotojo veiksmų kiekis, reikalingas specifiuoti reikalavimą. Kai neautomatizuotu būdu vartotojo veiksmų kiekis buvo lygus 10, o automatizuotu būdu šis skaičius sumažėjo iki 7. Vadinasi galima teigti, kad sumažėjus 30% vartotojo atliekamų veiksmų, sumažės vartotojo įdedamos pastangos reikalavimui specifiuoti, taip pat sumažės tam skirtas laikas. Iš to seka, kad padidės reikalavimų surinkimo proceso našumas ir efektyvumas.

## IŠVADOS

1. Aptarta aukštos kokybės reikalavimų proceso nauda kuriamoms sistemoms. Pastebėta, kad programinės įrangos kūrimo reikalavimų inžinerijos yra aktualus ir svariai įtakojantis galutinį produkto rezultatą.
2. Išanalizuoti reikalavimų specifikacijos dokumento šablonai IEEE 830-1998 ir Volere, bei išanalizuota Volere šablono reikalavimų surinkimo forma.
3. Apžvelgti egzistuojantys reikalavimų inžinerijos programiniai įrankiai, atlikta detali IBM Rational RequisitePro, Telelogic Doors, TCP Sistemas & Ingenieria IRqA, Borland CaliberRM, Gatherspace programinių įrankių analizė. Surastas geriausias iš analizuotų įrankių - Telelogic kompanijos produktas Doors. Jame realizuota daugiausiai analizėje pateiktų kriterijų.
4. Pastebėta, kad yra gana sudėtinga įvertinti programinius įrankius atsižvelgiant į jų poveikį organizacijos procesams, be to yra sunku identifikuoti gamintojo suteiktų įrankiui galimybių įvairovę tam tikro projekto kontekste. Nepaisant šių įrankių įvertinimo sudėtingumo, be abejonės galima teigti, jog dideliuose projektuose reikalavimų specifikavimo įrankiai suteikia daug našesnę ir produktyvesnę darbą, kurį atlikus gaunamas produktas, geriau ir tiksliau patenkinantis užsakovo norus ir poreikius.
5. Sukurtas Volere šablono pritaikymo Doors reikalavimų įrankyje modelis. Taip pat sukurtas Volere reikalavimų adaptuotos surinkimo formos modelis. Modelis realizuojamas pritaikant reikalavimų specifikavimo programinės įrangos įrankį Telelogic Doors. Sukurto surinkimų formos modelio pagalba, sistema dalį vartotojo funkcijų automatizuoja, taip sudarydama didesnę vartotojo darbo našumą ir efektyvumą reikalavimų specifikavimo procese.
6. Sukurtas pradinis projektuoto modelio realizacijos prototipas, panaudojant Telelogic Doors modulių kūrimo funkcijas ir išplėtimo programavimo kalbą DXL (Doors eXtension Language). Įdiegtas Volere šablonas, bei sukurtas surinkimų formos prototipas.
7. Atliktas reikalavimo specifikavimo proceso veiklų palyginimas automatizuotu ir neautomatizuotu reikalavimų surinkimo būdu. Palyginus paaiškėjo, kad sukurto prototipo pagalba 30% sumažinamas vartotojo veiksmų kiekis, iš ko seka padidėjęs reikalavimų surinkimo proceso našumas ir efektyvumas.

## LITERATŪROS SĄRAŠAS

1. Šilingas, Darius. Programinės įrangos kūrimo metodologija. Paskaitų medžiaga. 2005m
2. Volere Requirements Specification template. [žiūrėta 2006-09-18]. Prieiga internete: <http://www.volere.co.uk/template.htm>
3. Richardson, Debra. Desirable Characteristics of a Requirements Specification. [žiūrėta 2006-09-18]. Prieiga internete: <http://www.ics.uci.edu/~djr/classes/ics121/Topic3/006.htm>
4. IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998. Publisher: The Institute of Electrical and Electronics Engineers, Inc. ISBN 0-7381-0332-2. Year of Publication: 1998. [žiūrėta 2006-11-05]
5. Butleris, Rimantas; Danikauskas, Tomas. Reikalavimo specifikavimo Oracle case terpėje plėtra. [žiūrėta 2006-11-06]. Prieiga internete: <http://www.leidykla.vu.lt/inetleid/inf-mok/19/str6.html>
6. IncoSE requirements management tools survey. [žiūrėta 2006-12-10]. Prieiga internete: <http://www.paper-review.com/tools/rms/read.php>
7. Requirements engineering tools. [žiūrėta 2007-01-28] Prieiga internete: <http://easyweb.easynet.co.uk/%7Eiany/other/vendors.htm>
8. Software requirements definition standards. [žiūrėta 2007-02-01]. Prieiga internete: <http://www.12207.com/requirements.htm>
9. Moore, Jim. Software Life – Cycle Standards. [žiūrėta 2007-02-01]. Prieiga internete: <http://www.acm.org/tsc/lifecycle.html>
10. Robertson, Suzanne; Robertson, James. Mastering the Requirements Process second edition. Publisher: Addison Wesley Professional. Date of Publication: March 17, 2006, Print ISBN-10: 0-321-41949-9, Print ISBN-13: 978-0-321-41949-1. [žiūrėta 2007-01-09].
11. Wiegers, Karl Eugene. Software Requirements, Second Edition. Publisher: Microsoft Press. Year of Publication: 2003, ISBN:0735618798. [žiūrėta 2007-03-17].
12. Alexander, . Ian F.; Stevens, Richard. Writing Better Requirements. ISBN:0321131630 Year of Publication: 2002. [žiūrėta 2007-04-26].
13. IBM Rational RequisitePro Release Notes, 7.0, Windows. [žiūrėta 2007-04-26]. Prieiga internete: <http://www.elink.ibm.com/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=GI11-6382-00>
14. Business Benefits of IRqA PRL White Paper. [žiūrėta 2007-09-27]. Prieiga internete <http://www.programmingresearch.com/>
15. Requirements Management for Advanced Systems and Software Development. [žiūrėta 2007-10-06]. Prieiga internete: <http://www.telelogic.com/Products/doors/doors/index.cfm>
16. Hapl Context and Index, Telelogic Doors. [žiūrėta 2008-04-28].
17. DXL Reference Help. Telelogic Doors. [žiūrėta 2008-05-05].



## PRIEDAS NR.1 Volere reikalavimų šablono programinis kodas.

```
// Volere Reikalavimų Specifikavimo šablonas (Volere)
/*
*/
Template Volere = template //
"Projekto varovai (Project Drivers)" <<
  "Sistemos paskirtis" <<
    "Projekto kurimo pagrindas (pagrindimas)" <>
    "Sistemos tikslai (paskirtis)" >>
  "Užsakovai, pirkejai ir kiti sistema suinteresuoti
asmenys" <>
  "Vartotojai" >>
"Projekto Apribojimai" <<
  "Ipareigojantys apribojimai" <<
    "Apribojimai sprendimui" <>
    "Diegimo aplinka" <>
    "Bendradarbiaujancios sistemos" <>
    "Komeraciniai specializuoti programu paketai" <>
    "Numatoma darbo vietos aplinka" <>
    "Sistemos kurimo terminai" <>
    "Sistemos kurimo biudžetas" >>
  "Terminu žodynas" <>
  "Svarbus faktai ir prielaidos" >>
"Funkciniai reikalavimai" <<
  "Veiklos sudetis (The scope of the work) / darbo
užmojai" <<
    "Veiklos kontekstas" <>
    "Veiklos padalinimas-paskirstymas" >>
  "Sistemos sudetis (The scope of the product)" <<
    "Sistemos ribos" <>
    "" >>
  "Funkciniai reikalavimai ir reikalavimai duomenims" <<
    "Funkciniai reikalavimai" <>
    "Reikalavimai duomenims" >> 1 >>
"Nefunkciniai reikalavimai" <<
  "Reikalavimai sistemos išvaizdai (Look and feel)" <>
  "Reikalavimai panaudojamumui (Usability)" <>
  "Reikalavimai vykdymo charakteristikoms
(Performance)" <>
  "Reikalavimai veikimo salygoms (Operational)" <>
  "Reikalavimai sistemos priežiurai (Maintainability and
portability)" <>
  "Reikalavimai saugumui (Security)" <>
  "Kultūriniai-politiniai reikalavimai" <>
  "Teisiniai reikalavimai" >>
"Projekto išėiga (Project issues)" <<
  "Atviri klausimai (problemos)" <>
  "Egzistuojantys sprendimai (Off-the-Shelf Solutions)" <<
    "Pagamintos sistemos, kurios gali buti nupirktos" <>
    "Pagaminti komponentai, kurie gali buti panaudoti" <>
    "Galimas pakartotinas panaudojimas" >>
  "Naujos problemos" <<
    "Poveikis egzistuojanciai aplinkai" <>
    "Itaka jau instaliuotoms sistemoms" <>
    "Neigiamas vartotoju nusiteikimas" <>
    "Kliudantys diegimo aplinkos apribojimai" <>
    "Galimos naujos sistemos sukeltos problemos" >>
  "Uždaviniai" <<
    "Sistemos pateikimo žingsniai (etapai)" <>
    "Vystymo etapai" >>
  "Pritaikymas (Cutover)" <<
    "Reikalavimai esamu duomeniu perkėlimui" <>
    "Reikalingas duomeniu transformavimas perkėliant i
nauja sistema" >>
  "Rizikos" <<
    "Galimos sistemos kurimo rizikos" <>
    "Atsitiktinumu (riziku) planas" >>
  "Kaina" <>
  "Vartotojo dokumentacija ir apmokymas" <>
  "Perspektyviniai reikalavimai (Waiting room)" <>
  "Idejos ir sprendimai (Ideas for solutions)"
instance Volere
```

## PRIEDAS NR.2 Volere reikalavimų surinkimo formos programinis kodas.

```
DB reqBox = create "Volere reikalavimu surinkimo forma"
string tipas[] = {
  "Funkcinis",
  "Sistemos isvaizdos",
  "Panaudojamumo",
  "Valdymo charakteristikos",
  "Veikimo salygos",
  "Sistemos prieziuros",
  "Saugumo",
  "Kulturos ir politikos",
  "Teisinis"
}
string funkcinio_tipas[] = {
  "Veiklos kontekstas",
  "Veiklos paskirstymas",
  "Sistemos ribos",
  "Funkciniai reikalavimai ir reikalavimai duomenims",
  "Funkciniai reikalavimai",
  "Reikalavimai duomenims"
}
string lfeel_tipas[] = {
  "4.1.1. Vartotojo sasaja",
  "4.1.2. Stilius"
}
string panaudojamumo_tipas[] = {
  "4.2.1. Panaudojimo paprastumas",
  "4.2.2. Pritaikymas asmeniui ir nacionaliniai
reikalavimai"
}
string nasumo_tipas[] = {
  "4.3.1. Greicio ir velavimu reikalavimai",
  "4.3.2. Saugumo kritiniai reikalavimai",
  "4.3.3. Tikslumo reikalavimai",
  "4.3.4. Patikimumo ir tinkamumo naudoti
reikalavimai",
  "4.3.5. Stiprumo ir klaidos tolerancijos reikalavimai",
  "4.3.6. Talpumo reikalavimai"
}
string veikimosalygos_tipas = {
  "4.4.1. Fazines alpinks reikalavimai",
  "4.4.2. Reikalavimai saveikai su gretimomis
sistemomis" ,
  "4.4.3. Produkcijos (producttization) reikalavimai" ,
  "4.4.4. Isleidimu (releases) reikalavimai"
}
string sprieziura_tipas = {
  "4.5.1. Palaikymu (maintainnability) reikalavimai",
  "4.5.2. Palaikymu (supportability) reikalavimai" ,
  "4.5.3. Prisitaikymo reikalavimai"
}
string saugumas_tipas = {
  "4.6.1. Naudojimosi teises",
  "4.6.2. Vientisumas",
  "4.6.3. Privatumas",
  "4.6.4. Imunitetas"
}
const string reikTipai = "Reikalavimu tipai"
const string konfliktuoja = "Konfliktai"
const string priklaus = "Priklausomybe"
const string aOH = "Object Heading"
const string uc = "Panaudos atvejai(PA)"
DBE tipoImport = choice(reqBox, "Reikalavimo tipas: ",
tipas, 70)
DBE import = choice(reqBox, " ", subtipas, 2)
DBE aprasymas = text(reqBox, "Aprasymas:", "", 50,
false)
```

```
DBE pagrindimas = text(reqBox, "Pagrindimas:", "", 50, false)
```

```
DBE panaudosa = field(reqBox, "Panaudojimo atvejai", "", 35, false)
```

```
DBE saltinis = field(reqBox, "Saltinis", "", 15)
```

```
DBE tikimoKriterijus = text(reqBox, "Tikimo kriterijus:", "", 50, false)
```

```
string importance[] = {"1", "2", "3", "4", "5" }
```

```
DBE reqImport = choice(reqBox, "Uzsakovo tenkinimas:", importance, 2)
```

```
DBE reqImport2 = choice(reqBox, "Uzsakovo netenkinimas:", importance, 2)
```

```
DBE priklausomybes = field(reqBox, "Priklausomybes:", "", 35, false)
```

```
DBE conflicts = field(reqBox, "Konfliktai:", "", 35, false)
```

```
void subtipas () {
```

```
string req = get tipoImport
```

```
if (req == "Funkcinis")  
then {subtipas = funkcinio_tipas }
```

```
if (req == "Sistemas isvaizdos")  
then { subtipas = lfeel_tipas }
```

```
if (req == "Panaudojamumo")  
then {subtipas = panaudojamumo_tipas }
```

```
if (req == "Nasumo")
```

```
then {subtipas = nasumo_tipas }
```

```
if (req == "Veikimo salygos")  
then {subtipas = veikimosalygos_tipas }
```

```
if (req == "Sistemas prieziuros")  
then {subtipas = sprieziura_tipas }
```

```
if (req == "Saugumo")  
then {subtipas = saugumas_tipas }
```

```
}
```

```
void sarysis (Module reikalavimai, string aprasymas, ID, bool aInherit) {
```

```
Module currMod = current  
current = linkmod
```

```
if !exists ID then  
create linkmod ID inherit aprasymas  
current = currMod  
else linkmod = !null  
break
```

```
}
```

```
void siusti(DB reqBox) {
```

```
string req2 = get subtipas
```

```
string ap = get aprasymas
```

```
string pa = get panaudosa
```

```
string pagr = get pagrindimas
```

```
string salt = get saltinis
```

```
string tikimokrit = get tikimoKriterijus
```

```
int i = get reqImport
```

```
int j = get reqImport2
```

```
string priklaus = get priklausomybes
```

```
string konfliktai = get conflicts
```

Object findtipas (Module reikalavimai) {

filtering off

Object rtip = null

bool found = false

string s

for rtip in m do {

    s = rtip.req2

    if s == req2 then {

        found = true

        current = rtip

        break

    }

}

if null rtip then return req

}

Object o = create last below rtip

o.reiktip = req

o.aOH = ap

o.pagrindimas = pagr

o.saltinis = salt

o.tikimas = tikimokrit

o.tenkinimas = i

o.netenkinimas = j

o.uc = pa

o.priklausomybe = priklaus

o.conflicts = konfliktai

sarysis (reikalavimai, conflicts, ID, true)

sarysis (reikalavimai, priklaus, ID, true)

sarysis (use cases, uc, ID, true)

}

apply(reqBox, "Ivesti", siusti)

show reqBox