

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

**Paulius Indriūnas**

**PROGRAMINĖS ĮRANGOS DEFEKTŲ, SAVYBIŲ  
IR REIKALAVIMŲ VALDYMO SISTEMA**

Magistro darbas

**Vadovas  
prof. S. Gudas**

**KAUNAS, 2006**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

**PROGRAMINĖS ĮRANGOS DEFEKTŲ, SAVYBIŲ  
IR REIKALAVIMŲ VALDYMO SISTEMA**

Magistro darbas

Vadovas  
prof. S. Gudas

Recenzentas  
doc. dr. S. Maciulevičius

Atliko  
IFM 0/4 gr. stud.  
P. Indriūnas

**KAUNAS, 2006**

## **Software Defect, feature and requirements management system**

### **Summary**

Software development is an iterative process which is based on teamwork and information exchange. In order to keep this process running, proper informational flow control techniques in a software development company have to be applied. As number of employees grows, manual control of this process becomes ineffective and automated solutions takes over this task. The most common informational units in the software development process are defects, new features and requirements. This paper addresses problems that occur during software development process and intends to create techniques and methodologies for software defect, feature and requirement management.

Following tasks are performed while automating mentioned process:

- Analysis of software development company workflow;
- Design and documentation of software defect, feature and requirement management method, which would ease information exchange in design, implementation and quality assurance stages of software development process;
- Analysis and comparison of existing similar solutions. Identification of most attractive and useful features;
- Design and implementation of software defect, feature and requirement management system which covers newly designed information flow control method.

Practical task includes designing, programming, installing and testing newly created system. Design and development is based on reusable component approach which results in several layers of reusable code.

## Turinys

1. ĮVADAS.....	7
1.1. Uždaviniai.....	7
2. ANALIZĖS DALIS.....	9
2.1 Tyrimo sritis ir objektas.....	9
2.2 Analizės metodų ir priemonių parinkimas.....	10
2.3 Organizacijos veiklos analizė.....	11
2.3.1 Programinę įrangą kuriančios organizacijos veiklos sąveikos.....	11
2.3.2 Programinę įrangą kuriančios organizacijos tikslai.....	12
2.3.3 Programinę įrangą kuriančios organizacijos panaudojimo atvejai.....	13
2.3.4 Programinę įrangą kuriančios organizacijos veiklos objektai.....	14
2.3.5 Programinę įrangą kuriančios organizacijos struktūra.....	15
2.3.6 Programinės įrangos kūrimo procesas.....	16
2.3.7 Defektų, savybių ir reikalavimų valdymo sistemos vartotojų reikalavimai.....	17
2.3.8 Defektų, savybių ir reikalavimų valdymo sistemos nefunkciniai reikalavimai.....	18
2.3.9 Programinę įrangą kuriančios organizacijos veiklos analizės išvados.....	19
2.4 Pasaulinio lygio analogiškų sistemų lyginamoji analizė.....	19
2.4.1 Vertinimo kriterijai.....	19
2.4.2 Defektų valdymo sistema „PRTracker“.....	21
2.4.3 Defektų ir savybių valdymo sistema „OnTime“.....	25
2.4.4 Reikalavimų valdymo sistema „Reconcile“.....	29
2.4.5 Reikalavimų valdymo sistema „DOORS“.....	31
2.4.6 Lyginamosios analizės rezultatai.....	35
2.5 Literatūros šaltiniuose pateikti sprendimai problemai spręsti.....	36
2.6 Projektavimo metodų ir priemonių parinkimas.....	37
2.8 Analizės išvados.....	37
3. DEFEKTŲ, SAVYBIŲ IR REIKALAVIMŲ VALDYMO METODAS.....	38
4. PROJEKTO DALIS.....	58
4.1 Sistemai keliamų funkcinių ir nefunkcinių reikalavimų modelis.....	58
4.1.1 Defektų, savybių ir reikalavimų sistemos vartotojų panaudos atvejų diagramos.....	58
4.1.3 Dalykinės srities klasių diagrama.....	78
4.1.4 Vartotojų interfeiso modelis.....	79
4.1.5 Nefunkciniai reikalavimai.....	81
4.2 Projekto modelis.....	82
4.2.1 Projekto tikslas.....	82
4.2.2 Sistemos architektūra.....	82
4.2.3 Duomenų bazės modelis.....	86
4.2.4 Realizacijos modelis.....	93
4.2.5 Reikalavimai sistemos funkcionavimo palaikymui.....	96
4.2.6 Sistemos naudojimo instrukcija.....	97
4.3 Projekto išvados.....	109
IŠVADOS.....	110
LITERATŪRA:.....	111

## Lentelių sąrašas

2.1 lentelė. Vartotojų keliamų funkcinių reikalavimų lentelė.....	17
2.2 lentelė. Vartotojų keliamų nefuncinių reikalavimų lentelė.....	18
2.3 lentelė. Analogiškų sistemų palyginimo vertinimo kriterijų lentelė .....	20
2.4 lentelė. Analogiškų sistemų palyginimo lentelė .....	35
2.5 lentelė. Analogiškų sistemų palyginimo rezultatai.....	35
4.1 lentelė. Nefuncinių reikalavimų lentelė .....	81
4.2 lentelė. Pakartotinai panaudotini veiklos komponentai .....	84
4.3 lentelė. Pakartotinai panaudotini vartotojo sąsajos komponentai.....	84
4.4 lentelė. Projekto komponentai .....	85
4.5 lentelė. Reikalavimai sistemos funkcionalumo palaikymui .....	96

## Paveikslų sąrašas

1.1 pav. Programinės įrangos defektų, savybių ir reikalavimų valdymo sistemos pritaikymas .....	8
2.1 pav. Programinę įrangą kuriančios organizacijos veiklos sąveikų modelis ( <i>paketu diagrama</i> ) .....	11
2.2 pav. Programinę įrangą kuriančios organizacijos tikslų modelis .....	12
2.3 pav. Programinę įrangą kuriančios įmonės veiklos panaudojimo atvejų modelis .....	13
2.4 pav. Programinę įrangą kuriančios organizacijos veiklos objektų modelis.....	14
2.5 pav. Programinę įrangą kuriančios organizacijos struktūros modelis.....	15
2.6 pav. Programinės įrangos kūrimo procesų modelis.....	16
2.7 pav. Paketo „PRTracker“ pagrindinis langas.....	21
2.8 pav. Naujo projekto sukūrimas .....	22
2.9 pav. Defekto istorija .....	23
2.10 pav. Defektų filtravimo užklausos sudarymas.....	24
2.11 pav. Paketo „OnTime“ pagrindinis langas.....	25
2.12 pav. Defekto sukūrimo forma .....	26
2.13 pav. Defekto informacijos redagavimo forma .....	26
2.14 pav. Naujos savybės sukūrimo forma .....	27
2.15 pav. Savybės informacijos redagavimo forma .....	27
2.16 pav. Projekto sukūrimo dialogas .....	28
2.17 pav. Sistemos vartotojų valdymo langas.....	28
2.18 pav. „Reconcile“ reikalavimų valdymo sistemos pagrindinis langas .....	29
2.19 pav. Reikalavimų sekimo matrica .....	30
2.20 pav. Sistemos DOORS vartotojai.....	31
2.21 pav. Reikalavimų / reikalavimų specifikacijos redagavimas.....	32
2.22 pav. Reikalavimų struktūros peržiūros langas .....	33
2.23 pav. Reikalavimų sąrašo peržiūrėjimas naršyklės lange.....	33
2.24 pav. Reikalavimų valdymo sistema DOORS integruota į MS Word teksto redaktorių.....	34
2.25 pav. Reikalavimų specifikavimo modelis.....	36
3.1 pav. Defektų, savybių ir reikalavimų valdymo metodo etapų tinklinė diagrama.....	38
4.1 pav. Informacinės sistemos vartotojų grupės.....	58
4.2 pav. Sistemos vartotojo panaudos atvejų diagrama.....	59
4.3 pav. Testuotojo panaudos atvejų diagrama.....	60
4.4 pav. Programuotojo panaudos atvejų diagrama.....	61
4.5 pav. Projekto vadovo panaudos atvejų diagrama .....	62

4.6 pav. Projektuotojo panaudos atvejų diagrama .....	63
4.7 pav. Sistemos administratoriaus panaudos atvejų diagrama.....	63
4.8 pav. Defektų, savybių ir reikalavimų dalykinės srities klasės .....	78
4.9 pav. Pagrindinė programos forma .....	79
4.10 pav. Projekto sukūrimo / redagavimo forma .....	80
4.11 pav. Defekto sukūrimo / redagavimo forma .....	80
4.12 pav. Prisijungimo prie sistemos langas .....	81
4.13 pav. Defektų, savybių ir reikalavimų valdymo sistemos komponentų modelis.....	83
4.14 pav. Defektų, savybių ir reikalavimų valdymo sistemos duomenų bazės modelis.....	86
4.15 pav. Defektų, savybių ir reikalavimų valdymo sistemos komponentų modelis.....	94
4.16 pav. Defektų, savybių ir reikalavimų sistemos „Softrack“ diegimo diagrama.....	95
4.17 pav. Prisijungimo langas prie sistemos „Softrack“ .....	97
4.18 pav. Pagrindinis programos langas po prisijungimo .....	97
4.19 pav. Programos „Softrack“ defektų valdymo langas .....	98
4.20 pav. Projektų medžio ir defektų valdymo kontekstiniai meniu .....	99
4.21 pav. Projekto sukūrimo / redagavimo dialogas .....	100
4.22 pav. Projekto sukūrimo / redagavimo dialogas .....	100
4.23 pav. Programos „Softrack“ savybių valdymo langas.....	101
4.24 pav. Projektų medžio ir savybių valdymo kontekstiniai meniu .....	101
4.25 pav. Savybės sukūrimo / redagavimo dialogas.....	102
4.26 pav. Programos „Softrack“ reikalavimų valdymo langas.....	103
4.27 pav. Projektų medžio ir reikalavimų valdymo kontekstiniai meniu.....	103
4.28 pav. Reikalavimo sukūrimo / redagavimo dialogas .....	104
4.29 pav. Papildomos informacijos (failo) prisegimo prie redaguojamo objekto lentelė .....	105
4.30 pav. Reikalavimo susiejimo su objektu lentelė .....	105
4.31 pav. Reikalavimų ryšių lentelė .....	106
4.32 pav. Reikalavimo paieškos langas.....	106
4.33 pav. Sistemos vartotojų valdymo langas.....	107
4.34 pav. Vartotojų valdymo kontekstinis meniu .....	107
4.35 pav. Sistemos vartotojo sukūrimo / redagavimo dialogas .....	108

## 1. ĮVADAS

Programinės įrangos kūrimas – iteratyvus procesas, kurio metu dažnai kinta ir atsiranda nauji reikalavimai, pakinta pradinė kuriamos sistemos architektūra, todėl neišvengiamai atsiranda klaidos. Labai dažnai problemų ir klaidų atsiradimo priežastimi tampa informacijos stygius arba jos šaltinių nepasiekiamumas. Siekiant sumažinti klaidų skaičių, būtina automatizuoti informacijos keitimosi procesus, projektų informaciją kaupti visiems, programinę įrangą kuriančios organizacijos, darbuotojams pasiekiamose duomenų bazėse.

Šios problemos aktualumas ir įtaka programinę įrangą kuriančios organizacijos darbui, labai priklauso nuo joje dirbančių žmonių skaičiaus. Kuo daugiau darbuotojų, tuo daugiau informacijos sukurama, tuo sunkiau sekasi ja keistis, ją apdoroti ir saugoti. Vidutinio dydžio įmonėse šios problemos gali būti išspręstos įdiegiant kompiuterizuotą informacijos valdymo sistemą. Didelėse kompanijose į informacijos valdymo sistemas investuojami milijonai, jos yra reguliariai atnaujinamos, tobulinamos ir prižiūrimos šimtų darbuotojų.

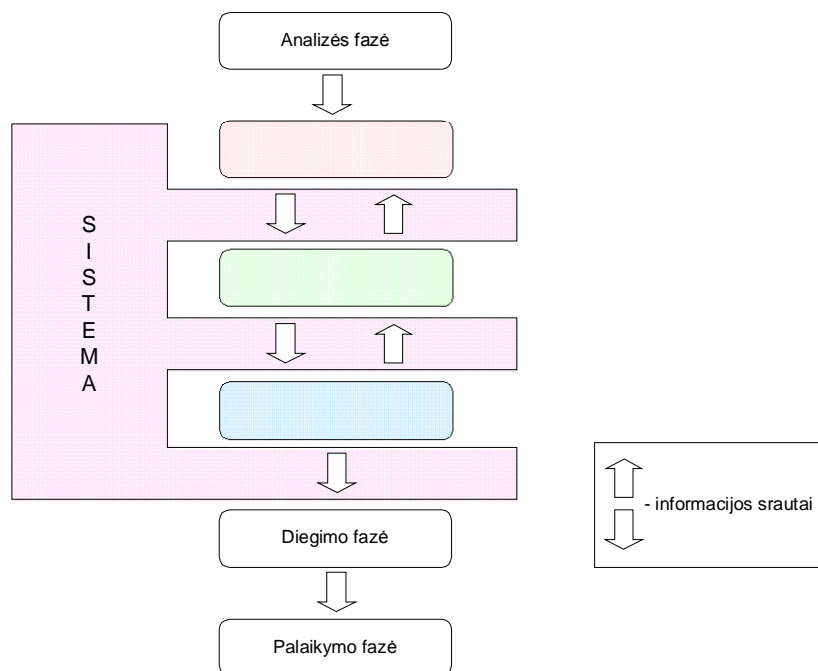
Šiame darbe analizuojama vidutinio dydžio programinę įrangą kurianti organizacija, joje vykstantys procesai ir programinės įrangos projektavimo, realizavimo ir kokybės užtikrinimo fazės. Informacijos kaupimui ir jos srautų valdymui tarp šių trijų projekto fazių pritaikomas naujai sukurtas programinės įrangos defektų, savybių ir reikalavimų valdymo metodas.

### 1.1. Uždaviniai

Programinės įrangos projektavimo, realizavimo ir kokybės užtikrinimo fazėse įmonės darbuotojai kuria, apdoroja ir keičiasi kelių rūšių informacija:

- Reikalavimai, jų specifikacijos ir modeliai;
- Įvairūs nauji pasiūlymai ir pakeitimai sistemos funkcionalumui (savybės);
- Testavimo planai, įvairios ataskaitos;
- Defektai (programuotojų klaidos, neatitikimai standartams ir pan.), jų atkūrimo informacija;

Siekiant patobulinti programinės įrangos kūrimo procesą ir palengvinti kompanijos darbuotojų darbą, ši informacija turi būti susisteminta ir išdalinta už ją atsakingiems asmenims. Šiai užduočiai automatizuoti turi būti sukurtas defektų, savybių, reikalavimų ir su jais susijusios informacijos valdymo metodas, kurio pagrindu būtų sukurta programinės įrangos defektų, savybių ir reikalavimų valdymo sistema. Tokios sistemos darbą galima iliustruoti 1.1 paveiksle pateikta schema:



**1.1 pav. Programinės įrangos defektų, savybių ir reikalavimų valdymo sistemos pritaikymas**

Schemoje (pav. 1.1) pavaizduota sistemos pritaikymo sritis programinės įrangos kūrimo procese. Pagrindinė tokios sistemos funkcija yra informacijos saugojimas ir duomenų srautų valdymas tarp projektavimo, realizavimo ir kokybės užtikrinimo fazių.

Norint sukurti ir kompiuterizuoti minėtąjį defektų, savybių ir reikalavimų valdymo metodą bei jį realizuoti, reikia atlikti tokius uždavinius:

- Išanalizuoti programinę įrangą kuriančios organizacijos veiklą ir pritaikyti jai kuriamą programinės įrangos defektų, savybių ir reikalavimų valdymo metodą.
- Sumodeliuoti ir aprašyti programinės įrangos defektų, savybių ir reikalavimų valdymo metodą, kuris palengvintų įmonės darbuotojų darbą projektavimo, realizavimo ir kokybės užtikrinimo fazėse.
- Išnagrinėti keletą analogiškų pasaulinio lygio sprendimų ir panaudoti geriausias jų savybes projektuojant programinės įrangos defektų, savybių ir reikalavimų valdymo sistemą.
- Suprojektuoti ir realizuoti valdymo metodo informacijos srautų valdymą kompiuterizuojančią informacinę sistemą, sukurti sistemos vartotojo vadovą.

Praktinis šio darbo tikslas – sukurti klaidų, naujų programinės įrangos savybių ir reikalavimų valdymo sistemą, programinę įrangą kuriančiai organizacijai, kuria naudodamiesi šios organizacijos darbuotojai galėtų keistis informacija bei lengvai ir efektyviai valdyti klaidų registravimo, jų taisymo bei naujų savybių ir reikalavimų realizavimo procesą.



## 2. ANALIZĖS DALIS

Analizės tikslas – apibrėžti šio darbo tyrimo sritį, objektą, išnagrinėti organizacijos veiklą, atlikti problemos sprendimo metodų bei panašių programinės įrangos projektų lyginamąją analizę ir pasirinkti geriausias ir labiausiai šiam darbui tinkančias jų savybes.

Analizės dalyje pateikta:

- Organizacijos veiklos ir pagrindinių procesų analizė;
- Pasaulinio lygio, analogiškų sistemų lyginamoji analizė;
- Nefunkciniai ir vartotojų reikalavimai;
- Literatūroje pateiktų problemų susijusių su defektų ir savybių valdymo sistemomis analizė.

Šiame darbe keliamą problema yra defektų minimizavimas ir naujų reikalavimų bei savybių realizavimas per įmanomai kuo trumpesnę laiką. Vienas iš šios problemos sprendimo būdų ir yra defektų, savybių ir reikalavimų valdymo sistemos įdiegimas. Produkto pateikimo rinkai laikas, pastaruoju metu, yra vienas svarbiausių programinės įrangos projektų sėkmės kriterijų. Naudojant tokią sistemą, žymiai supaprastėja bendravimas tarp įmonės programuotojų, testuotojų ir projektų vadovų, sutrumpėja klaidų taisymo, naujų reikalavimų ir savybių realizavimo laikas.

### 2.1 Tyrimo sritis ir objektas

Tyrimo sritis – programinės įrangos defektų, savybių ir reikalavimų valdymo sistemų (*angl. Software defect, feature and requirement management systems*), kūrimo metodų nagrinėjimas bei programinės įrangos kūrimo proceso tobulinimas, analizuojant defektų, naujų savybių ir reikalavimų spartaus realizavimo galimybes, pasiekiant maksimalų programinės įrangos funkcionalumo atitikimą vartotojo pateiktiems reikalavimams.

Tyrimo objektas – programinę įrangą kurianti įmonė, jos veiklos analizė bei defektų, savybių ir reikalavimų valdymo sistemos įtaka įmonės darbo efektyvumui.

## 2.2 Analizės metodų ir priemonių parinkimas

Programinę įrangą kuriančios organizacijos veiklai išanalizuoti bus sudaryti tokie veiklos analizės modeliai, naudojant UML:

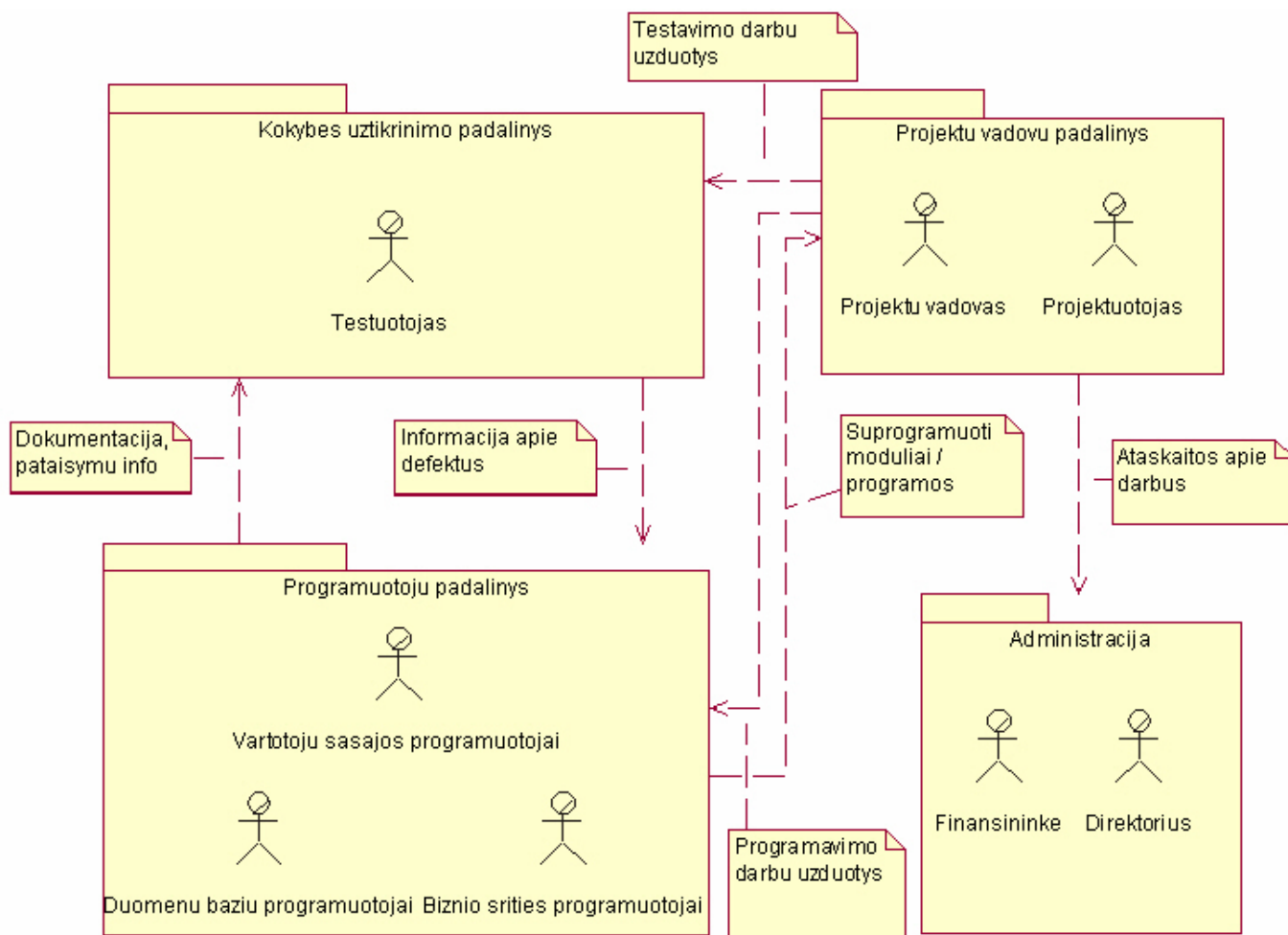
1. Veiklos sąveikų modelis
2. Veiklos tikslų modelis
3. Veiklos panaudojimo atvejų modelis
4. Veiklos objektų modelis
5. Organizacinės struktūros modelis
6. Veiklos procesų modelis

UML modeliavimas yra pranašesnis už struktūrinį projektavimą tuo, kad pagal sudarytus UML modelius, bus galima geriau įsigilinti į organizacijoje vykstančius procesus, lengvai ir vienareikšmiškiau suvokti vartotojų reikalavimus. Struktūrinis projektavimas yra taikytinas, kai organizacijoje vykstantys procesai yra griežčiau apibrėžti, reikalavimai konkretūs ir formalizuoti.

## 2.3 Organizacijos veiklos analizė

### 2.3.1 Programinę įrangą kuriančios organizacijos veiklos sąveikos

2.1 pav. pateiktoje diagramoje, pavaizduotas programinę įrangą kuriančios organizacijos sąveikų modelis. Paketais pavaizduoti pagrindiniai organizacijos padaliniai, rodyklėmis - duomenų srautai tarp jų. Taip pat pavaizduoti kiekvienam padaliniiui priklausantys dalyviai (įmonės darbuotojai).

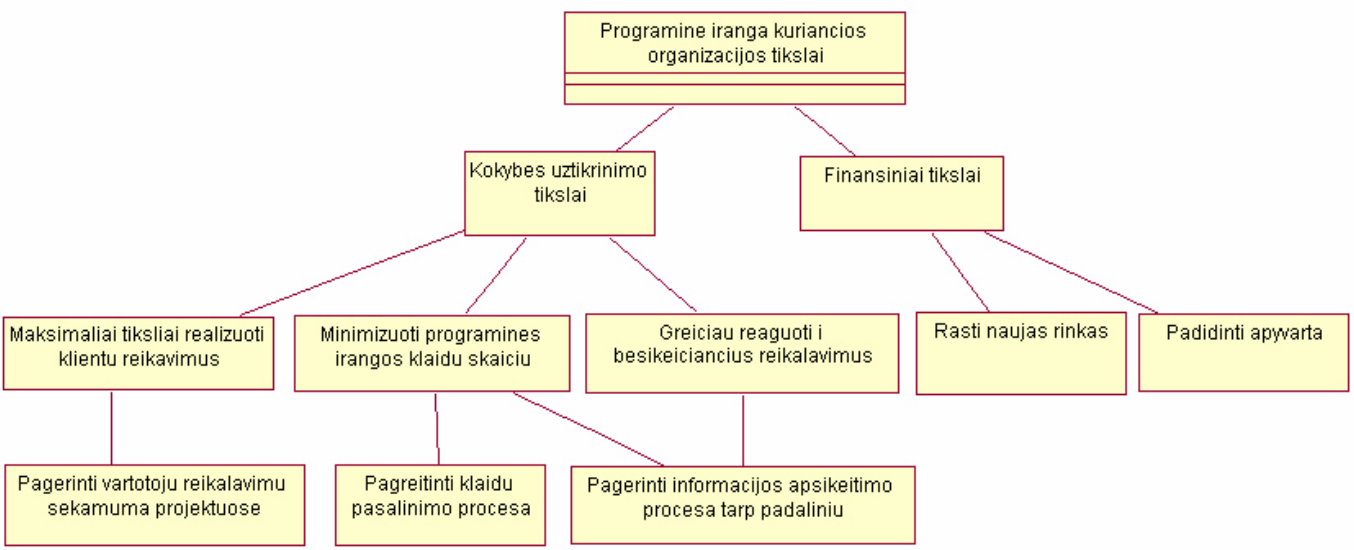


2.1 pav. Programinę įrangą kuriančios organizacijos veiklos sąveikų modelis (*paketų diagrama*)

Šioje diagramoje pavaizduoti pagrindiniai programinę įrangą kuriančios organizacijos padaliniai: *Kokybės užtikrinimo padalinys*, *Programuotojų padalinys*, *Projektų vadovų padalinys* ir *Administracija*. Kokybės užtikrinimo padalinys yra atsakingas už programinės įrangos testavimą, defektų suradimą ir priskyrimą programuotojų padaliniui. Projektų vadovų padalinys atsako už programinės įrangos kūrimo proceso valdymą, projekto modelių sudarymą ir darbų koordinavimą. Administracija tvarko įmonės finansus, analizuoja įvairias darbų atlikimo ataskaitas.

### 2.3.2 Programinę įrangą kuriančios organizacijos tikslai

Nagrinėjamos organizacijos pagrindinis tikslas gerinti programinės įrangos kokybę, minimizuojant klaidų skaičių, tiksliai bei greitai realizuojant besikeičiančius reikalavimus. Organizacijos tikslai pavaizduoti medžio tipo struktūra 2.2 pav.

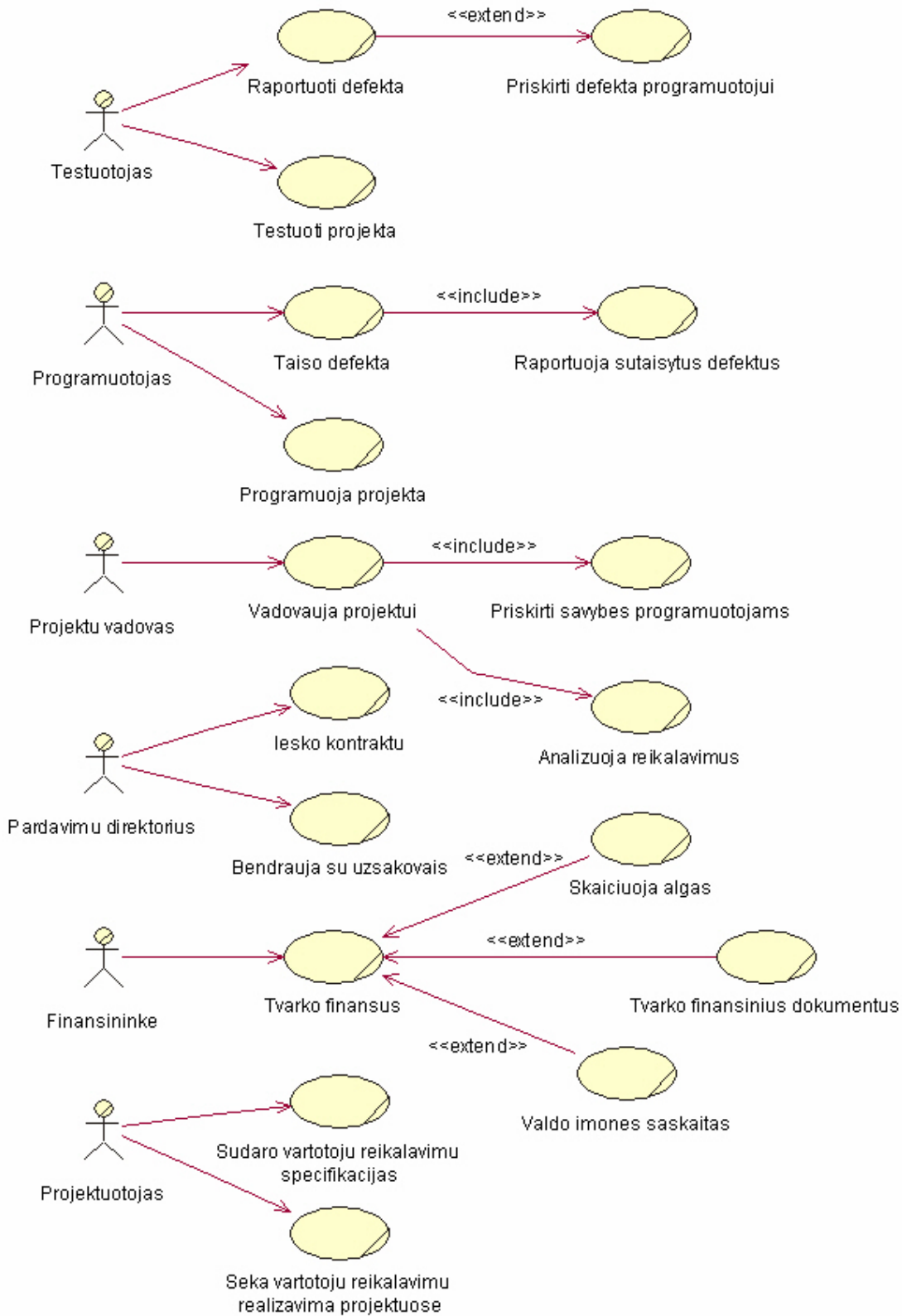


2.2 pav. Programinę įrangą kuriančios organizacijos tikslų modelis

Projektuojant programinę įrangą, organizacijos veiklos tikslų išskyrimas padeda sukonzentruoti dėmesį į svarbiausius iš jų, automatizuoti daugiausiai darbo ir laiko reikalaujančius procesus.

**2.3.3 Programinę įrangą kuriančios organizacijos panaudojimo atvejai**

2.3 paveiksle pavaizduotas organizacijos veiklos panaudojimo atvejų modelis.

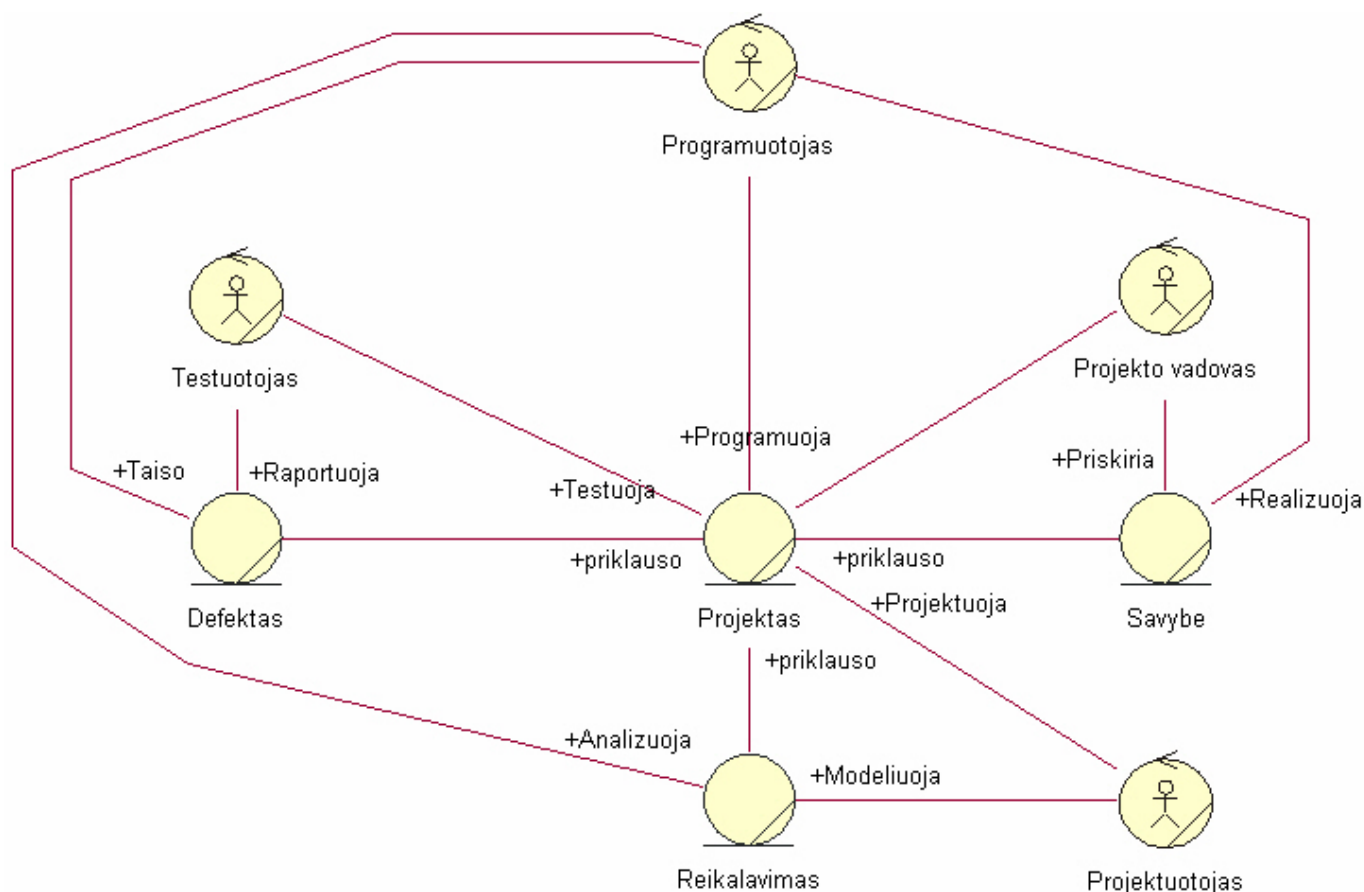


**2.3 pav. Programinę įrangą kuriančios įmonės veiklos panaudojimo atvejų modelis**

Diagramoje dalyviais pavaizduoti įmonės darbuotojai. Panaudojimo atvejai atitinka dalyvių atliekamas darbus. Į ši panaudojimo atvejų modelį įtraukti ir dalyviai tiesiogiai nedalyvaujantys programinės įrangos kūrimo procese, tokie kaip *Finansininkė* ir *Pardavimų direktorius*, kadangi šis panaudojimo atvejų modelis atspindi nekompiuterizuotą įmonės veiklą.

### 2.3.4 Programinę įrangą kuriančios organizacijos veiklos objektai

Šioje diagramoje (2.4 pav.) pavaizduota kaip veiklos objektai sąveikauja tarpusavyje. Pagrindinės veiklos esybės, su kuriomis atliekami svarbiausi veiksmai yra *Projektas*, *Defektas*, *Savybė* ir *Reikalavimas*. Veiksnius su esybėmis atlieka veiklos darbuotojai: *Projekto vadovas*, *Programuotojas*, *Testuotojas*, *Projektuotojas*.

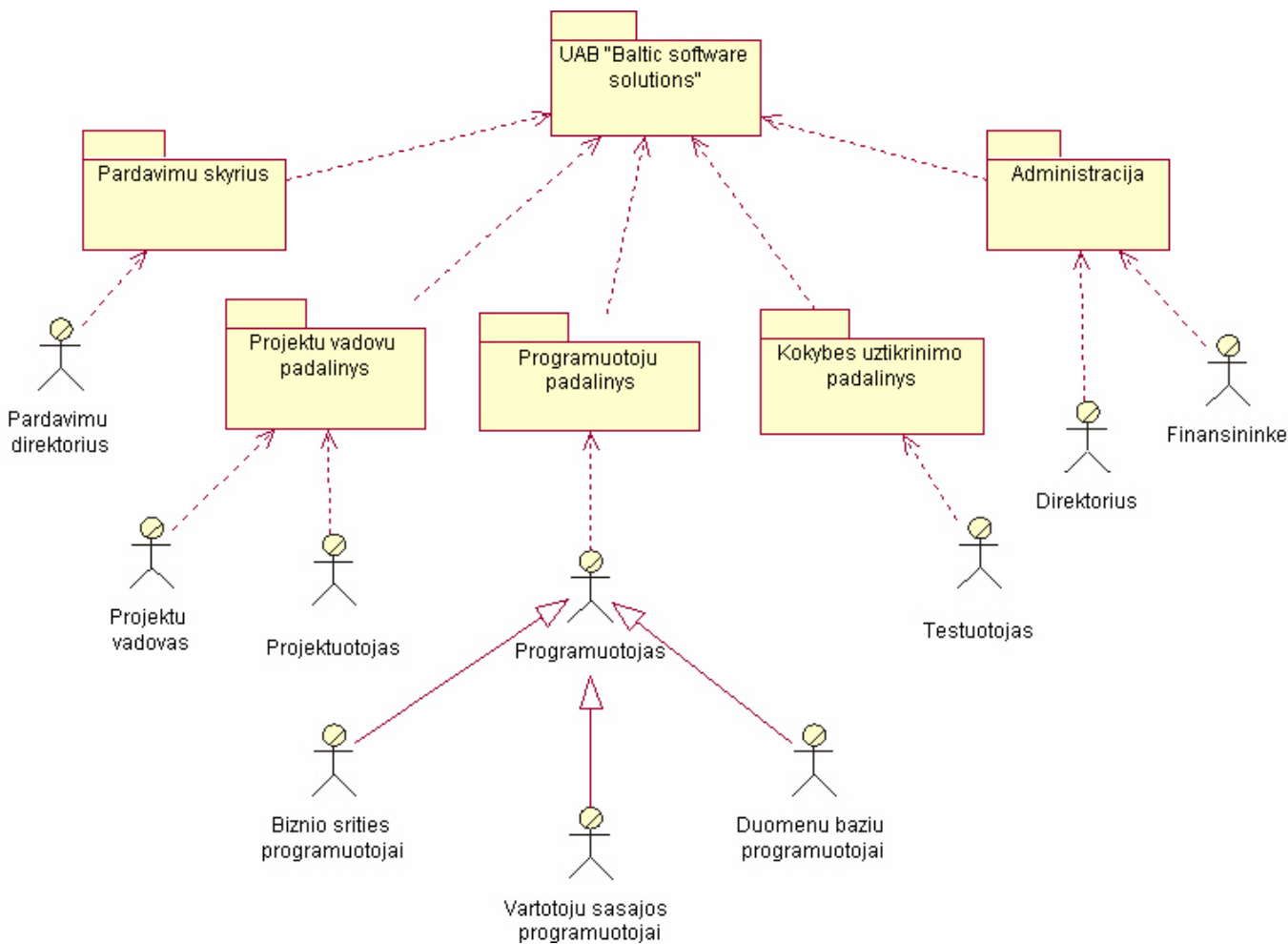


2.4 pav. Programinę įrangą kuriančios organizacijos veiklos objektų modelis

Pagal šiuos veiklos objektus ir jų tarpusavio ryšius, galima sudaryti abstraktų informacinės sistemos architektūros, bei esybių – ryšių modelį.

### 2.3.5 Programinę įrangą kuriančios organizacijos struktūra

2.5 paveiksle pateikta programinę įrangą kuriančios organizacijos struktūra. Joje atvaizduoti pagrindiniai padaliniai ir juose dirbantys darbuotojai.

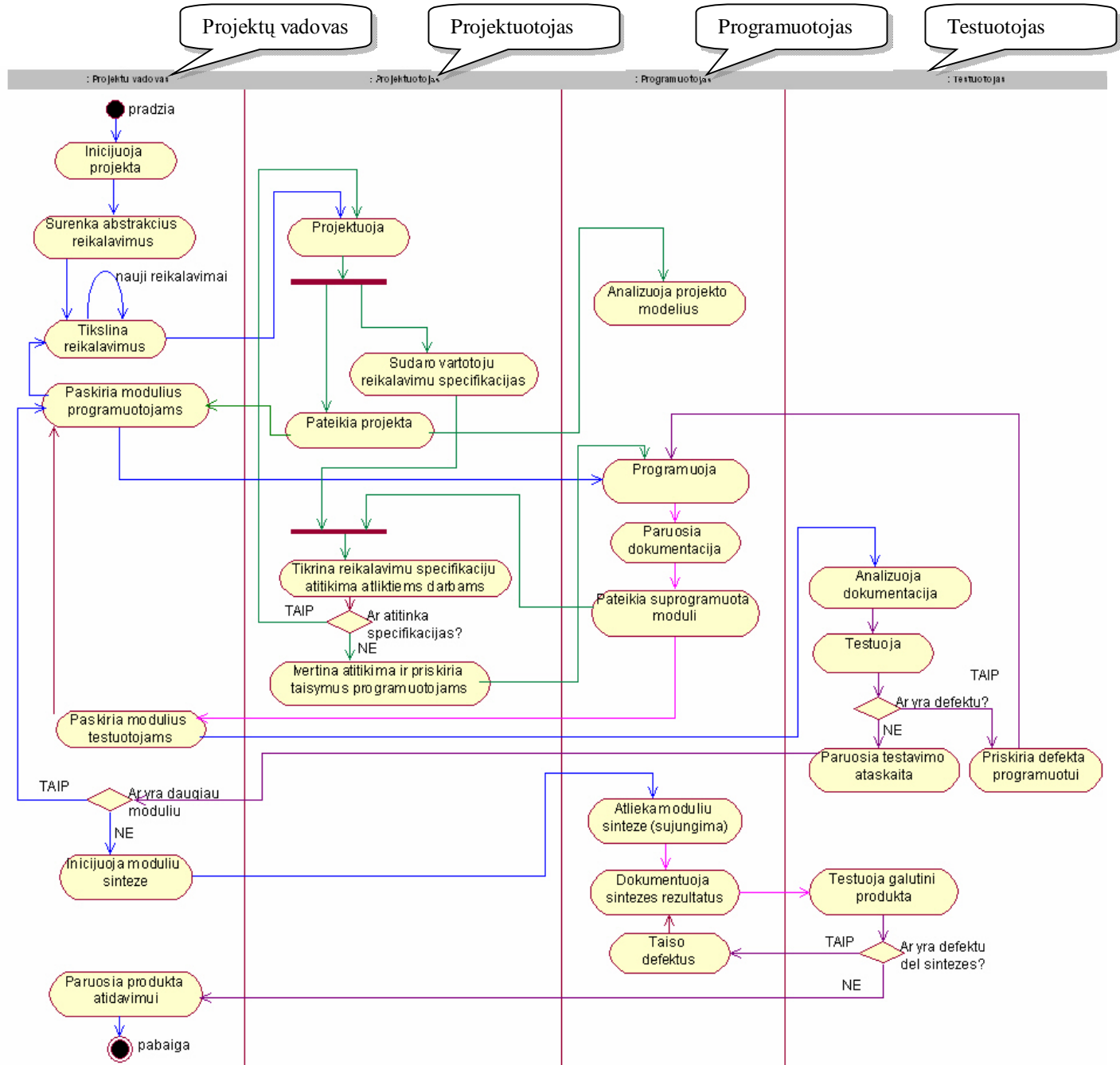


**2.5 pav. Programinę įrangą kuriančios organizacijos struktūros modelis**

Šio modelio pagalba galima nemažai sužinoti apie pačią organizaciją, jos veiklą ir kokių profesijų žmonės joje dirba.

### 2.3.6 Programinės įrangos kūrimo procesas

Šioje diagramoje (2.6 pav.) pateiktas programinės įrangos kūrimo procesas. Joje pavaizduoti keturi įmonės darbuotojai: *Projektų vadovas*, *Projektuotojas*, *Programuotojas* ir *Testuotojas*. Kiekvieno dalyvio veiklos juostoje pavaizduoti atliekami veiksmai ir perėjimai tarp jų.



2.6 pav. Programinės įrangos kūrimo procesų modelis



Šiame modelyje yra pavaizduota dauguma programinės įrangos kūrimo proceso etapų:

1. Reikalavimų išgavimas, analizė ir specifikavimas;
2. Programinės įrangos projektavimas;
3. Modulių programavimas;
4. Modulių atitikimo reikalavimų specifikacijoms tikrinimas;
5. Modulių testavimas;
6. Modulių sintezė;
7. Galutinio produkto (sistemos) testavimas;
8. Produkto paruošimas naudojimui ir atidavimas klientui.

### 2.3.7 Defektų, savybių ir reikalavimų valdymo sistemos vartotojų reikalavimai

Sistema naudosis keturi veiklos dalyviai:

1. Projektų vadovai;
2. Projektuotojai;
3. Programuotojai;
4. Testuotojai.

Kiekvienam vartotojo tipui turi būti sukurti patogūs įrankiai jų veiklos pagrindiniams procesams valdyti.

<b>Vartotojas</b>	<b>Funkcinis reikalavimas</b>
<i>Projektų vadovas</i>	<ol style="list-style-type: none"> <li>1. Turi būti galimybė įvesti naujus reikalavimus ir savybes bei priskirti jas konkrečiam programuotojui.</li> <li>2. Turi būti galimybė priskirti projekto modulį (kaip reikalavimą) testavimui konkrečiam testuotojui.</li> <li>3. Projektų vadovas turi matyti visus šiuo metu vykdomus projektus ir informaciją apie juos.</li> </ol>
<i>Projektuotojas</i>	<ol style="list-style-type: none"> <li>1. Turi turėti įrankius vartotojų reikalavimų specifikacijai sudaryti ir jai sekti. Taip pat įvertinti programuotojo atliktų darbų atitikimą sudarytoms reikalavimų specifikacijoms.</li> <li>2. Projektavimas atliekamas su kitu šiai sistemai nepriklausančiu CASE įrankiu, todėl projektavimo įrankių šioje sistemoje pateikti nereikia.</li> </ol>
<i>Programuotojas</i>	<ol style="list-style-type: none"> <li>1. Turi būti sudaryta galimybė peržiūrėti jam ir kitiems programuotojams priskirtus reikalavimus ir defektus.</li> <li>2. Peržiūrėti vartotojų reikalavimų specifikacijas ir jų atitikimo įvertinimus.</li> <li>3. Turėti galimybę rašyti defektų ir savybių taisymo dokumentaciją, keisti defektų ir savybių būsenas.</li> </ol>
<i>Testuotojas</i>	<ol style="list-style-type: none"> <li>1. Turi turėti galimybę raportuoti klaidas, priskirti jas programuotojams, suteikti joms prioritetą, nurodyti jų svarbą funkcionalumui, apibrėžti defekto atkūrimo veiksmus bei kitą su defektu susijusią informaciją.</li> <li>2. Turi turėti galimybę generuoti klaidų ataskaitas.</li> </ol>

**2.1 lentelė. Vartotojų keliamų funkcinių reikalavimų lentelė**

### 2.3.8 Defektų, savybių ir reikalavimų valdymo sistemos nefunkciniai reikalavimai

Kam skirtas reikalavimas	Reikalavimo tipas	Reikalavimas
Techninė įranga	Sudėtis	<p><i>Klientas:</i></p> <p>Kompiuteris turintis ne lėtesnį nei 800 MHz procesorių, ne mažiau nei 128 MB operatyviosios atminties, 800 MB laisvos vietos diske ir su įdiegta bent 10 Mbps (rekomenduojama 100 Mbps) spartos tinklo plokšte.</p> <p><i>Serveris:</i></p> <p>Kompiuteris turintis ne lėtesnį nei 1,4 GHz procesorių, ne mažiau nei 512 MB operatyviosios atminties, 40 GB kietuoju disku ir su įdiegta 10/100Mbps spartos tinklo plokšte.</p>
	Darbingumas	<p>Informacinė sistema turi būti parengta dirbti visą parą. Sistema naudotis turi galėti daug vartotojų vienu metu.</p>
Sisteminė programinė įranga	Sudėtis	<p><i>Klientas:</i></p> <ol style="list-style-type: none"> <li>1. Operacinė sistema „Windows“ su įdiegta „Microsoft .net Framework 1.0“ platforma.</li> </ol> <p><i>Serveris:</i></p> <ol style="list-style-type: none"> <li>1. Operacinė sistema „Windows“.</li> <li>2. „Microsoft SQL Server 2000“ DBVS.</li> </ol>
	Naudojimas	<p>Informacinės sistemos interfeisas turi būti paprastas naudoti, komponentai išdėstyti pagal galiojančias vartotojo interfeiso kūrimo rekomendacijas ir standartus. Vartotojo interfeiso kalba pagal nutylėjimą – anglų.</p>
Programinės įrangos tobulinimas	Išplečiamumas	<p>Programa turi būti objektiškai suprojektuota ir būti lengvai išplečiama naujomis savybėmis per trumpą laiką.</p>
Duomenys	Saugumas	<p>Naudotis informacine sistema turi galėti tik registruoti vartotojai. Duomenų kriptografija nėra reikalinga.</p>

2.2 lentelė. Vartotojų keliamų nefunkcinių reikalavimų lentelė

### 2.3.9 Programinę įrangą kuriančios organizacijos veiklos analizės išvados

Atlikus organizacijos programinę įrangą kuriančios organizacijos veiklos analizę ištirta:

- Organizacijos struktūra, nustatyti jos padaliniai, ir juose dirbančių darbuotojų funkcijos;
- Sąveikos tarp organizacijos veiklos vienetų (padalinių) ir informacijos srautai tarp jų;
- Organizacijos tikslai, potiksliai ir sąlygos reikalingos tikslams įgyvendinti;
- Pagrindiniai veiklos objektai, nustatyti jų tarpusavio ryšiai;
- Kiekvieno įmonės darbuotojo veikla ir pagal tai sudarytas organizacijos panaudojimo atvejų modelis;
- Pagrindiniai organizacijoje vykstantys procesai, pagal kuriuos sudarytas detalus programinės įrangos kūrimo proceso modelis;
- Būsimos informacinės sistemos vartotojų poreikiai, jų keliami funkciniai ir nefunkciniai reikalavimai.

## 2.4 Pasaulinio lygio analogiškų sistemų lyginamoji analizė

Kuriant defektų, savybių ir reikalavimų valdymo sistemą yra tikslinga apžvelgti pasaulinio lygio sprendimus ir jau sukurtą programinę įrangą, kad galima būtų įvertinti jų privalumus, trūkumus, pasiūlyti naujų problemos sprendimo metodų ir papildomų savybių.

Kadangi, kuriamoji sistema veiks Windows operacinėje sistemoje, apžvelgsime tik šioje sistemoje veikiančias programas ir nenagrinėsime internetinių sprendimų.

### 2.4.1 Vertinimo kriterijai

3 lentelėje pateikti analogiškų sistemų palyginimo kriterijai, jų svarbumo svoriai ir galimos reikšmės. Kiekvieną lyginamą sistemą įvertinus pagal šiuos kriterijus bus apskaičiuojamas bendras sistemos tinkamumo koeficientas.

Nr.	Kriterijus ir jo paaiškinimas	Svoris; S [0-10]	Kriterijaus reikšmės; x [1-3]
1.	Įrankiai defektų valdymui	8	Yra (3) / Nėra (1)
2.	Įrankiai savybių valdymui	8	Yra (3) / Nėra (1)

## 2.3 lentelės tęsinys

3.	Įrankiai reikalavimų valdymui	8	Yra (3) / Nėra (1)
4.	Naudojimo paprastumas (intuityvumas)	6	Geras (3) / Vidutinis (2) / Blogas (1)
5.	Grafinio vartotojo interfeiso išvaizda ir taisyklingumas	7	Gražus (3) / Vidutinis (2) / Prastas (1)
6.	Integravimo su kitomis sistemomis (MS Word ir pan.) galimybė	3	Yra (3) / Nėra (1)
7.	Praplėtimo / pritaikymo ( <i>angl. customization</i> ) priemonės	3	Lanksčios (3) / Vidurinės (2) / Nėra (1)
8.	Veikimo sparta	9	Greita (3) / Vidutinė (2) / Lėta (1)
9.	Kaina	10	Didelė (1) / Vidutinė (2) / Maža (3)

**2.3 lentelė. Analogiškų sistemų palyginimo vertinimo kriterijų lentelė**

Sistemos tinkamumo koeficientą apskaičiuosime kiekvieno kriterijaus įvertinimo reikšmę padauginę iš kriterijaus svorio. Kriterijaus reikšmės įvertinimo reikšmė yra intervale [1; 3]. Kriterijams turintiems tik dvi reikšmes priskiriama 1 arba 3.

Bendra sistemos tinkamumo koeficiento formulė:

$$T = \frac{\sum_{i=1}^n S_i x_i}{n}$$

**2.1 formulė. Sistemos tinkamumo koeficiento apskaičiavimo formulė**

## 2.4.2 Defektų valdymo sistema „PRTracker“

Paveiksle (2.7 pav.) pateiktas pagrindinis sistemos „PRTracker“ langas su išvesta problemų (klaidų) informacija.

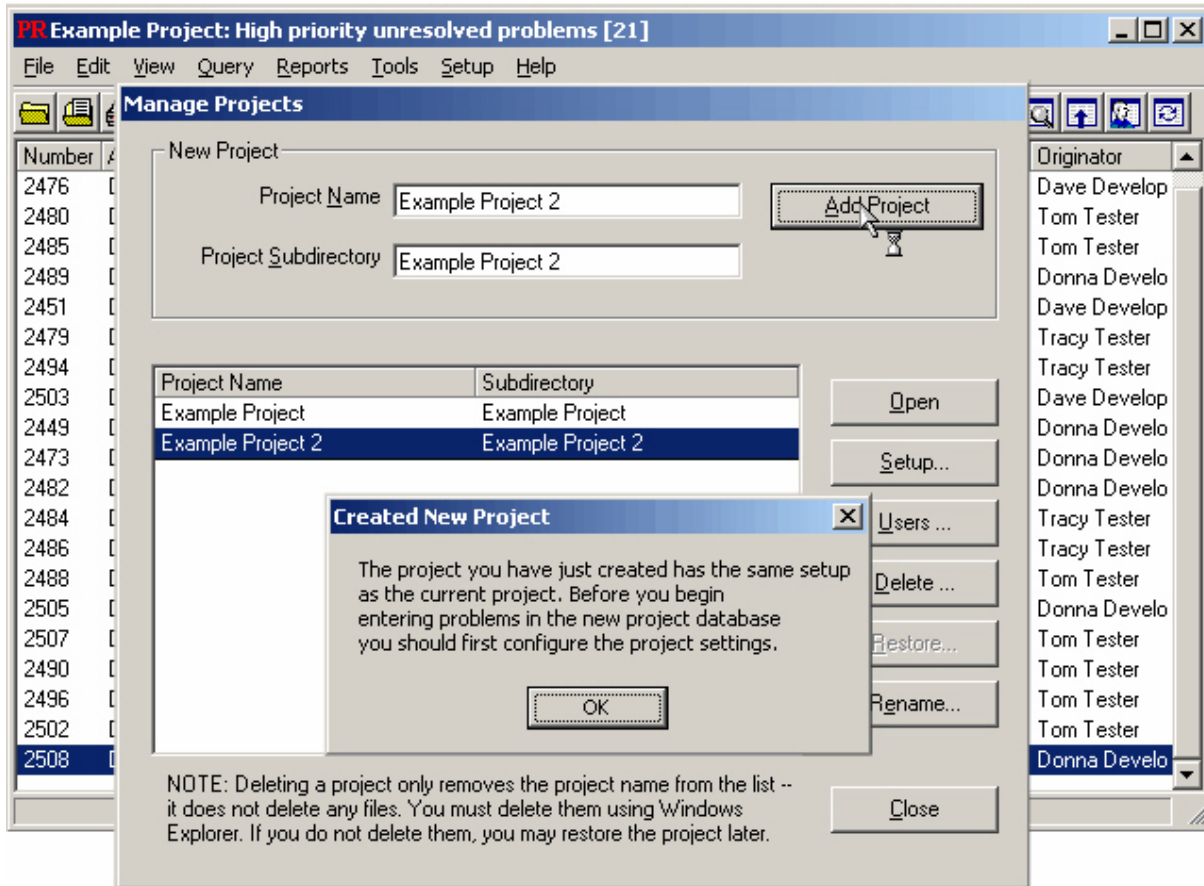
Number	Status	Priority	Title	Assigned To
1981	Investigated	2	Make webpage that performs counter syntax checking	Donna Developer
1982	Assigned	1	Evaluate counter performance using MFC40.DLL	Dave Developer
1983	Closed	1	Mixed case URLs cause problem when hosted on UNIX	
1984	New	1	Add option to put count on every frame	Dave Developer
1985	New	3	Add option not to display the counter	Donna Developer
1986	Assigned	3	Disabled account should show last page accessing counter	Dave Developer
1987	Resolved	3	Delete a range of counter lines in counter file	Tracy Tester
1988	Verified	2	Create showcounts.exe to show counter file properties	Donna Developer
1989	Closed	2	Make disable function to be invoke from button	
1990	Closed	2	Auditing webpages with counter	
1991	Postponed	3	syntaxchecker.exe doesn't work for page w/ cgi.exe	
1992	Closed	2	syntaxchecker.exe couldn't fetch the whole html file	
1993	Assigned	4	Audit.exe hangs when sending email to non-existent address	Donna Developer
1994	Resolved	1	All counts are wiped out when maxpages is exceeded	Tom Tester
1995	Closed	1	suffix "port:80" should be cut from http_referer	
1996	Closed	1	Emailcount doesn't detect error condition	
1997	Failed	4	Custom colors not used consistently	Donna Developer
1998	Assigned	3	Counter image is not showing on some easycounter pages	Dave Developer
1999	Postponed	4	The reset date isn't always logged correctly	
2000	Failed	4	Design recommendation that will improve counter performance	Donna Developer
2001	Assigned	4	both customchecker.exe and testset can't fetch ex.html fi	Dave Developer

2.7 pav. Paketo „PRTracker“ pagrindinis langas

Šis programinis paketas yra defektų valdymo sistema, naudojama bendravimui tarp testuotojų ir programuotojų. Sistemos funkcionalumas yra apribotas klaidų (defektų) informacijos valdymu, ataskaitų generavimu, paieškos ir filtravimo įrankiais.

Vienas iš esminių informacinės sistemos trūkumų yra tai, kad vienu metu galima dirbti tik su vienu projektu. Tai reiškia, kad norint pažiūrėti kito projekto informaciją, reikės uždaryti esamą ir atsidaryti kitą. Tai ypatingai nepatogu darbuotojams dirbantiems su keliais projektais. Informacija pateikiama labai „lėkštai“ ir nepatraukliai, ypač kai informacijos kiekis yra didelis.

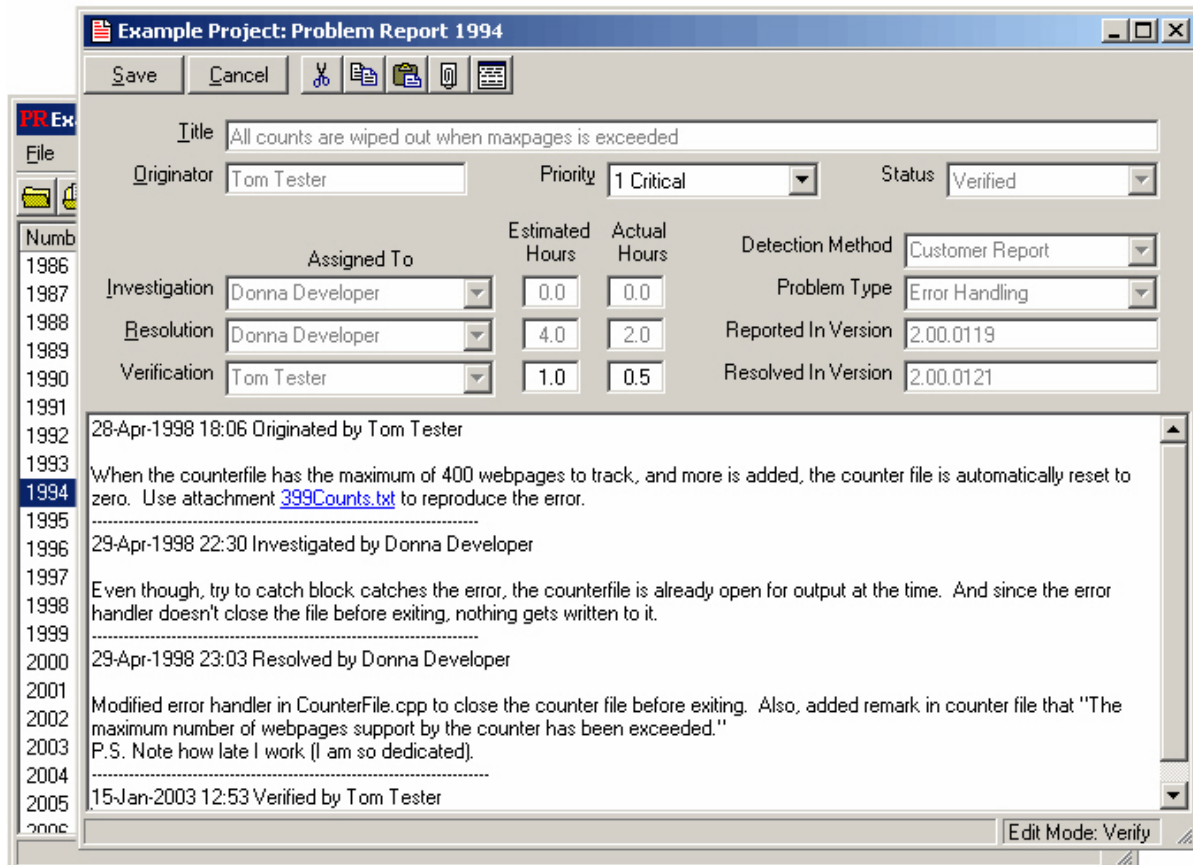
2.8 paveiksle pateiktas naujo projekto sukūrimo langas.



2.8 pav. Naujo projekto sukūrimas

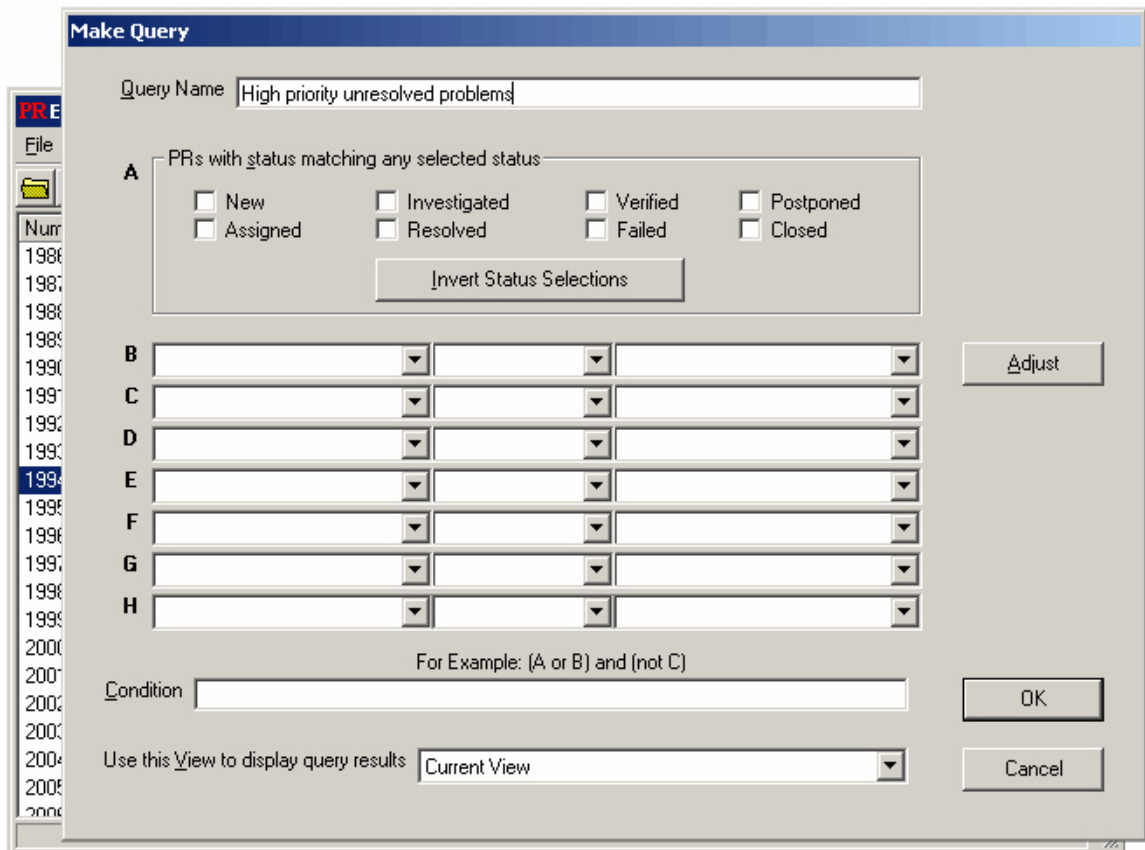
Kaip ir daugelyje kitų tokio tipo sistemų, defektų raportavimui ir jų taisymui valdyti, defektai turi savo identifikacinius numerius, prioritetus, svarbos žymeklius, už jo taisymą atsakingą programuotoją ir t.t.

Defekto informacijos įvedimo ir valdymo langas pavaizduotas 2.9 paveikslėlyje. Jame matomi anksčiau minėti defekto atributai, taip pat už testavimą atsakingas testuotojas, problemos tipas, versija ir numatytas problemai spręsti užtruksiantis laikas.



## 2.9 pav. Defekto istorija

2.10 pav. pavaizduotas defektų filtro užklauso sudarymo langas. Sudarant užklausą, galima nurodyti defekto statusą(-us), kurie bus įtraukti į užklauso rezultatus, taip pat iki 7 užklauso sąlygų (B-H). Filtravimo kriterijų kiekio apribojimas yra labai nelankstus sprendimas. Taip pat reikėtų pažymėti, kad užklauso lango struktūra neatitinka vartotojo sąsajos standartų. Didelė langų erdvės dalis yra išnaudojama neefektyviai, komponentai išdėstyti padrikai. Visa tai sukuria nerimto, nekonkurencingo ir skubotai sukurto produkto įvaizdį.



### 2.10 pav. Defektų filtravimo užklauso sudarymas

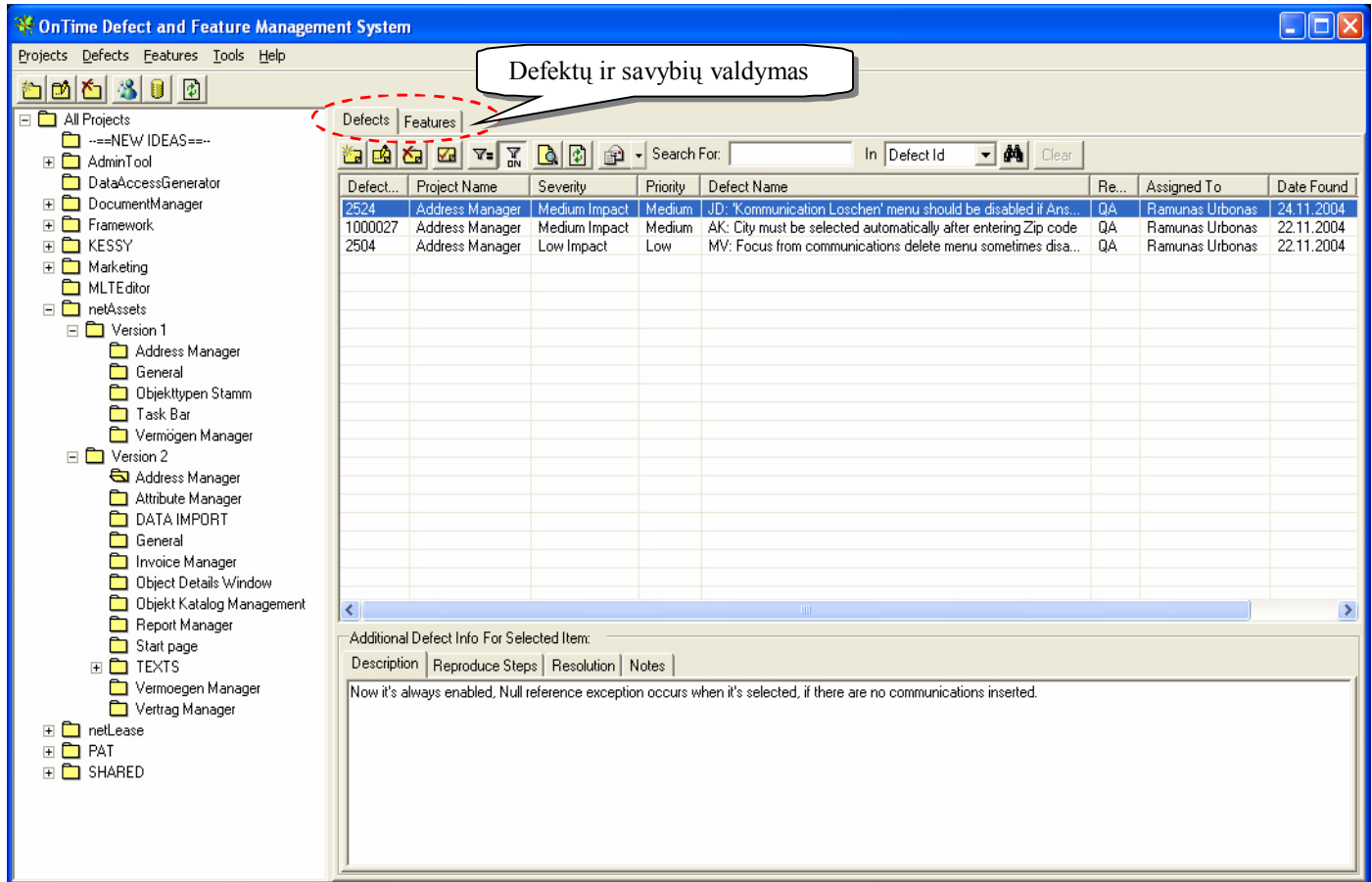
Atikus trumpą defektų valdymo sistemos „PRTracker“ analizę padarytos tokios išvados:

- Šioje sistemoje negalima struktūrizuoti projekto į modulius ar sub-projektus. Tai sukuria didelę painiavą. Programuotojams dirbantiems su konkrečiais projekto moduliais tenka susikurti specialius filtrus, pagal kuriuos jie galėtų atmesti kitiems moduliams priskirtus defektus.
- Sistema apribota tik defektų valdymu, nėra naujų savybių ar reikalavimų specifikavimo įrankių.
- Vartotojo interfeisas sudarytas gana paviršutiniškai, nesilaikant komponentų kraštų išlygiavimo, minimalaus ekrano ploto panaudojimo ir kitų vartotojo sąsajos projektavimo standartų (2.9 pav.).
- Informacija apie defektą yra pakankamai išsami.



### 2.4.3 Defektų ir savybių valdymo sistema „OnTime“

2.11 paveiksle pateiktas paketo „OnTime“ pagrindinis langas. Jame yra žymiau daug daugiau informacijos, nei prieš tai nagrinėto paketo („PRTracker“) lange.



2.11 pav. Paketo „OnTime“ pagrindinis langas

Programos „OnTime“ lango kairiojoje pusėje yra projektų medis, kuris žymiai palengvina navigaciją tarp projektų. Labai paprastas projektų sukūrimas, pašalinimas, informacijos redagavimas. Pagrindinėje lango dalyje, panašiai kaip ir pakete „PRTracker“ pateikta lentelė su defektų informacijos sąrašu. Reikia pažymėti, kad ši sistema gali būti panaudota taip pat ir savybių („Features“) valdymui. Keičiant korteles „Defects“ ir „Features“, lango viršuje, galima persijunginėti tarp šių dviejų režimų.

Šiame pakete taip pat yra vartotojų valdymas, įrankiai defektų ir savybių filtravimo užklausoms sudaryti, ir t.t.

Šiuose paveikslėliuose (2.12 pav. ir 2.13 pav.) pateikti naujo defekto sukūrimo ir redagavimo langai. Juose informacija yra gerai struktūrizuota, tačiau ne tokia detali kaip „PRTracker“ sistemoje. Langu apačioje suvedama defekto atkūrimo, jo sprendimo ir kita papildoma informacija.

2.12 pav. Defekto sukūrimo forma

2.13 pav. Defekto informacijos redagavimo forma

**Add a New Feature**

Primary Feature Info

Name:

Description:

Project:

Start Date:  Due Date:

Work Estimate:  Priority:

Additional Feature Info

Assigned To:

Status:

Build Number:

Notes:

Percent Complete:  100

**2.14 pav. Naujos savybės sukūrimo forma**

**Edit Feature**

Primary Feature Info

Name:

Description:

Project:

Start Date:  Due Date:

Work Estimate:  Priority:

Additional Feature Info

Assigned To:  Status:

Build Number:

Notes:

Percent Complete:  100

**2.15 pav. Savybės informacijos redagavimo forma**

Paveikslėliuose 2.14 ir 2.15 pavaizduoti naujos savybės sukūrimo langai. Savybės informacija šiek tiek skiriasi nuo defekto informacijos. Čia galima nustatyti savybės realizacijos pradžios datą bei numatomą jos realizavimo užbaigimo datą. Programuotojas realizuojantis jam priskirtą savybę, gali reguliariai atnaujinti informaciją apie savybės realizavimo progresą ir pateikti papildomą informaciją.

2.16 pav. pavaizduotas projekto sukūrimo dialogas. Naujai sukurti projektai dedami į projektų medį. Informacija apie projektą yra ganėtinai kukli, tačiau daugeliu atvejų pakankama.

**2.16 pav. Projekto sukūrimo dialogas**

Šioje, kaip ir daugelyje kitų tokio tipo sistemų, visi vartotojai yra registruoti. Vartotojų valdymo langas yra pateiktas 2.17 paveiksle.

FirstName	LastName	Email	LoginId	Windows Auth
Arturas	Katutis		ARK	False
Julijus	Katinas	julijusk@bss.biz	JUK	False
Jurgis	Klimavicius	jurgis.klimavicius@bss.biz	jk	False
Linas	Kricenas		LK	False
Linas	Mikutaitis		LM	False
Mindaugas	Krapauskas	mindaugask@bss.biz	MK	False
QA			Admin	False
Ramunas	Urbonas		ru	False
Rimas	Zavisius	rimasz@bss.biz	RZ	False
Rokas	Valantinas	rokasv@bss.biz	RV	False

**2.17 pav. Sistemos vartotojų valdymo langas**

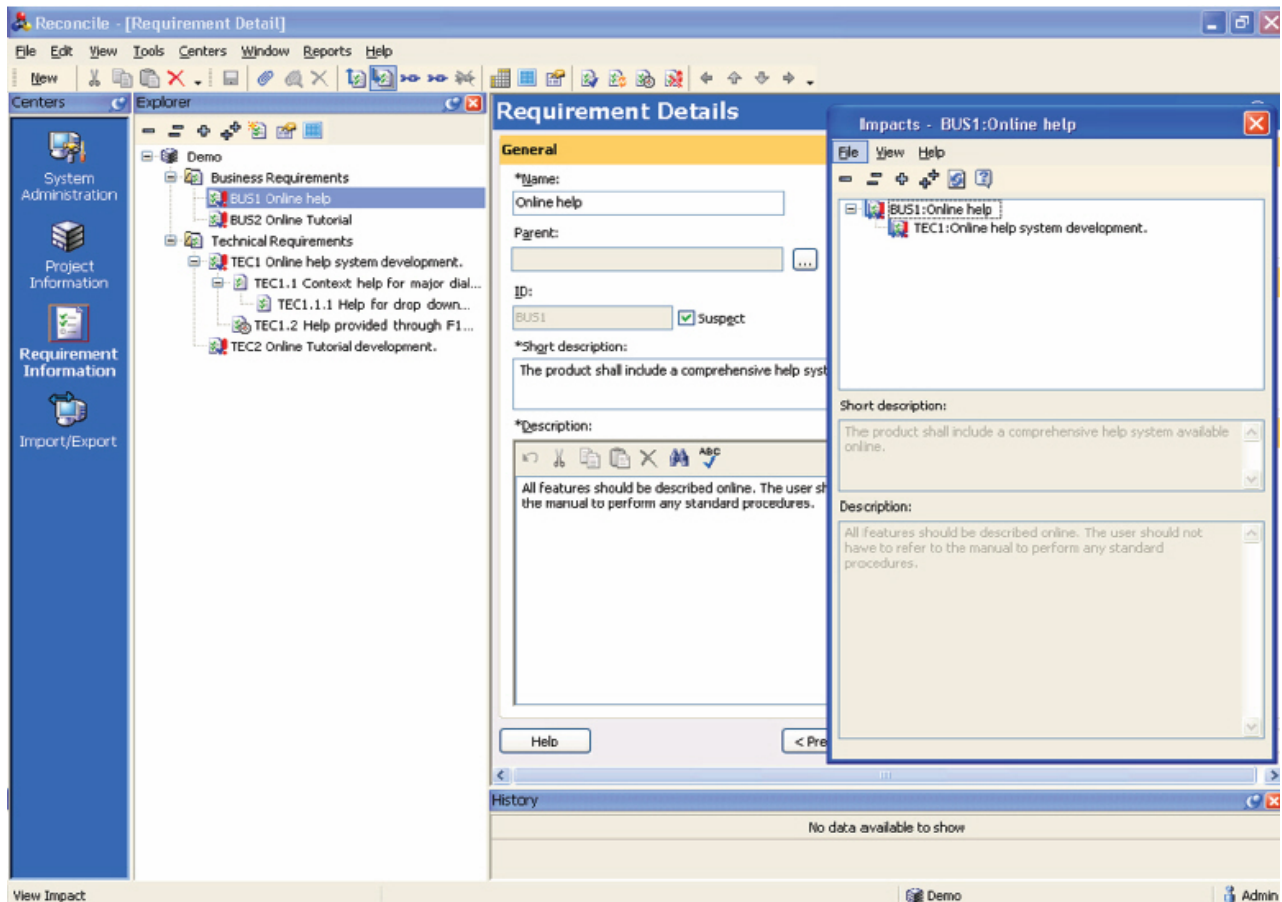
Vartotojų valdymo langas yra paprastas, tačiau nelabai patogus vartotojui. Esant dideliame sistemos vartotojų skaičiui, darosi sunku surasti norimą įrašą.

Atikus defektų ir savybių valdymo sistemos „OnTime“ analizę padarytos tokios išvados:

- Ši sistema daugeliu atvejų yra pranašesnė ir patogesnė naudotis nei „PRTracker“ sistema.
- „OnTime“ sistema, skirtingai nei „PRTracker“ turi ne tik defektų bet programinės įrangos savybių valdymo sistemą.
- Programos vartotojo interfeisas suprojektuotas žymiai geriau nei anksčiau nagrinėtos sistemos.
- Projektų medis yra labai patogi vartoti grafinio interfeiso dalis.
- Sistema neturi jokio vartotojų reikalavimų specifikavimo ir sekimo įrankių.
- Naudojantis sistema pastebimos veikimo spartos problemos. Pavyzdžiui kas kartą atidarius / uždarius defektų ar savybių dialogus, jų sąrašai yra kas kartą atnaujinami. Esant lėtam ryšiui su serveriu tai gali visiškai paralyžiuoti sistemos darbą arba padaryti jį labai neefektyviu.

#### 2.4.4 Reikalavimų valdymo sistema „Reconcile“

2.18 paveikslėlyje, pavaizduotas programinio paketo „Axosoft Reconcile“ pagrindinis langas. Tai yra vienas iš didesnių paketų skirtų projektų valdymui. Viena iš esminių šio produkto funkcijų yra reikalavimų valdymas ir jų sekamumas.

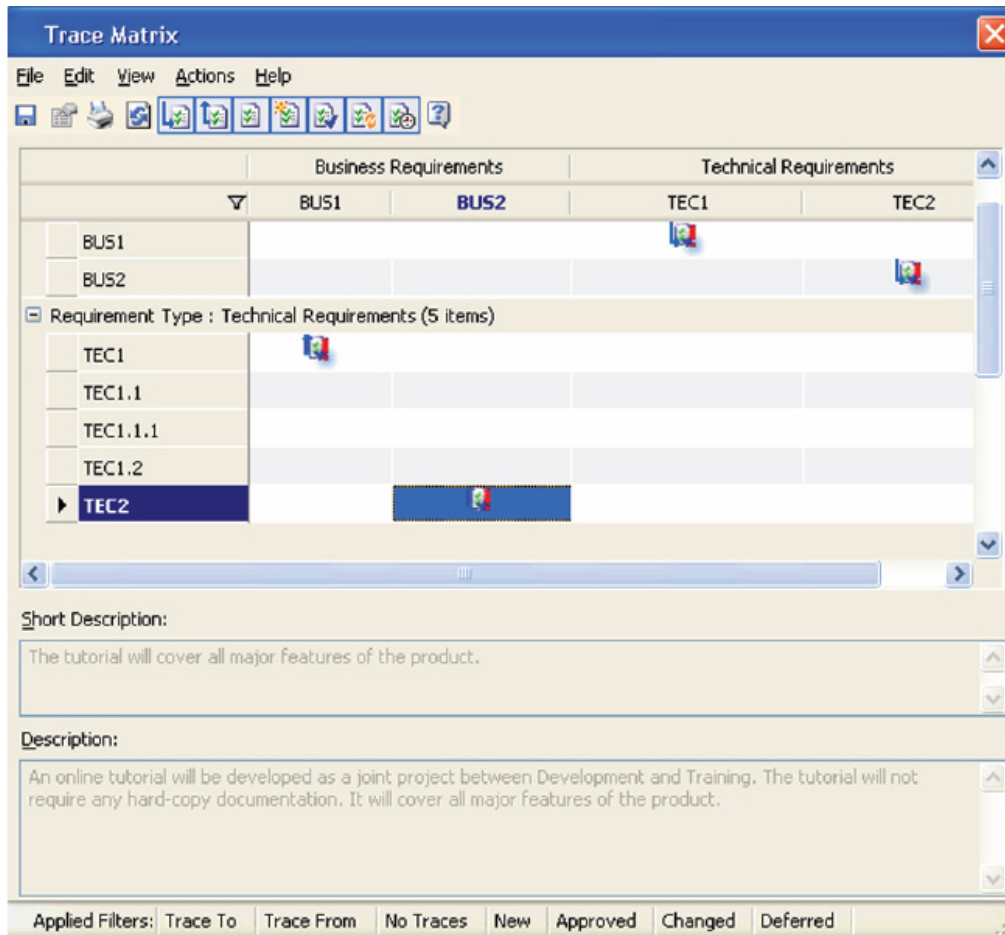


2.18 pav. „Reconcile“ reikalavimų valdymo sistemos pagrindinis langas

Naudojantis šiuo paketu galima specifikuoti trijų tipų reikalavimus:

1. Vartotojų reikalavimus;
2. Biznio veiklos reikalavimus;
3. Techninius reikalavimus;

Šiame paveiksle (2.19 pav.) pateiktas reikalavimų sekamumo matricos sudarymo langas. Naudojantis šia matrica, galima matyti kaip biznio reikalavimai susiję su techniniais ir atvirkščiai.



**2.19 pav. Reikalavimų sekimo matrica**

Atikus projekto ir reikalavimų valdymo sistemos „Reconcile“ analizę padarytos tokios išvados:

- Ši sistema skirta projektų administravimui, reikalavimų specifیکavimui ir sekamumui;
- Sistemoje galima specifikuoti kelių tipų reikalavimus ir juos susieti sekamumo matricoje;
- Vartotojo interfeisas yra taisyklingas ir patrauklus;

## 2.4.5 Reikalavimų valdymo sistema „DOORS“

DOORS – sistema skirta didelių ir labai didelių projektų reikalavimų valdymui automatizuoti: „DOORS is a sophisticated product that can manage requirements on large products“. Tai viena iš plačiausiai naudojamų reikalavimų valdymo sistemų pasaulyje.

Tarp lyginamų sistemų tai vienintelė sistema, kuri apima visas projekto gyvavimo fazes, nuo reikalavimų surinkimo iki produkto likvidavimo. Sistema skirta tik reikalavimų valdymui. Reikalavimai sistemoje traktuojami kaip atskiri objektai, tačiau sistema palaiko daug jų vaizdavimo formų. Viena iš jų yra reikalavimų specifikavimo standartus atitinkantis reikalavimų dokumentas.

2.20 paveikslėlyje pavaizduoti sistemos „DOORS“ vartotojai.

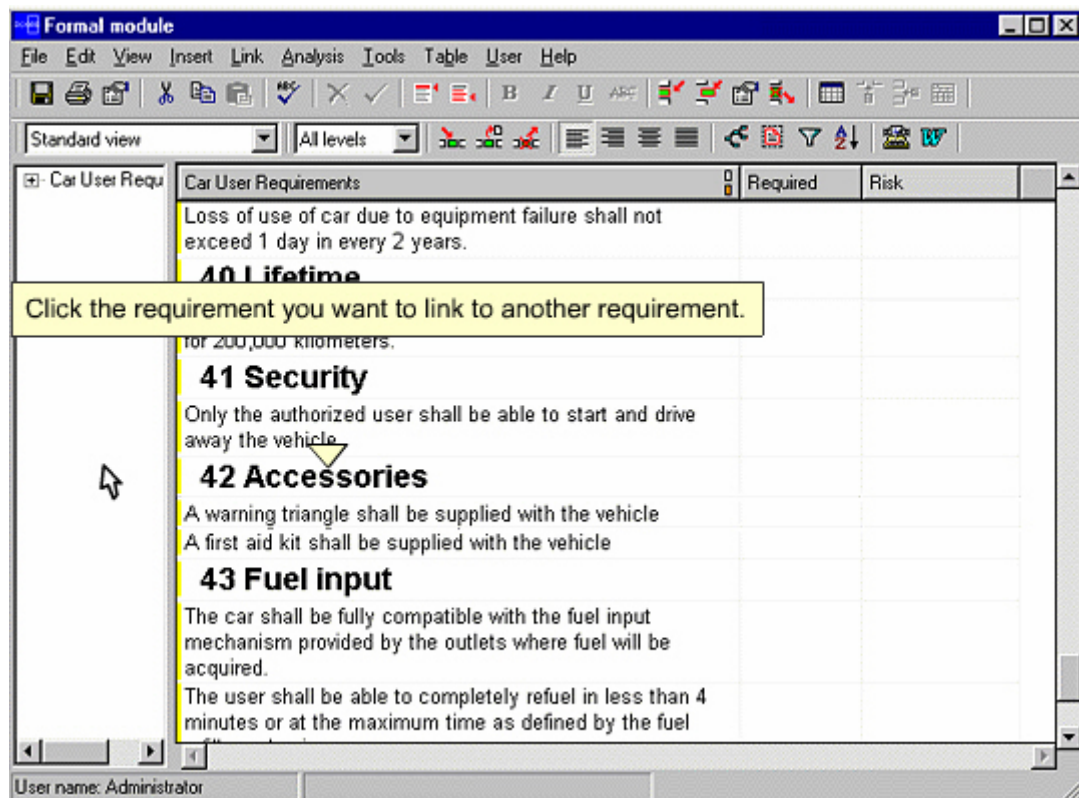


2.20 pav. Sistemos DOORS vartotojai

Sistema DOORS, skirtingai nuo daugelio analogiškų reikalavimų valdymo sistemų, suteikia galimybę būsimiems sistemos vartotojams dalyvauti sistemos kūrimo procese. Jiems leidžiama peržiūrėti reikalavimų medžiagą, rašyti atsiliepimus, pastabas bei kurti pasiūlymus (savybes).

Deja sistemos vartotojo interfeisas nėra labai intuityvus. Pradedant naudotis sistema, vartotojui prireiks ilgai klaidžioti tarp sistemos langų. Tai didžiulė sistema, automatizuojanti visą projekto reikalavimų valdymo procesą, todėl mažiems ir santykinai lengvai valdomiems projektams (1 – 2 projekto vadovai) šią sistemą naudoti yra netikslinga.

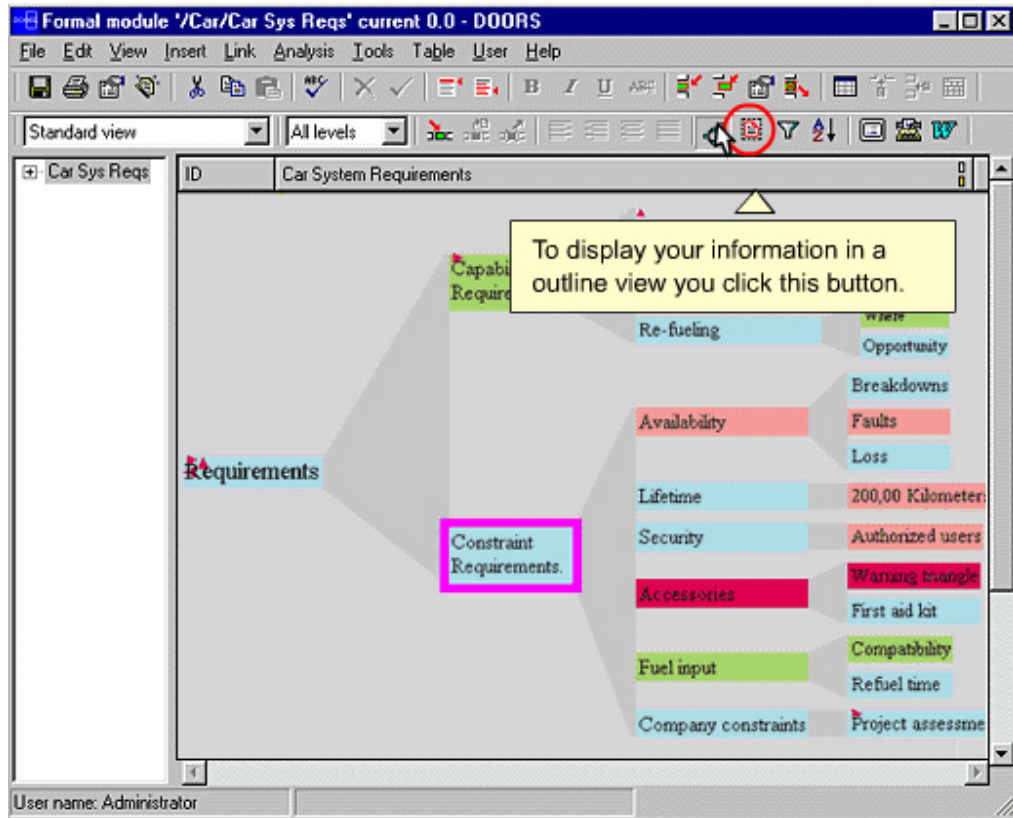
Reikalavimai sistemoje DOORS pateikiami kaip reikalavimų specifikacijos (2.21 pav.). Vartotojas gali redaguoti reikalavimų specifikaciją, tuo pačiu metu redaguodamas atskirus reikalavimus.



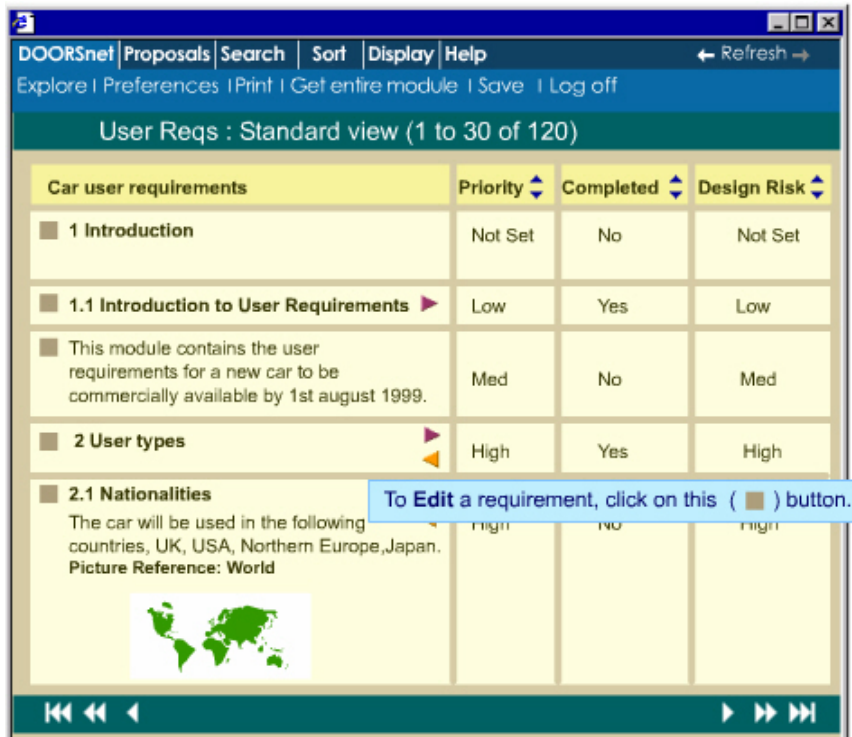
2.21 pav. Reikalavimų / reikalavimų specifikacijos redagavimas

Paveiksluose 2.22 ir 2.23 pateikti langai, kuriuose pavaizduotos reikalavimų pateikimo vartotojui alternatyvos.



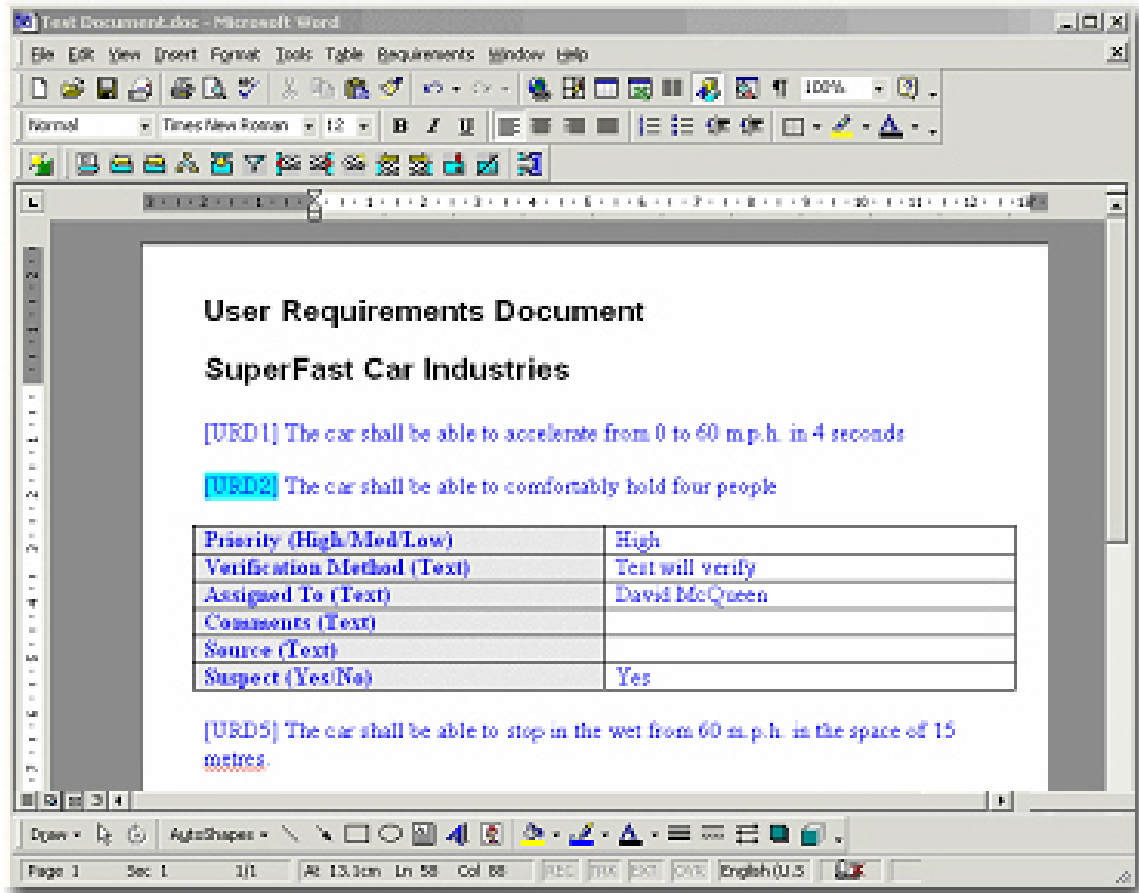


2.22 pav. Reikalavimų struktūros peržiūros langas



2.23 pav. Reikalavimų sąrašo peržiūrėjimas naršyklės lange

Skirtingai nei anksčiau aptartose analogiškose sistemose, sistema DOORS integruojasi į Microsoft Office darbinę aplinką. Integruoti įrankiai leidžia vartotojui betarpiškai redaguoti reikalavimus tekstinio redaktoriaus pagalba (2.24 pav.). Ši sistema taip pat integruojasi į MS Project, Rational Rose ir kitas plačiai naudojamas sistemas.



2.24 pav. Reikalavimų valdymo sistema DOORS integruota į MS Word teksto redaktorių

Atikus reikalavimų valdymo sistemos „DOORS“ analizę padarytos tokios išvados:

- Ši sistema skirta didelių ir labai didelių projektų valdymui;
- Sistema turi daug skirtingų reikalavimų valdymo ir modeliavimo įrankių;
- Sistema integruojasi į daugelį gerai žinomų paketų, tokių kaip MS Office, MS Project, Rational Rose ir kt. Taip pat turi savo vidinę, programavimo kalbą (kurios sintaksė panaši į C);
- Vartotojo interfeisas yra taisyklingas, atitinkantis standartus, tačiau neintuityvus ir gana painus pradedančiajam vartotojui;
- Tai brangi sistema, todėl netinka mažoms ir vidutinėms programinei įrangą kuriančioms organizacijoms.

## 2.4.6 Lyginamosios analizės rezultatai

Pagal 2.3 lentelėje pateiktus lyginamosios analizės kriterijus, sudaryta sistemų palyginimo lentelė:

Nr.	Kriterijus ir jo paaiškinimas	S <sub>i</sub>	Lyginamos sistemos			
			PRTracker	Ontime	Reconcile	DOORS
1.	Įrankiai defektų valdymui	8	Yra (3)	Yra (3)	Nėra (1)	Nėra (1)
2.	Įrankiai savybių valdymui	8	Nėra (1)	Yra (3)	Nėra (1)	Nėra (1)
3.	Įrankiai reikalavimų valdymui	8	Nėra (1)	Nėra (1)	Yra (3)	Yra (3)
4.	Naudojimo paprastumas (intuityvumas)	6	Vidutinis (2)	Geras (3)	Geras (3)	Blogas (1)
5.	Grafinio vartotojo interfeiso išvaizda ir taisyklingumas	7	Blogas (1)	Vidutinis (2)	Geras (3)	Vidutinis (2)
6.	Integravimo su kitomis sistemomis (MS Word ir pan.) galimybė	3	Nėra (1)	Nėra (1)	Yra (3)	Yra (3)
7.	Praplėtimo / pritaikymo ( <i>angl. customization</i> ) priemonės	3	Nėra (1)	Nėra (1)	Nėra (1)	Yra (3)
8.	Veikimo sparta	7	Prasta (1)	Prasta (1)	Vidutinė (2)	Gera (3)
9.	Kaina	10	Maža (3)	Maža (3)	Didelė (1)	Didelė (1)

**2.4 lentelė. Analogiškų sistemų palyginimo lentelė**

Atsižvelgdami į kriterijų palyginimo koeficientų svorius (žr. 2.3 lentelė), naudodamiesi 2.4.1 skyriuje pateikta formule (2.1 formulė), apskaičiuojame bendrą sistemos tinkamumą. Analogiškų sistemų palyginimo rezultatai pateikti 2.5 lentelėje.

$$T = \frac{\sum_{i=1}^n S_i x_i}{n}$$

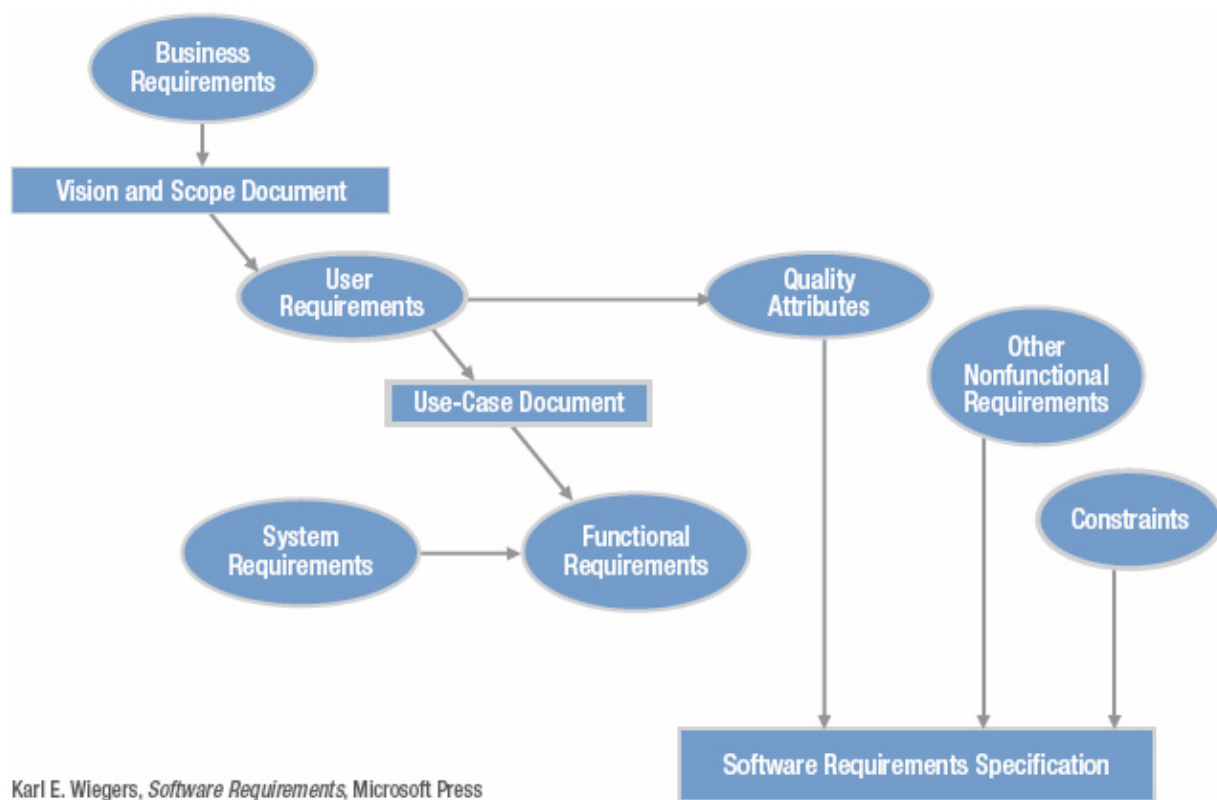
Lyginamoji sistema	Bendras sistemos tinkamumas; T
PRTracker	11,3
Ontime	13
Reconcile	12,8
DOORS	12,1

**2.5 lentelė. Analogiškų sistemų palyginimo rezultatai**

Pagal analogiškų sistemų palyginimo rezultatus gauta, kad sistemų Ontime ir Reconcile sistemos tinkamumo rodikliai yra didžiausi. Kadangi Ontime yra defektų ir savybių valdymo sistema, o Reconcile – reikalavimų, būtų tikslinga geriausias šių sistemų savybes panaudoti kuriant bendrą defektų, savybių ir reikalavimų valdymo sistemą. Sistema DOORS turi daugybę funkcijų kurios nedidelėms programine įrangą kuriančioms įmonėms yra nenaudingos.

## 2.5 Literatūros šaltiniuose pateikti sprendimai problemai spręsti

Literatūroje pateikiamas toks reikalavimų specifikavimo modelis (2.25 pav.):



2.25 pav. Reikalavimų specifikavimo modelis

Biznio reikalavimai sudaromi remiantis dalykinės srities informacija. Šie yra patikslinami vartotojo reikalavimais, pagal kuriuos sudaromas *panaudos atvejų* modelis iš kurio galima išvelgti funkcinis ir sisteminius reikalavimus. Pagal vartotojų reikalavimus, taip pat specifikuojami kokybės atributai, nustatomi apribojimai, nefunkciniai reikalavimai iš ko sukuriama galutinė *programos reikalavimų specifikacija*.

## 2.6 Projektavimo metodų ir priemonių parinkimas

Projektavimo įrankių pasirinkimas gali stipriai įtakoti projektavimo darbų laiką ir rezultatus. Kuriant programinės įrangos realizavimo projektą geriausia pasirinkti anksčiau naudotus ir laiko patikrintus įrankius.

Informacinės sistemos projektavimui pasirinktos UML projektavimo notacijos. UML projektavimo naudojimas labai palengvina programavimo darbus, kadangi dauguma UML palaikančių įrankių turi automatinio kodo generavimo funkcijas, be to UML notacijos palengvina objektiškai orientuotą analizę (*angl. object oriented analysis*), lengviau identifikuojami objektai ir jų operacijos. Struktūrinis projektavimo metodas yra pasenęs ir labiau tinkamas funkcinio programavimo užduotims aprašyti. Jo taikymas būtų naudingas atliekant liktinių (*angl. legacy*) sistemų plėtojimą.

Sistemai projektuoti pasirinkti gerai žinomi UML palaikantys paketai: „Rational Rose 2002“ ir „Microsoft Visio 2003“. Abu šie paketai yra gerai žinomi ir plačiai naudojami pasaulyje.

## 2.8 Analizės išvados

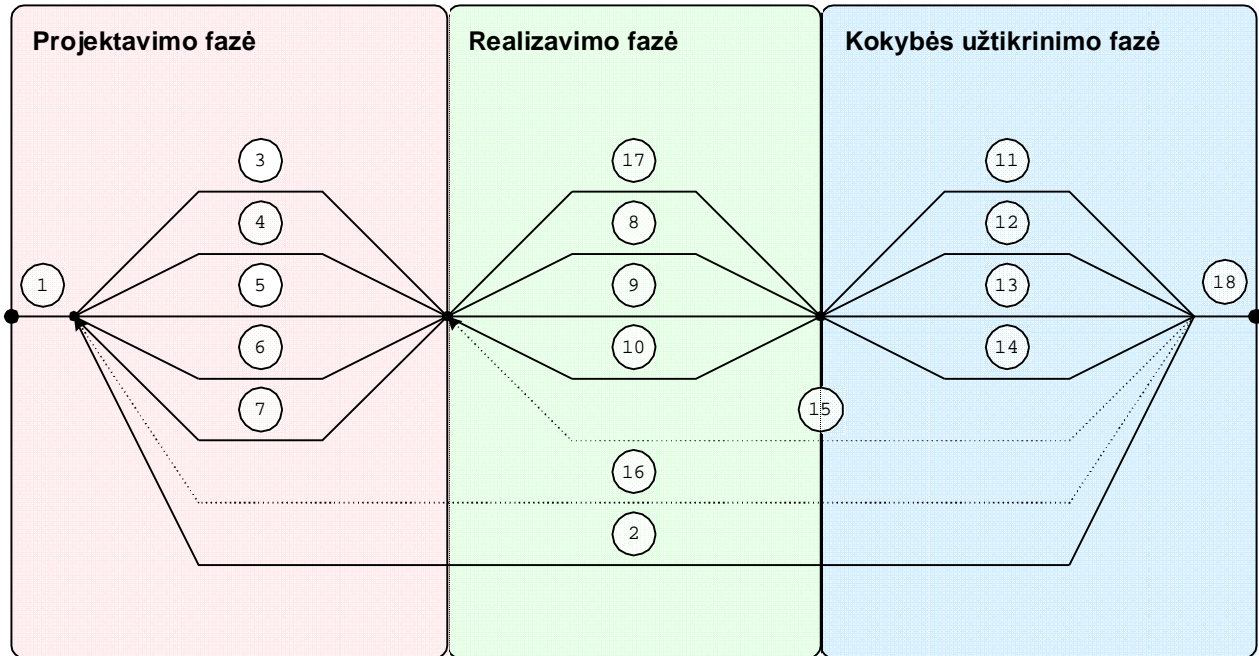
Šiame analizės skyriuje:

- Apibrėžta tyrimo sritis, objektas, suformuluota šio darbo problema;
- Atlikta programinę įrangą kuriančios organizacijos veiklos analizė;
- Ištirti organizacijoje vykstantys programinės įrangos kūrimo procesai bei sudarytas tų procesų kompiuterizavimo planas;
- Atlikta vartotojų bei jų funkcinių ir nefunkcinių reikalavimų analizė;
- Išnagrinėtos panašios pasaulinio lygio informacinės sistemos, ištirti jų trūkumai, privalumai, apskaičiuoti naudingumo koeficientai, padarytos šių sistemų palyginimo išvados, kuriomis remiantis bus kuriama tiriamoji informacinė sistema;
- *Atliktas šia tema pasaulinės literatūros ir esamų problemos sprendimo metodų tyrimas.*

Defektų, savybių ir vartotojų specifkavimo valdymo sistemos įdiegimas organizacijoje teigiamai įtakotų jos darbo spartą, kokybę bei palengvintų jos darbuotojų darbą.

### 3. DEFEKTŲ, SAVYBIŲ IR REIKALAVIMŲ VALDYMO METODAS

Šiame skyriuje pateikiamas defektų savybių ir reikalavimų valdymo metodas. Paveiksle 3.1 pareikiamas metodo etapų tinklinė diagrama. Žemiau pateikiamas pavaizduotų etapų sąrašas ir etapų analizių lentelės.



3.1 pav. Defektų, savybių ir reikalavimų valdymo metodo etapų tinklinė diagrama

Defektų, savybių ir reikalavimų valdymo metodo etapai:

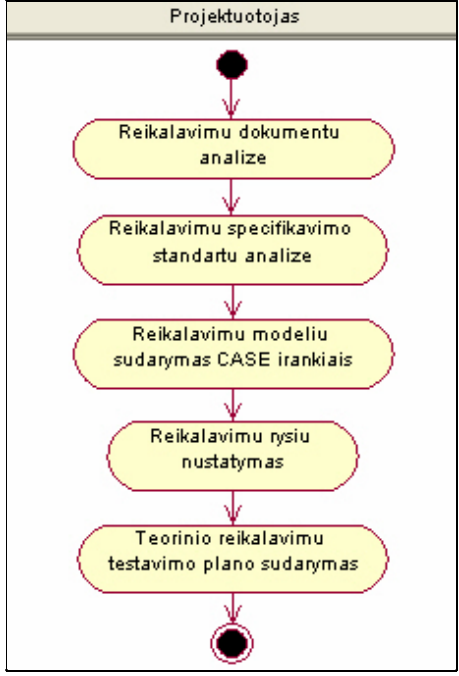
1. Projekto sukūrimas
2. Projekto darbų stebėjimas ir valdymas
3. Projekto reikalavimų analizė
4. Reikalavimų modeliavimas / formalizavimas
5. Reikalavimų skaidymas į savybes
6. Reikalavimų priskyrimas programuotojams
7. Savybių priskyrimas programuotojams
8. Reikalavimų realizavimas
9. Savybių realizavimas
10. Rezultatų dokumentavimas
11. Testavimas, defektų priskyrimas programuotojams
12. Ištaisytų defektų peržiūra ir įvertinimas
13. Realizuotų reikalavimų ir savybių peržiūra ir įvertinimas
14. Testavimo ataskaitų ruošimas
15. Defektų, savybių ar reikalavimų pakartotinis priskyrimas / realizacijų atmetimas (pakartotiniam realizavimui arba taisymui)
16. Naujų / Pasikeitusių reikalavimų įvedimas / atnaujinimas
17. Defektų, savybių ar reikalavimų pakartotinė realizacija / taisymas
18. Projekto išleidimas

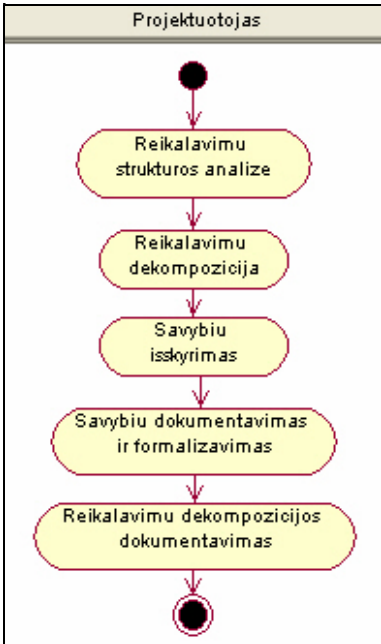
ETAPAS 1: Projekto sukūrimas		
<b>PAAIŠKINIMAS</b>	Šiame etape surenkami pagrindiniai abstraktūs vartotojų reikalavimai, paruošiama pradinė projekto dokumentacija bei skiriami resursai analizės ir projektavimo darbams atlikti.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
<ul style="list-style-type: none"> <li>- Projekto dalykinės srities informacija;</li> <li>- Abstraktūs, neformalūs vartotojo reikalavimai</li> <li>- Projekto darbų sąrašas</li> <li>- Resursų paskirstymo grafikas</li> </ul>	<ul style="list-style-type: none"> <li>- Paskirtas projektų vadovas</li> <li>- Paskirti darbai projektuotojui</li> <li>- Užregistruotas projektas informacinėje sistemoje</li> <li>- Sudaryti abstrakčių reikalavimų dokumentai</li> </ul>	Pradinės projekto dokumentacijos paruošimas ir darbų paskirstymas, priklausomai nuo projekto dydžio, turėtų užimti 1-2 darbo dienas (jeigu pradiniai reikalavimai yra pateikti).
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>IRANKIAI</b>
<ul style="list-style-type: none"> <li>- Projektų vadovas</li> </ul>	-	<ul style="list-style-type: none"> <li>- Teksto redaktorius projekto dokumentams kurti.</li> </ul>
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center; background-color: #f2f2f2; margin-bottom: 5px;">Projektu vadovas</p> <pre> graph TD     Start(( )) --&gt; A[Projekto iniciavimas]     A --&gt; B[Abstrakčiu reikalavimu dokumento sudarymas]     B --&gt; C[Projekto informacijos talpinimas]     C --&gt; D[Projektuotojo paskyrimas]     D --&gt; End((( )))           </pre> </div>	<p>„<i>Projekto inicijavimas</i>“ – Projektų vadovas suderina projekto finansavimo klausimus su organizacijos administracija, gauna kitus projekto darbų pradžiai reikalingus dokumentus.</p> <p>„<i>Abstrakčių reikalavimų dokumento sudarymas</i>“ – Projektų vadovas surenka projekto analitikų paruoštą dokumentaciją, ją susistemina, jeigu reikia apjungia į vieną ar kelis dokumentus.</p> <p>„<i>Projekto informacijos talpinimas</i>“ – Projektų vadovas patalpina surinktą informaciją į darbuotojams prieinamą vietą (pvz.: portalą, viešą diską arba jeigu naudojama specializuota informacinė sistema, prisega dokumentus prie projekto).</p> <p>„<i>Projektuotojo paskyrimas</i>“ – Projektų vadovas paskiria projektuotoją, tolesnei projekto analizei bei projektavimo ir modeliavimo darbams atlikti.</p>	

<b>ETAPAS 2: Projekto darbų stebėjimas ir valdymas</b>		
<b>PAAIŠKINIMAS</b>	Šis etapas tęsiasi visą projekto kūrimo laiką. Šio žingsnio darbus sudaro susitikimų įvairiais klausimais organizavimas, iškilusių problemų sprendimas, ataskaitų analizė ir pan.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Projekto dokumentacija - Sistemos kūrimo planas - Darbo laiko ataskaitos - Testavimo ataskaitos	- Darbų būsenų ataskaitos - Susitikimų planai - Pakeisti sistemos kūrimo planai - Kokybės užtikrinimo rezultatai	Darbų stebėjimas ir valdymas trunka nuo projekto inicijavimo iki jo išleidimo ( <i>angl. release</i> ).
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Projektų vadovas - Projektuotojas - Programuotojai - Būsimi sistemos vartotojai - Įmonės administracija - Kokybės užtikrinimo padalinys	- Darbų būsenos stebėjimas - Resursų valdymas - Projektų valdymas - Kokybės užtikrinimas	- Projektų valdymo programinė įranga
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
1. Sistemos kūrimo planų peržiūra 2. Testavimo rezultatų peržiūra 3. Darbų būsenos ir darbuotojų našumo įvertinimai 4. Susitikimų organizavimas ir dalyvavimas juose 5. Iškilusių klausimų sprendimas, bendravimas su būsima sistema vartotojais 6. Reikiamų įrankių tiekimas darbuotojams	<p>„Sistemos kūrimo planų peržiūra“ – projektuotojų sudarytų reikalavimų realizavimo planų peržiūra ir aptarimas specialiuose susitikimuose.</p> <p>„Testavimo rezultatų peržiūra“ – kokybės užtikrinimo padalinio pateiktų ataskaitų analizė, klaidų kiekio ir ištaisymo laiko stebėjimas.</p> <p>„Darbų būsenos ir darbuotojų našumo įvertinimai“ – atliktų darbų ataskaitų analizė bei darbų atliktų per tam tikrą laiką stebėjimas. Sistemos kūrimo darbotvarkės koregavimas.</p> <p>„Susitikimų organizavimas ir dalyvavimas juose“ – susijusių asmenų įtraukimas į susitikimus iškilusioms problemoms spręsti. Susitikimuose aptartų temų ir priimtų sprendimų dokumentavimas.</p> <p>„Iškilusių klausimų sprendimas, bendravimas su būsima sistema vartotojais“ – kilus neaiškumams realizuojant sistemą ar pastebėjus dviprasmiškus, nesuderinamus reikalavimus kreipimasis į užsakovus, ar kitus suinteresuotus asmenis.</p> <p>„Reikiamų įrankių tiekimas darbuotojams“ – reikiamų projektavimo, programavimo, testavimo įrankių tiekimas darbuotojams.</p>	



ETAPAS 3: Projekto reikalavimų analizė		
<b>PAAIŠKINIMAS</b>	Šiame etape atliekama vartotojo reikalavimų analizė ir formuluojami nauji reikalavimai: sisteminiai, techniniai, veiklos. Šiame etape pasiekti rezultatai bus naudojami atliekant reikalavimų modeliavimą.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Projekto dokumentacija - Vartotojų reikalavimų specifikacija	- Konkretūs inžineriniai reikalavimai - Realizavimo apribojimai - Nefunkciniai sistemos reikalavimai - Kokybės reikalavimai	Projekto reikalavimų analizė trunka nuo projekto pradžios iki realizacijos darbų pradžios. Konkreti trukmė priklauso nuo projekto dydžio ir nuo reikalavimus analizuojančių darbuotojų žinių projekto dalykinėje srityje.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Projektų vadovas - Projektuotojas	- Prezencijos ir susitikimai - Panaudojimo atvejų modelio kūrimas, funkcinų ir sisteminių reikalavimų atradimui - Techninių reikalavimų išskyrimas - Sisteminių reikalavimų išskyrimas	- UML palaikantis CASE įrankis panaudojimo atvejų ( <i>angl. „Use-Case“</i> ) diagramoms kurti - Prezencijų kūrimo įrankiai
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div data-bbox="186 1018 649 1774"> <p>Projektu vadovas ir Projektuotojas</p> <pre> graph TD     Start(( )) --&gt; A[Projekto informacijos analize]     A --&gt; B[Panaudojimo atveju modelio analize]     B --&gt; C[Veiklos, sisteminiu ir techniniu reikalavimu radimas]     B --&gt; D[Nefunkcinu reikalavimu radimas]     C --&gt; E[Realizavimo apribojimu specifikavimas]     D --&gt; E     E --&gt; F[Prezentaciju kurimas]     F --&gt; End((( )))   </pre> </div>	<p>„<i>Projekto informacijos analize</i>“ – analizuojama analitikų pateikta projekto reikalavimų specifikacija, išskiriami pagrindiniai sistemos atributai, kuriama sistemos vizija.</p> <p>„<i>Panaudojimo atvejų modelio kūrimas</i>“ – analizuojamas analitikų sukurtas panaudojimo atvejų modelis, pagal šį modelį bus randami veiklos, sisteminiai, techniniai ir kiti reikalavimai.</p> <p>„<i>Veiklos, sisteminių ir techninių reikalavimų radimas</i>“ – analizuojant panaudojimo atvejus ir projekto informaciją, randami inžineriniai reikalavimai. Vėlesniuose etapuose šie reikalavimai bus sumodeliuoti, specifiukuoti, paruoštos jų realizavimo gairės.</p> <p>„<i>Nefunkcinių reikalavimų radimas</i>“ – suformuluojami sistemos nefunkciniai reikalavimai. Tai reikalavimai sistemos našumui, naudojamumui, patogumui, plečiamumui ir kt.</p> <p>„<i>Realizavimo apribojimų specifikavimas</i>“ – sudaromas apribojimų, kurių derėtų laikytis realizuojant sistemą, sąrašas. Apribojimai gali būti ir rekomendacinio pobūdžio. Pavyzdžiui tai galėtų būti rekomendacijos sistemos architektūriniais sprendimams, kurios išplaukia iš reikalavimų sistemos plečiamumui ir pan.</p> <p>„<i>Prezencijų kūrimas</i>“ – kuriamos supažindinimo su sistemos reikalavimais prezencijos. Šios prezencijos bus panaudotos projekto komandos susitikimuose.</p>	

ETAPAS 4: Reikalavimų modeliavimas / formalizavimas		
<b>PAAIŠKINIMAS</b>	Šiame etape atliekamas išanalizuotų reikalavimų modeliavimas ir formalizavimas turimomis CASE priemonėmis. Reikalavimai apibrėžiami vienareikšmiškais struktūromis. Priklausomai nuo sudėtingumo, reikalavimai parengiami realizacijai arba dekompozicijai į savybes ar smulkesnius reikalavimus.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Reikalavimų analizės dokumentai - Techniniai, sisteminiai reikalavimai	- Sudaryti formalūs, nevienareikšmiški reikalavimų modeliai, paruošti realizacijai - Nustatyti tarpusavyje susijusių reikalavimų ryšiai	Reikalavimų modeliavimas ir formalizavimas didžiąją projektavimo fazės laiką užimantis etapas. Jis turėtų užtrukti du trečdalius projektavimui skirtą laiką. Idealiu atveju reikalavimams taip pat turi būti sudaromas testavimo planas.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Projektuotojas - Testuotojas	- Objektiškai-orientuota analizė - Struktūrinė analizė - Testavimo planavimas	- CASE įrankiai reikalavimų modeliavimui. - Reikalavimų specifikavimo standartai
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center; background-color: #f2f2f2; margin: 0;">Projektuotojas</p>  <pre> graph TD     Start(( )) --&gt; A(Reikalavimų dokumentų analize)     A --&gt; B(Reikalavimų specifikavimo standartų analize)     B --&gt; C(Reikalavimų modelių sudarymas CASE įrankiais)     C --&gt; D(Reikalavimų ryšių nustatymas)     D --&gt; E(Teorinio reikalavimų testavimo plano sudarymas)     E --&gt; Stop((( )))           </pre> </div>	<p>„<i>Reikalavimų dokumentų analizė</i>“ – atliekama analizės metu sukurtų reikalavimų dokumentų apžvalga, kuriami reikalavimų modelių eskizai, atliekami kiti pasiruošimo darbai.</p> <p>„<i>Reikalavimų specifikavimo standartų analizė</i>“ – projektuotojas prieš pradėdamas modeliuoti reikalavimus, privalo susipažinti su egzistuojančiais reikalavimų specifikavimo standartais. Standartų laikymasis modeliuojant reikalavimus, palengvina modelius analizuojančių programuotojų darbą.</p> <p>„<i>Reikalavimų modelių sudarymas CASE įrankiais</i>“ – kuriamos įvairios reikalavimus apibrėžiančios diagramos, suprojektuojama sistemos architektūra. Surenkama techninė informacija, kuri bus panaudota atliekant reikalavimų realizavimą.</p> <p>„<i>Reikalavimų ryšių nustatymas</i>“ – nurodomi vienas nuo kito priklausantys, ar kaip nors kitaip vienas kitą įtakojantys reikalavimai.</p> <p>„<i>Teorinio reikalavimų testavimo plano sudarymas</i>“ – projektuotojas ir testuotojų komanda, iš sudarytų reikalavimų modelių sudaro preliminarų, teoriškai pagrįstą reikalavimų testavimo planą. Planas yra teorinis, todėl kad realizacijos metu, reikalavimai dažnai keičiasi, todėl pilną reikalavimų testavimo atvejų modelį galima sudaryti tik turint realizuotą sistemą ar jos dalį.</p>	

ETAPAS 5: Reikalavimų skaidymas į savybes		
<b>PAAIŠKINIMAS</b>	Šiame etape atliekamas sudėtingų reikalavimų skaidymas į smulkesnes dalis – savybes. Savybės ( <i>angl. features</i> ) tai konkretūs nurodymai, kaip realizuoti vieną ar kitą reikalavimą. Reikalavimų skaidymas į savybes padeda programuotojams lengviau suvokti reikalavimo struktūrą ir	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Reikalavimų dokumentai - Reikalavimų modeliai	- Reikalavimai suskaidyti į savybes – labiau suprantamus ir labiau valdomus vienetus - Savybes priklausančias tam pačiam reikalavimui galima priskirti skirtingiems programuotojams ar jų grupėms.	Reikalavimų skaidymas į savybes tęsiasi visą reikalavimų modeliavimo ir formalizavimo etapo laiką (4 etapas).
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Projektuotojas	- Reikalavimų dekompozicija. <sup>1</sup>	- Teksto redaktorius reikalavimų dekompozicijai dokumentuoti. - CASE įrankis reikalavimų ir savybių diagramų modeliavimui.
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
 <pre> graph TD     A[Projektuotojas] --&gt; B(Reikalavimų struktūros analizė)     B --&gt; C(Reikalavimų dekompozicija)     C --&gt; D(Savybių išskyrimas)     D --&gt; E(Savybių dokumentavimas ir formalizavimas)     E --&gt; F(Reikalavimų dekompozicijos dokumentavimas)     F --&gt; G(( ))   </pre>	<p>„<i>Reikalavimų struktūros analizė</i>“ – analizuojama reikalavimų struktūra, išskiriami reikalavimų realizavimo žingsniai, numatomas reikalavimo realizacijos vaidmuo veikiančioje sistemoje. Analizė yra būtina norint tinkamai „susmulkinti“ reikalavimus į suprantamus, lengvai realizuojamus vienetus.</p> <p>„<i>Reikalavimų dekompozicija</i>“ – reikalavimai skaidomi į smulkesnius, labiau konkrečius reikalavimus, skaidymas vykdomas tol, kol reikalavimai tampa užduotimis, kurias turi realizuoti programuotojai.</p> <p>„<i>Savybių išskyrimas</i>“ – iš reikalavimų dekompozicijoje gautų reikalavimų išskiriami, „apčiuopiami“ neformalia kalba paaiškinami reikalavimai, kurie yra pavadinami sistemos savybėmis.</p> <p>„<i>Savybių dokumentavimas ir formalizavimas</i>“ – išskirtos savybės papildomai dokumentuojamos ir nors tai ir trivialūs ir gerai suprantami programuotojams reikalavimai, apibrėžiami formaliai.</p> <p>„<i>Reikalavimų dekompozicijos dokumentavimas</i>“ – dokumentuojama reikalavimų skaidymo logika. Nurodoma, kokių reikalavimo skaidymo principų buvo laikomasi atliekant reikalavimų dekompoziciją, pagal kokius kriterijus išskirtos sistemos savybės.</p>	

<sup>1</sup> Reikalavimų dekompozicija (*angl. Requirement decomposition*) – „Requirements decomposition, a critical systems engineering function, is key to product success.“ [[http://www.rbccorp.com/process/2\\_requirements\\_decomposition.asp](http://www.rbccorp.com/process/2_requirements_decomposition.asp)]

ETAPAS 6: Reikalavimų priskyrimas programuotojams		
<b>PAAIŠKINIMAS</b>	Šiame etape išanalizuoti, sumodeliuoti ir paruošti realizacijai reikalavimai priskiriami programuotojams atsakingiems už jų realizavimą. Po šio etapo reikalavimas tampa programuotojo užduotimi.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Reikalavimų dokumentai - Reikalavimų modeliai - Programuotojų darbo patirties anketos	- Reikalavimai priskirti atsakingiems už realizaciją programuotojams. - Programuotojams pateikta visa įmanoma informacija apie reikalavimus ir jų realizavimo metodus.	Reikalavimų skirstymas programuotojams atliekamas baigus vieno arba kelių susijusių reikalavimų modeliavimą ir turėtų trukti ne ilgiau nei vieną darbo valandą per dieną.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>IRANKIAI</b>
- Projektuotojas - Projektų vadovas	- Paskirstymas pagal reikalavimų realizavimo sudėtingumą atitinkamą patirtį turintiems programuotojams.	-
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">Projektu vadovas   Projektuotojas</p> <pre> graph TD     Start(( )) --&gt; A[Programuotojų darbo patirties anketų analize]     A --&gt; B[Reikalavimų specifikacijų analize]     B --&gt; C[Reikalavimų priskyrimas programuotojams pagal patirtį ir specializacijos sritį]     C --&gt; D[Susitikimo su programuotojais organizavimas]     D --&gt; End(( ))           </pre> </div>	<p>„<i>Programuotojų darbo patirties anketų analizė</i>“ – analizuojama programuotojų darbo patirtis, jų specializacijos sritys, pasiekimai ir darbai ankstesniuose projektuose.</p> <p>„<i>Reikalavimų specifikacijų analizė</i>“ – pakartotinai peržiūrimos reikalavimų specifikacijos, nustatomi reikalavimų prioritetai, sudaromas preliminarus darbų sąrašas, kuriuos reikia atlikti.</p> <p>„<i>Reikalavimų priskyrimas programuotojams pagal patirtį ir specializacijos sritį</i>“ – pagal ankstesniuose veiksmuose atliktų analizių rezultatus sudaromas reikalavimų paskirstymo programuotojams planas. Reikalavimai programuotojams priskiriami pagal jų specializavimosi sritį, kad jie būtų realizuoti maksimaliai kokybiškai ir greitai.</p> <p>„<i>Susitikimo su programuotojais organizavimas</i>“ – jeigu priskiriamas didelis reikalavimų kiekis, surengiamas programuotojų ir kitų asmenų dalyvaujančių kūrimo procese, susitikimas. Susitikime aptariamos realizacijos problemos, diskutuojama apie programuotojams priskirtus reikalavimus, prireikus, atliekami priskyrimo pakeitimai.</p>	

ETAPAS 7: Savybių priskyrimas programuotojams		
<b>PAAIŠKINIMAS</b>	Šiame etape dokumentuotos ir realizacijai paruoštos savybės priskiriamos programuotojams atsakingiems už tų savybių realizavimą. Po šio etapo savybė tampa programuotojo užduotimi.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Reikalavimų susijusių su savybėmis dokumentacija - Savybių aprašymai - Programuotojų darbo patirties anketos	- Savybės priskirtos atsakingiems už realizaciją programuotojams. - Programuotojams pateikta reikalinga informacija reikalavimams realizuoti.	Savybių skirstymas programuotojams atliekamas skirstant reikalavimus (6 etapas) arba po reikalavimų paskirstymo. Šis darbas turėtų trukti ne ilgiau nei vieną darbo valandą per dieną.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Projektuotojas - Projektų vadovas	- Paskirstymas pagal savybių realizavimo sudėtingumą atitinkamą patirtį turintiems programuotojams.	-
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">Projektu vadovas   Projektuotojas</p> <pre> graph TD     Start(( )) --&gt; A[Programuotojų darbo patirties anketų analizė]     A --&gt; B[Reikalavimų susijusių su savybe analizė]     B --&gt; C[Savybių priskyrimas programuotojams pagal patirtį ir specializacijos sritį]     C --&gt; D{Ar savybės aptartos reikalavimų priskyrimo susitikime?}     D -- TAIP --&gt; E[Susitikimo su programuotojais organizavimas]     D -- NE --&gt; D     E --&gt; End((( )))           </pre> </div>	<p>„<i>Programuotojų darbo patirties anketų analizė</i>“ – analizuojama programuotojų darbo patirtis, jų specializacijos sritys, pasiekimai ir darbai ankstesniuose projektuose.</p> <p>„<i>Reikalavimų susijusių su savybe analizė</i>“ – pakartotinai peržiūrimi reikalavimai susiję su priskiriamomis savybėmis, nustatomi savybių prioritetai.</p> <p>„<i>Savybių priskyrimas programuotojams pagal patirtį ir specializacijos sritį</i>“ – pagal ankstesniuose veiksmuose atliktų analizių rezultatus sudaromas savybių paskirstymo programuotojams planas. Savybės programuotojams priskiriamos pagal jų specializavimosi sritį, kad jos būtų realizuotos maksimaliai kokybiškai ir greitai.</p> <p>„<i>Susitikimo su programuotojais organizavimas</i>“ – jeigu priskiriamos savybės nebuvo aptartos ankstesniuose reikalavimų priskyrimo susitikimuose arba priskiriamas didelis savybių kiekis, surengiamas programuotojų ir kitų asmenų dalyvaujančių kūrimo procese, susitikimas. Susitikime aptariamos realizacijos problemos, diskutuojama apie programuotojams priskirtas savybes ir reikalavimus, pritekus, atliekami priskyrimo pakeitimai.</p>	

ETAPAS 8: Reikalavimų realizavimas		
<b>PAAIŠKINIMAS</b>	Šiame etape programuotojas analizuoja susijusius reikalavimus ir su jais pateiktą reikalavimų dokumentaciją. Pagal pateiktus reikalavimų modelius, atlieka programinių modulių programavimą ir sintezę. Baigęs realizavimo darbus, programuotojas reikalavimą pažymi kaip realizuotą ir paruoštą testavimui.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Projekto informacija - Reikalavimų aprašymai - Reikalavimų modeliai - Susitikimuose išspręstų reikalavimų realizavimo problemų aprašymai	- Klasių diagramos - Reikalavimų realizavimo išeities tekstai - Derinimui ( <i>angl. debug</i> ) skirti vykdomieji failai ir bibliotekos	Reikalavimų realizacijos laikas yra skirtingas kiekvienam reikalavimui. Tikslus realizavimo darbų planus kiekvienam atvejui sudaro projektų vadovas.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Programuotojas	- Objektinis programavimas - CASE įrankių palaikomas automatinis kodo generavimas - Projektavimo šablonai ( <i>angl. design patterns</i> ) - Pakartotinis kodo ir komponentų panaudojimas	- Išeities tekstų redaktorius - Kompiliatorius - Derintuvas ( <i>angl. debugger</i> ) - CASE priemonės klasių diagramų kūrimui ir automatiniam kodo generavimui. - Sistemos (duomenų bazių ir kt.) administravimo įrankiai.
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; D1{Projekto informacija isnagrineta?}     D1 -- NE --&gt; A1(Projekto informacijos analize)     D1 -- TAIP --&gt; A2(Reikalavimų modelių analize)     A1 --&gt; A2     A2 --&gt; A3(Klasių diagramų kūrimas ir kodo generavimas)     A3 --&gt; A4(Programavimas)     A4 --&gt; A5(Testavimas ir derinimas)     A5 --&gt; D2{Tenkina reikalavimus?}     D2 -- NE --&gt; A4     D2 -- TAIP --&gt; A6(Vykdomųjų modulių ir bibliotekų kompiliavimas)     A6 --&gt; End((( )))   </pre>	<p>„Projekto informacijos analizė“ – programuotojai pakartotinai susipažįsta su projekto dokumentais ir pradiniais, neformaliais, reikalavimais.</p> <p>„Reikalavimų modelių analizė“ – programuotojai analizuoja projektuotojų parengtus reikalavimų modelius. Reikalavimų modeliuose, reikalavimai pateikiami formaliai ir vienareikšmiškai. Taip pat analizuojami susiję reikalavimai į kuriuos programuotojas turi atsižvelgti.</p> <p>„Klasių diagramų kūrimas ir kodo generavimas“ – programuotojai atlieka projektavimą, klasių diagramų kūrimą ir kodo „griaučių“ generavimą.</p> <p>„Programavimas“ – Programuotojai atlieka algoritmų, reikalingų reikalavimams realizuoti, programavimą.</p> <p>„Testavimas ir derinimas“ – Atliekamas sukurtų programinių modulių testavimas ir derinimas (trasavimas ir t.t.). Atliekant derinimą surandamos ir ištaisomos realizacijos klaidos. Testavimas ir derinimas atliekamas tol, kol programuotojas pasiekia rezultatą aprašytą reikalavimų specifikacijoje.</p> <p>„Vykdomųjų modulių ir bibliotekų kompiliavimas“ – Atliekamas galutinio programinio kodo kompiliavimas ir paruošimas testavimui. Sukompilijuoti programiniai moduliai ir bibliotekos patalpinamos į testavimui skirtas saugyklas.</p>	

ETAPAS 9: Savybių realizavimas		
<b>PAAIŠKINIMAS</b>	Šiame etape programuotojas analizuoja pateiktą savybės aprašymą, su savybe susietus reikalavimus ir jų modelius. Pagal aprašymą atlieka naujų programinės įrangos savybių programavimą. Baigęs realizavimo darbus, programuotojas savybę pažymi kaip realizuotą ir paruoštą testavimui.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Projekto informacija - Savybių aprašymai - Reikalavimų susijusių su savybėmis modeliai - Susitikimuose išspręstų savybių realizavimo problemų aprašymai	- Klasių diagramos - Savybių realizavimo išeities tekstai - Derinimui ( <i>angl. debug</i> ) skirti vykdomieji failai ir bibliotekos	Savybių realizacijos laikas yra skirtingas kiekvienam reikalavimui. Tikslius realizavimo darbų planus sudaro projektų vadovas. Mažiau svarbioms savybėms realizavimo laiką suplanuoja pats programuotojas.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Programuotojas	- Objektinis programavimas - Projektavimo šablonai ( <i>angl. design patterns</i> ) - Pakartotinis kodo ir komponentų panaudojimas	- Išeities tekstų redaktorius - Kompiliatorius - Derintuvas ( <i>angl. debugger</i> ) - CASE įrankiai klasių diagramų kūrimui.
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; D1{Projekto informacija isnagrineta?}     D1 -- NE --&gt; A1[Projekto informacijos analize]     D1 -- TAIP --&gt; A2[Susietu su savybe reikalavimu modelių analize]     A1 --&gt; A2     A2 --&gt; A3[Klasių diagramų kurimas ir atnaujinimas]     A3 --&gt; A4[Programavimas]     A4 --&gt; A5[Testavimas ir derinimas]     A5 --&gt; D2{Tenkina reikalavimus?}     D2 -- NE --&gt; A4     D2 -- TAIP --&gt; A6[Vykdomųjų modulių ir bibliotekų kompiliavimas]     A6 --&gt; End((( )))   </pre>	<p>„Projekto informacijos analizė“ – programuotojai pakartotinai susipažįsta su projekto dokumentais ir pradiniais, neformaliais, reikalavimais.</p> <p>„Susietų su savybe reikalavimų modelių analizė“ – programuotojai analizuoja su savybe susietų reikalavimų modelius ir specifikacijas. Šis veiksmas yra labai svarbus jeigu savybė yra reikalavimo sudedamoji dalis.</p> <p>„Klasių diagramų kūrimas ir atnaujinimas“ – programuotojai atlieka egzistuojančių klasių diagramų atnaujinimą, papildydami jas savybei realizuoti reikalingomis klasėmis.</p> <p>„Programavimas“ – Programuotojai atlieka algoritmų, reikalingų savybėms realizuoti, programavimą, panaudojant reikiamas technologijas. Klasių kodas pakeičiamas pagal atliktus klasių diagramų pakeitimus.</p> <p>„Testavimas ir derinimas“ – Atliekamas sukurtų programinių modulių testavimas ir derinimas (trasavimas ir t.t.). Atliekant derinimą surandamos ir ištaisomos savybių realizacijos klaidos. Testavimas ir derinimas atliekamas tol, kol programuotojas pasiekia rezultatą aprašytą savybės aprašyme ir susijusių reikalavimų specifikacijose.</p> <p>„Vykdomųjų modulių ir bibliotekų kompiliavimas“ – Atliekamas galutinio programinio kodo kompiliavimas ir paruošimas testavimui. Sukompilijuoti programiniai moduliai ir bibliotekos patalpinamos į testavimui skirtas saugyklas.</p>	

ETAPAS 10: Rezultatų dokumentavimas		
<b>PAAIŠKINIMAS</b>	Šiame etape programuotojai aprašo atliktus darbus, naudodamiesi CASE įrankiais kuria ir dokumentuoja architektūros bei realizacijos diagramas, kuriomis prireikus galėtų pasinaudoti kiti programuotojai gavę darbo plėtojimo ar taisymo užduotis.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Reikalavimų dokumentai - Klasių diagramos - Išsities tekstai - Sistemos architektūros dokumentacija	- Dokumentuotos klasių diagramos - Aprašyti realizuojami algoritmai ir panaudotos technologijos - Užpildytos darbų atlikimo ataskaitos	Atliktų darbų dokumentavimą programuotojai privalo atlikti realizavę vieną ar kelis mažesnius reikalavimus. Rezultatų dokumentavimui programuotojas teoriškai turi skirti 2-3 valandas kas antrą darbo dieną.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Programuotojas	- UML modeliavimas - Darbų atlikimo ataskaitų pildymas	- Teksto redaktorius rezultatams dokumentuoti - CASE įrankis palaikantis UML
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; A(Reikalavimų modelių analize)     A --&gt; B(UML klasių diagramų redagavimas ir dokumentavimas)     B --&gt; C(Realizuotu algoritmu ir technologijų dokumentavimas)     C --&gt; D(Darbu atlikimo ataskaitų pildymas)     D --&gt; End((( )))   </pre>	<p>„<i>Reikalavimų modelių analize</i>“ – programuotojai analizuoja projektuotojų parengtus reikalavimų modelius. Reikalavimų modeliuose, reikalavimai pateikiami formaliai ir vienareikšmiškai. Pakartotinai patikrinama ar realizacijos atitinka reikalavimus.</p> <p>„<i>UML klasių diagramų redagavimas ir dokumentavimas</i>“ – CASE įrankiais dokumentuojamos sukurtos klasių diagramos. Atliekamas automatinis dokumentacijos generavimas. Dokumentacija papildoma programuotojo komentarais.</p> <p>„<i>Realizuotų algoritmų ir technologijų dokumentavimas</i>“ – atliekama panaudotų technologijų ir algoritmų apžvalga. Analizuojami ir dokumentuojami atskiri algoritmo žingsniai, parametrai, rezultatai, pateikiami algoritmo panaudojimo pavyzdžiai.</p> <p>„<i>Darbų atlikimo ataskaitų pildymas</i>“ – atliekamas darbo valandų ataskaitų pildymas.</p>	



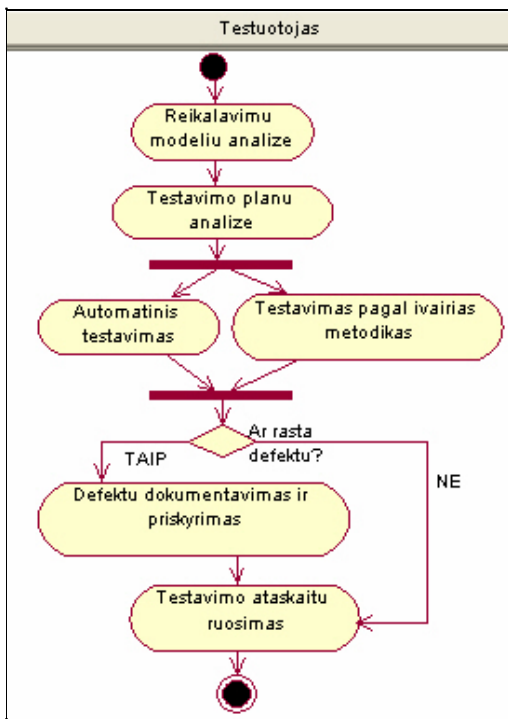
**ETAPAS 11: Testavimas, defektų priskyrimas programuotojams**

<b>PAAIŠKINIMAS</b>	Šiame etape atliekamas programuotojų atliktų darbų testavimas, testavimo rezultatų analizė, paruošiamos testavimo darbų ataskaitos, sukuriama defekto dokumentacija, nurodomi su defektu susiję reikalavimai. Defektas su visa pateikta informacija priskiriamas programuotojui. Po šio etapo defektas tampa programuotojo užduotimi.
---------------------	---

<b>DUOMENYS</b> - Reikalavimų modeliai - Reikalavimų testavimo planai	<b>REZULTATAI</b> - Dokumentuoti defektai - Nustatytas defektų taisymo prioritetas ir įtaka galutiniam produktui - Užpildytos testavimo darbų ataskaitos	<b>TRUKMĖ</b> Testavimas ir defektų paieška tęsiasi visą testavimo fazės laiką, iki pat projekto išleidimo. Testavimo laikų grafikus sudaro projektų vadovas pagal projektuotojų parengtą reikalavimų testavimo planą.
---	---	---

<b>RESURSAI</b> - Testuotojas - Projektų vadovas	<b>METODIKOS</b> - „Juodos dėžės“ testavimas - „Baltos dėžės“ testavimas - „Stresinis“ testavimas - Automatinis testavimas - Scenarijų testavimas	<b>ĮRANKIAI</b> - Testavimo duomenų generatoriai - Automatinės testavimo sistemos - Sistemos (duomenų bazių ir kt.) administravimo įrankiai.
--	--	---

<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>
-----------------	-----------------------------



„*Reikalavimų modelių analizė*“ – testuotojai analizuoja projektuotojų parengtus reikalavimų modelius. Reikalavimų modeliuose, reikalavimai pateikiami formaliai ir vienareikšmiškai.

„*Testavimo planų analizė*“ – analizuojami projektuotojo ir testuotojo parengti reikalavimų testavimo planai. Pagal testavimų planus sudaromi testavimo scenarijai, rengiami pasitarimai (dalyvaujant projekto vadovui), pasirošama testavimo darbams.

„*Automatinis testavimas*“ – panaudojama automatinio modulių testavimo metodika. Automatiniam testavimui, pagal scenarijus, sukuriama testavimo duomenų masyvai, padengiantys visus testuojamus kelius. Automatinis testavimas padeda surasti defektus retai vartotojo naudojamame funkcionalume, kur žmogus klaidų dažniausiai nepastebi.

„*Testavimas pagal įvairias metodikas*“ – pasirenkama viena ar keletas testavimo metodikų („Baltos dėžės“, „Juodos dėžės“ ir pan.) ir atliekamas sistemos testavimas.

„*Defektų dokumentavimas ir priskyrimas*“ – Rastos klaidos dokumentuojamos, sudaromi klaidų atkūrimo žingsniai ir priskiriami programuotojams kaip defektai.

„*Testavimo ataskaitų ruošimas*“ - paruošiamos detalios ataskaitos apie atliktus testavimo darbus, aptiktus defektus, jų kiekį, taip pat reikalavimus, kurių realizacijose aptikta daugiausiai klaidų.

ETAPAS 12: Ištaisytų defektų peržiūra ir įvertinimas		
<b>PAAIŠKINIMAS</b>	Šiame etape atliekama programuotojų ištaisytų defektų analizė ir pakartotinis modulio, susijusio su defektu testavimas. Atlikus pakartotinį testavimą pataisytas funkcionalumas įvertinamas pagal susijusius reikalavimus.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Ištaisyti programinės įrangos defektai - Programuotojų atliktų darbų rezultatų dokumentai - Reikalavimų modeliai - Reikalavimų testavimo planai	- Ištaisytų defektų patvirtinimas ir „uždarymas“ - Neištaisyti ar ištaisyti ne pagal reikalavimus defektai paruošiami pakartotiniam taisymui (15 etapas).	Tai vienas pagrindinių ir daugiausiai laiko reikalaujantis kokybės užtikrinimo fazės etapas. Jis vyksta lygiagrečiai su testavimo etapu (11 etapas). Tęsiasi visą testavimo fazės laiką, iki pat projekto išleidimo.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Testuotojas	- Pakartotinis (regresinis) testavimas	- Teksto redaktorius dokumentacijai redaguoti
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; A(Ištaisytų defektu analize)     A --&gt; B(Su defektais susijusiu reikalavimu analize)     B --&gt; C(Testavimas)     C --&gt; D{Ištaisyta pagal reikalavimus?}     D -- NE --&gt; E(Grazinti defekta pakartotinai realizacijai)     D -- TAIP --&gt; F(Defekto uždarymas)     E --&gt; G(Testavimo ataskaitu ruošimas)     F --&gt; G     G --&gt; End((( )))   </pre>	<p>„<i>Ištaisytų defektų analizė</i>“ – atliekama programuotojų ištaisytų defektų peržiūra. Analizuojami jų pateikti problemos sprendimo aprašymai. Gilinamasi į defekto esmę, nustatoma pakeitimų, atliktų taisant defektą, grėsmė kitoms, jau egzistuojančioms, sistemos funkcijoms.</p> <p>„<i>Su defektais susijusių reikalavimų analizė</i>“ – analizuojami reikalavimai susiję su ištaisytais defektais. Pagal analizės rezultatus patikrinamas ir jei reikia koreguojamas testavimo planas.</p> <p>„<i>Testavimas</i>“ – atliekamas sistemos testavimas, defektų ištaisymo taisyklingumui ir atitikimui reikalavimams, patikrinti. Jeigu nustatoma, kad atlikti taisymo darbai gali įtakoti kitas sistemos funkcijas, atliekamas pakartotinis (regresinis) tų sistemos dalių testavimas.</p> <p>„<i>Grazinti defekta pakartotinei realizacijai</i>“ – neteisingai, ne pagal reikalavimus ar ištaisyti taip, kad sutrikdo kitų sistemos funkcijų darbą, defektai pakartotinai priskiriami už jų ištaisymą atsakingam programuotojui.</p> <p>„<i>Defekto uždarymas</i>“ – ištaisyti defektai po testavimo pažymimi kaip ištaisyti. „Uždarytų“ defektų informacija saugojama visą projekto gyvavimo laiką.</p> <p>„<i>Testavimo ataskaitų ruošimas</i>“ - paruošiamos detalios ataskaitos apie atliktus testavimo darbus, aptiktus defektus, jų kiekį, taip pat reikalavimus, kurių realizacijose aptikta daugiausiai klaidų.</p>	

ETAPAS 13: Realizuotų reikalavimų ir savybių peržiūra ir įvertinimas		
<b>PAAIŠKINIMAS</b>	Šiame etape atliekama programuotojų realizuotų reikalavimų ir savybių analizė ir testavimas. Atlikus testavimą naujas funkcionalumas įvertinamas pagal veikimo taisyklumą ir susijusius reikalavimus.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Realizuoti reikalavimai ir savybės - Programuotojų atliktų darbų rezultatų dokumentai - Reikalavimų modeliai - Reikalavimų testavimo planai	- Įvertinti programuotojų realizuoti reikalavimai - Reikalavimų specifikacijų neatitinkančios savybių ir reikalavimų realizacijos paruošiamos pakartotiniam realizavimui (15 etapas).	Šis etapas vyksta lygiagrečiai su testavimo ir defektų peržiūros etapais (11; 12 etapai). Tęsiasi visą testavimo fazės laiką, iki pat projekto išleidimo.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Testuotojas - Projektuotojas	- Pakartotinis (regresinis) testavimas	- Teksto redaktorius dokumentacijai redaguoti
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; A[Realizuotu reikalavimu ir savybiu analize]     A --&gt; B[Susijusiu reikalavimu analize]     B --&gt; C[Testavimas]     C --&gt; D{Realizuota pagal reikalavimus}     D -- NE --&gt; E[Grazinti pakartotiniam realizavimui]     D -- TAIP --&gt; F[Reikalavimo / Savybes uzdarymas]     E --&gt; F     F --&gt; G[Testavimo ataskaitu ruosimas]     G --&gt; End((( )))   </pre>	<p>„<i>Realizuotų reikalavimų ir savybių analizė</i>“ – atliekama programuotojų realizuotų reikalavimų ir savybių peržiūra. Analizuojami jų pateikti problemos sprendimo aprašymai. Gilinamasi į realizacijos metodus, nustatoma pakeitimų, atliekant realizavimą, grėsmė kitoms, jau egzistuojančioms, sistemos funkcijoms.</p> <p>„<i>Susijusių reikalavimų analizė</i>“ – analizuojami reikalavimai susiję su realizuotomis savybėmis ir reikalavimais. Pagal analizės rezultatus patikrinamas ir jei reikia koreguojamas testavimo planas.</p> <p>„<i>Testavimas</i>“ – atliekamas sistemos testavimas, savybių ir reikalavimų realizavimo taisyklumui patikrinti. Jeigu nustatoma, kad atlikti realizavimo darbai gali įtakoti kitas, jau egzistuojančias, sistemos funkcijas, atliekamas pakartotinis (regresinis) tų sistemos dalių testavimas.</p> <p>„<i>Grazinti pakartotiniam realizavimui</i>“ – neteisingai realizuoti, ne ar realizuoti taip, kad sutrikdo kitų sistemos funkcijų darbą, reikalavimai ir savybės pakartotinai priskiriami už jų realizavimą atsakingam programuotojui.</p> <p>„<i>Reikalavimo / Savybės uždarymas</i>“ – realizuoti reikalavimai / savybės po testavimo pažymimi kaip realizuoti. „Uždarytų“ reikalavimų / savybių informacija saugojama visą projekto gyvavimo laiką.</p> <p>„<i>Testavimo ataskaitų ruošimas</i>“ – paruošiamos detalios ataskaitos apie atliktus testavimo darbus, reikalavimus kurių realizacijose aptikta daugiausiai klaidų.</p>	

ETAPAS 14: Testavimo ataskaitų ruošimas		
<b>PAAIŠKINIMAS</b>	Šiame etape kokybės užtikrinimo padalinys paruošia detalias ataskaitas apie atliktus testavimo darbus, aptiktus defektus, jų kieki, realizuotus ir gražintus pakartotiniam realizavimui reikalavimus ir savybes.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
- Testuojamų modulių sąrašas - Aptiktų defektų, ne pagal reikalavimus realizuotų savybių ir reikalavimų sąrašas - Automatinio testavimo rezultatai	- Testavimo rezultatų suvestinė - Defektų analizės ataskaita - „Problematiškiausių“ reikalavimų ir savybių sąrašas. - Automatinio testavimo ataskaitos	Testavimo ataskaitos turėtų būti rašomos mažiausiai vieną kartą per savaitę. Ataskaitų rašymas gali trukti nuo 3 valandų iki visos darbo dienos priklausomai nuo atliktų darbų skaičiaus.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
- Testuotojas	- Testavimo ataskaitų šablonai	- Teksto arba specializuotas redaktorius ataskaitoms generuoti / užpildyti.
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     Start(( )) --&gt; A[Testavimo rezultatų analize]     A --&gt; B[Automatinio testavimo ataskaitų ruošimas]     B --&gt; C[Defektų ataskaitų ruošimas]     C --&gt; D[Jautriausių pakeitimams reikalavimų sąrašo sudarymas]     D --&gt; E[Testavimo rezultatų suvestinės ruošimas]     E --&gt; End((( )))   </pre>	<p>„<i>Testavimo rezultatų analizė</i>“ – analizuojami atliktų testavimo darbų rezultatai. Tiriama automatinio testavimo sistemų pateikti duomenys ir rezultatai.</p> <p>„<i>Automatinio testavimo ataskaitų ruošimas</i>“ – paruošiamos ataskaitos su dokumentuotomis automatinio testavimo sistemos duomenų ir rezultatų aibėmis.</p> <p>„<i>Defektų ataskaitų ruošimas</i>“ – paruošiamos defektų ataskaitos, kuriose pateikiami moduliai, kuriuose yra aptikti defektai, įvertinimas modulio funkcionalumo tinkamumas ir panaudojamumas. Sudaroma defektų radimo ir ištaisymo statistika, nurodomos silpniausios sistemos vietos.</p> <p>„<i>Jautriausių pakeitimams reikalavimų sąrašo sudarymas</i>“ – sudaromas sąrašas reikalavimų, kuriuose statistiškai pasitaiko daugiausiai defektų, taip pat kurie moduliai yra jautriausi pakeitimams.</p> <p>„<i>Testavimo rezultatų suvestinės ruošimas</i>“ – paruošiamas vienas dokumentas apibendrinantis visas šiame etape sudarytas ataskaitas. Šis dokumentas yra informacinio, ne techninio pobūdžio ir gali būti pateikiamas organizacijos administracijai, klientams ir kitiems suinteresuotiems asmenims.</p>	

<b>ETAPAS 15: Defektų, savybių ar reikalavimų pakartotinis priskyrimas / realizacijų atmetimas (pakartotiniam realizavimui / taisymui)</b>		
<b>PAAIŠKINIMAS</b>	Šiame etape, testavimo metu surasti netinkamai ištaisyti defektai, ne pagal reikalavimus realizuotos savybės ir reikalavimai yra „gražinami“ programuotojams pakartotiniam taisymui ar realizavimui.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
<ul style="list-style-type: none"> <li>- Neištaisyti arba ne pagal reikalavimus ištaisyti defektai</li> <li>- Neteisingai realizuoti reikalavimai ir savybės</li> <li>- Reikalavimų specifikacijos</li> <li>- Testavimo rezultatai</li> </ul>	<ul style="list-style-type: none"> <li>- Papildyta pakartotinai priskiriamo objekto dokumentacija.</li> <li>- Nurodyti papildomi reikalavimai, kuriais derėtų remtis atliekant realizavimą</li> <li>- Išnagrinėti ir pateikti problemos atkūrimo žingsniai ir testavimo duomenų aibės.</li> </ul>	Šis etapas vykdomas pasibaigus 12 arba 13 etapui. Jis nereikalauja daug laiko todėl turėtų būti vykdomas periodiškai, kiekvieną darbo dieną.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
<ul style="list-style-type: none"> <li>- Testuotojas</li> <li>- Projektų vadovas</li> </ul>	-	-
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center; margin: 0;"><b>Testuotojas</b></p> <pre> graph TD     Start(( )) --&gt; A[Pakartotinai realizuotinu / taisytinu objektu analize]     A --&gt; B[Pakartotinai realizuotinu / taisytinu objektu dokumentavimas]     B --&gt; C[Priskyrimas uz realizavima / taisyma atsakingam programuotojui]     C --&gt; End((( )))           </pre> </div>	<p>„Pakartotinai realizuotinu / taisytinu objektu analize“ – atliekama testavimo etape surastų ne pagal reikalavimus ištaisytu defektų ar netinkamai realizuotų savybių ir reikalavimų, pakartotinė analizė. Šios analizės tikslas surinkti kuo tikslesnius ir informatyvesnius duomenis, kurie bus panaudoti dokumentacijai.</p> <p>„Pakartotinai realizuotinu / taisytinu objektu dokumentavimas“ – remiantis atliktos analizės rezultatais, sukuriama detalūs dokumentai aprašantys surastas problemas ir galimus tų problemų sprendimo variantus.</p> <p>„Priskyrimas už realizavimą / taisymą atsakingam programuotojui“ – pakartotinai realizuotinas / taisytinas objektas priskiriamas programuotojui, atsakingam už klaidų likvidavimą.</p>	

ETAPAS 16: Naujų / Pasikeitusių reikalavimų įvedimas / atnaujinimas		
<b>PAAIŠKINIMAS</b>	Šiame etape vykdomas reikalavimų, pakitusių dėl įvairių realizavimo aspektų, atnaujinimas ir naujų sukūrimas. Susiję reikalavimai papildomi dokumentacija įrodančia reikalavimų pakeitimų būtinumą.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
<ul style="list-style-type: none"> <li>- Projekto dokumentacija</li> <li>- Reikalavimų specifikacijos ir modeliai</li> <li>- Projekto darbų planas</li> <li>- Nauji vartotojo reikalavimai</li> </ul>	<ul style="list-style-type: none"> <li>- Atnaujintos reikalavimų specifikacijos ir modeliai</li> <li>- Pakoreguoti projekto darbų planai</li> <li>- Sukurti / Pakoreguoti reikalavimų testavimo planai</li> </ul>	Reikalavimų pasikeitimo analizės darbai turėtų būti vykdomi periodiškai. Jiems turėtų būti skiriama 3-8 valandos per savaitę.
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
<ul style="list-style-type: none"> <li>- Projektų vadovas</li> <li>- Projektuotojas</li> <li>- Užsakovas</li> </ul>	<ul style="list-style-type: none"> <li>- Reikalavimų inžinerija</li> <li>- Susitikimai ir prezentacijos</li> <li>- Dialogai su užsakovais</li> <li>- UML panaudojimo atvejų modeliavimas</li> </ul>	<ul style="list-style-type: none"> <li>- CASE įrankiai reikalavimų modelių redagavimui</li> <li>- Prezentacijų kūrimo įrankiai</li> <li>- UML palaikantis CASE įrankis panaudojimo atvejų (<i>angl.</i> „Use-Case“) diagramoms kurti ir redaguoti</li> <li>- Reikalavimų specifikavimo standartai</li> </ul>
<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>	
<pre> graph TD     subgraph PM [Projektu vadovas]         Start(( )) --&gt; D1[Dialogai su užsakovu]         D1 --&gt; D1         D1 --&gt; R1[Reikalavimų nesutapimai, klaidos]         R1 --&gt; D2[Reikalavimų dokumentų keitimas]         D2 --&gt; D3[Susitikimo dėl pasikeitusių reikalavimų organizavimas]         D3 --&gt; End((( )))     end     subgraph DT [Projektuotojas]         D4[Reikalavimų modelių ir specifikacijų atnaujinimas]         D5[Naujų reikalavimų modeliavimas ir specifikavimas]     end     D2 --&gt; D4     D4 --&gt; D5   </pre>	<p>„Dialogai su užsakovu“ – projektų vadovas testavimo metu iškilusius klausimus sprendžia dialogų su užsakovu metodu. Iškilus nesusipratimams pradedamas reikalavimų keitimo procesas.</p> <p>„Reikalavimų dokumentų keitimas“ – projektų vadovas keičia reikalavimų dokumentus, iškilusiems nesusipratimams ir problemoms pašalinti.</p> <p>„Reikalavimų modelių ir specifikacijų atnaujinimas“ – projektuotojas atnaujiną pasikeitusių reikalavimų modelius ir specifikacijas.</p> <p>„Naujų reikalavimų modeliavimas ir specifikavimas“ – projektuotojas sukuria naujų reikalavimų, atsiradusių dėl pakeitimų, modeliavimą ir specifikacijų ruošimą.</p> <p>„Susitikimo dėl pasikeitusių reikalavimų organizavimas“ – projektų vadovas organizuoja projekto komandos susitikimą dėl pakitusių ir naujų reikalavimų, atnaujintų darbų grafiką.</p>	

**ETAPAS 17: Defektų, savybių ar reikalavimų pakartotinė realizacija / taisymas**

<b>PAAIŠKINIMAS</b>	Šį etapą kitaip galima pavadinti klaidų taisymo etapu. Jo metu analizuojama kokybės užtikrinimo padalinio (testuotojų) pateikta testavimo ataskaita, testavimo duomenys, problemos aprašymas ir galimi sprendimo variantai. Prireikus iš naujo analizuojami susiję reikalavimai, skaitoma dokumentacija. Pagal padarytas analizės išvadas atliekami reinžinerijos ir programavimo darbai.
---------------------	---

<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>
<ul style="list-style-type: none"> <li>- Defektai</li> <li>- Pakartotinam realizavimui gražintos savybės ir reikalavimai</li> <li>- Reikalavimų specifikacijos</li> <li>- Testavimo rezultatai</li> <li>- Problemos atkūrimo žingsniai ir testavimo duomenų aibės.</li> </ul>	<ul style="list-style-type: none"> <li>- Pakartotinai realizuoto objekto išeities tekstai</li> <li>- Derinimui (<i>angl. debug</i>) skirti vykdomieji failai ir bibliotekos</li> </ul>	Pakartotinės realizacijos trukmė labai priklauso nuo pakartotinai realizuojamo objekto taip pat pradinės sistemos architektūros ir jos lankstumo. Daug laiko reikalaujantys pakeitimai turi būti derinami su projekto vadovu.

<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>
<ul style="list-style-type: none"> <li>- Programuotojas</li> </ul>	<ul style="list-style-type: none"> <li>- Reinžinerija</li> <li>- Projektavimo šablonai (<i>angl. design patterns</i>)</li> <li>- Išeities tekstų trasavimas</li> </ul>	<ul style="list-style-type: none"> <li>- Išeities tekstų redaktorius</li> <li>- Kompiliatorius</li> <li>- Derintuvas (<i>angl. debugger</i>)</li> <li>- CASE įrankiai klasių diagramų redagavimui</li> </ul>

<b>VEIKSMAI</b>	<b>VEIKSMŲ PAAIŠKINIMAI</b>
<pre> graph TD     Start(( )) --&gt; A[Pakartotinai realizuojamų / taisomų objektų analizė]     A --&gt; B[Susijusių reikalavimų analizė]     B --&gt; C[Testavimo rezultatų analizė ir problemų atkūrimas]     C --&gt; D[Programavimas]     D --&gt; E[Testavimas ir derinimas]     E --&gt; F{Tenkina reikalavimus?}     F -- NE --&gt; D     F -- TAIP --&gt; G[Vykdomųjų modulių ir bibliotekų kompiliavimas]     G --&gt; End((( )))     </pre>	<p>„Pakartotinai realizuojamų / taisomų objektų analizė“ – analizuojami pakartotiniam taisymui ar realizavimui priskirtų defektų, savybių ir reikalavimų aprašymai ir dokumentacija.</p> <p>„Susijusių reikalavimų analizė“ – analizuojami reikalavimai susiję su pakartotinai taisomais ir realizuojamais defektais savybėmis ar reikalavimais.</p> <p>„Testavimo rezultatų analizė ir problemų atkūrimas“ – programuotojas analizuoja testuotojo pateiktus testavimo duomenis, rezultatus ir problemų aprašymus. Pagal nurodytus problemos atkūrimo žingsnius, atkuria ir lokalizuoja problemą.</p> <p>„Programavimas“ – Programuotojai atlieka algoritmų, kuriuose aptiktos problemos, taisymą.</p> <p>„Testavimas ir derinimas“ – Atliekamas pataisytų programinių modulių testavimas ir derinimas (trasavimas ir t.t.). Testavimas ir derinimas atliekamas tol, kol programuotojas įsitikina, kad sistema veikia teisingai, su visais testuotojų pateiktais kontroliniais duomenimis.</p> <p>„Vykdomųjų modulių ir bibliotekų kompiliavimas“ – Atliekamas pataisyto programinio kodo kompiliavimas ir paruošimas testavimui. Sukompiliuoti programiniai moduliai ir bibliotekos patalpinamos į testavimui skirtas saugyklas.</p>

ETAPAS 18: Projekto „išleidimas“			
<b>PAAIŠKINIMAS</b>		Šis etapas užbaigia defektų, savybių ir reikalavimų valdymo etapų seka. Tai yra paskutinis žingsnis nagrinėjamame projekto kūrimo procese. Prasidėjus projekto palaikymo / plėtojimo fazei (kuri atskirai šiame metode nenagrinėjama), procesas, išskyrus kai kuriuos projektavimo etapus, vyksta iš pradžių.	
<b>DUOMENYS</b>	<b>REZULTATAI</b>	<b>TRUKMĖ</b>	
- Projekto modulių išeiigos tekstai - Projekto dokumentacija - Darbų atlikimo dokumentai	- Projekto vykdomieji failai ir bibliotekos - Vartotojų dokumentacija - Realizavimo darbų pabaigos dokumentai	Projekto išleidimas, priklausomai nuo projekto dydžio, gali užtrukti nuo 1 darbo dienos iki kelių savaičių.	
<b>RESURSAI</b>	<b>METODIKOS</b>	<b>ĮRANKIAI</b>	
- Projektų vadovas - Projektuotojas - Programuotojai - Testuotojai - Administracija	- „Išleidimo“ valdymas ( <i>angl. Release Management</i> )	- Atskaitų generavimo įrankiai - Kompiliatorius - Programinės įrangos diegimo vedlių ( <i>angl. install wizard</i> ) kūrimo programinė įranga.	
<b>VEIKSMAI</b>			
<pre> graph TD     subgraph "Projektu vadovas"         A((Nustato galutinę išleidimo data))         B([Ruosia projekto išleidimo dokumentaciją])         C([Galutinės versijos kompiliavimas ir kurimas])         D([Sistemos diegimo vedlio kurimas])         E([Produkto pristatymas užsakovui])         F([Projektas išleistas])     end      subgraph "Projektuotojas"         G([Ruosia projekto techninę dokumentaciją])     end      subgraph "Programuotojai"         H([Likusių darbų vykdymas])         I{Ar atlikti visi darbai ir istaisytos klaidos?}         J([Patalpina programinį kodą į duomenų bazę])         K([Programavimo darbai baigti])     end      subgraph "Testuotojai"         L([Projekto kokybės kriterijų protokolo pildymas])     end      A --&gt; B     A --&gt; G     B --&gt; C     G --&gt; C     C --&gt; D     D --&gt; E     E --&gt; F     H --&gt; I     I -- TAIP --&gt; J     J --&gt; K     I -- NE --&gt; H     L --&gt; E   </pre>			



**VEIKSMŲ PAAIŠKINIMAI**

„*Nustato galutinę išleidimo datą*“ – projektų vadovas informuoja projekto komandą apie galutinę projekto išleidimo datą (*angl. deadline*). Taip pat rengiami susitikimai, kuriuose aptariami neatlikti darbai ir numatomas reikiamas laikas jiems užbaigti.

„*Ruošia projekto išleidimo dokumentaciją*“ – projektų vadovas sudaro projekto pristatymo dokumentus ir prezentacijas. Aprašo pasiektus rezultatus ir projekto plėtojimo galimybes, sudaro projekto kaštų ataskaitas.

„*Ruošia projekto techninę dokumentaciją*“ – projektuotojas atlieka projekto techninės dalies dokumentavimą, aprašo panaudotas technologijas bei naujai sukurtus sprendimus. Paruošia sistemos našumo ir efektyvumo ataskaitas.

„*Likusių darbų vykdymas*“ – programuotojai realizuoja likusius reikalavimus, savybes bei taiso defektus. Šiame etape likę defektai ir reikalavimai privalo būti paties žemiausio prioriteto ir neįtakoti galutinio produkto funkcionalumo.

„*Patalpina programinį kodą į duomenų bazę*“ – programuotojai padeda galutines programinio kodo versijas į kodo saugojimo duomenų bazę.

„*Projekto kokybės kriterijų protokolo pildymas*“ – testuotojai užpildo kiekvienai programinei įrangai kuriančiai įmonei specifinį kokybės kriterijų protokolą. Juose pagrįstai įvertinami pagrindiniai programinės įrangos kriterijai.

„*Galutinės versijos kompiliavimas ir kūrimas*“ – vykdomas projekto išėities kodo kompiliavimas, vykdomųjų modulių ir bibliotekų kūrimas, sistemos paruošimas darbui, trumpas testavimas.

„*Sistemos diegimo vedlio kūrimas*“ – kuriamas sistemos diegimo kliento kompiuteryje planas ir vedlys. Sukuriamas produkto diegimo diskas.

„*Produkto pristatymas užsakovui*“ – rengiama produkto prezentacija, demonstruojamos sistemos funkcijos, pasirašomas darbo įteikimo-priėmimo protokolai, produktas įteikiamas užsakovui.

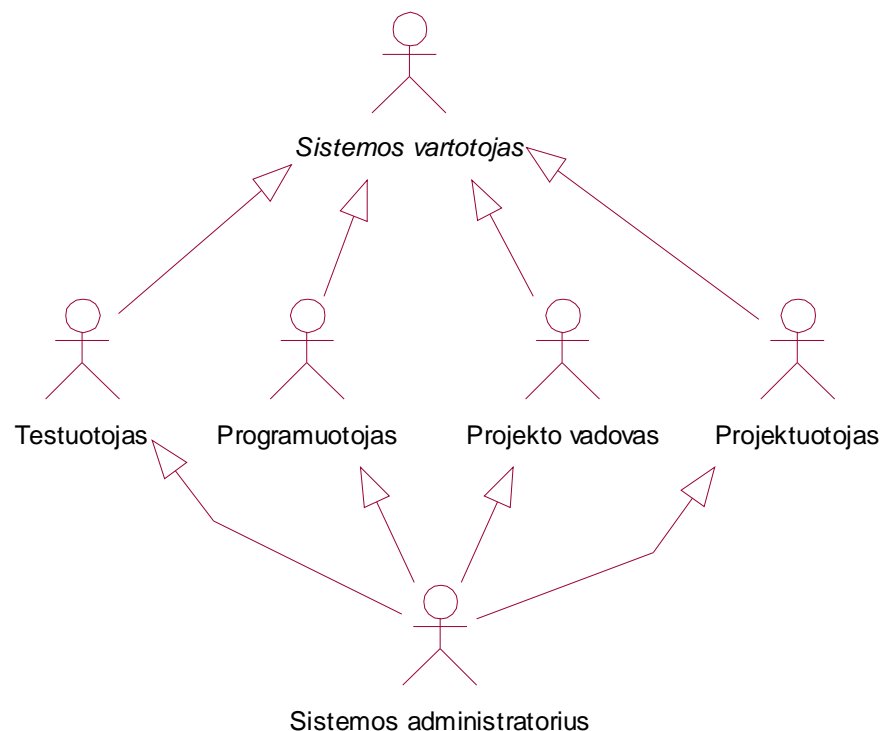
## 4. PROJEKTO DALIS

Projekto dalyje pateikiamas defektų, savybių ir reikalavimų valdymo sistemos realizavimo planas. Šį realizavimo planą sudaro būsimų sistemos vartotojų analizė, panaudojimo atvejai, specifikacijos, duomenų bazės, sistemos architektūros modeliai.

### 4.1 Sistemai keliamų funkcinių ir nefunkcinių reikalavimų modelis

#### 4.1.1 Defektų, savybių ir reikalavimų sistemos vartotojų panaudos atvejų diagramos

- *Sistemos vartotojai (dalyviai):*

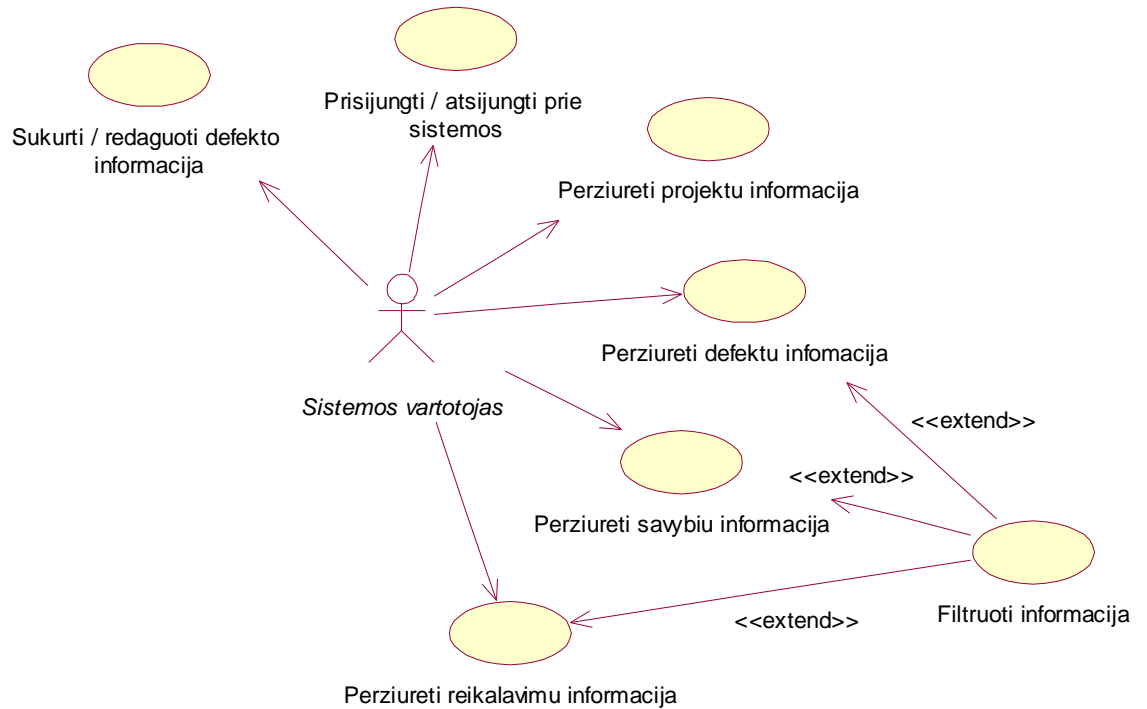


#### 4.1 pav. Informacinės sistemos vartotojų grupės

4.1 paveiksle pateiktoje diagramoje pavaizduotos sistemos vartotojų grupės (rolės). Iš viso jų yra penkios – *Administratorius* ir *Testuotojas*, *Programuotojas*, *Projekto vadovas* ir *Projektuotojas*. Juos galime apibendrinti abstrakčiu dalyviu: *Sistemos vartotojas*. Administratorius visų sistemos vartotojų teises. Papildomai jis gali sukurti kitus sistemos vartotojus, pakeisti jų slaptažodžius bei kitą asmeninę informaciją. Visus veiksmus, kuriuos gali atlikti sistemos vartotojai, pavaizduosime panaudos atvejų diagramose.

- **Sistemos vartotojas:**

Šioje panaudos atvejų diagramoje (4.2 pav.) pavaizduoti visi *sistemos vartotojo* veiklos panaudos atvejai.

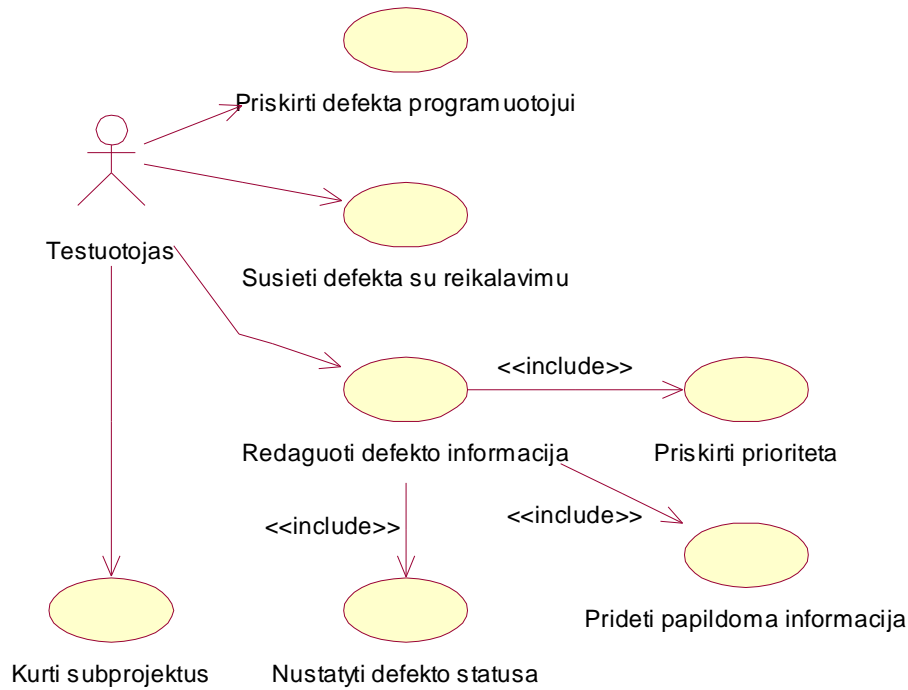


**4.2 pav. Sistemos vartotojo panaudos atvejų diagrama**

*Sistemos vartotojas* tai abstraktus dalyvis, kurio veiksmus paveldi visi kiti sistemos vartotojai. Šis vartotojas gali atlikti visus informacijos peržiūrėjimo veiksmus, panaudojant įvairius filtrus. Taip pat gali sukurti ir redaguoti (raportuoti) defektus.

- **Testuotojas:**

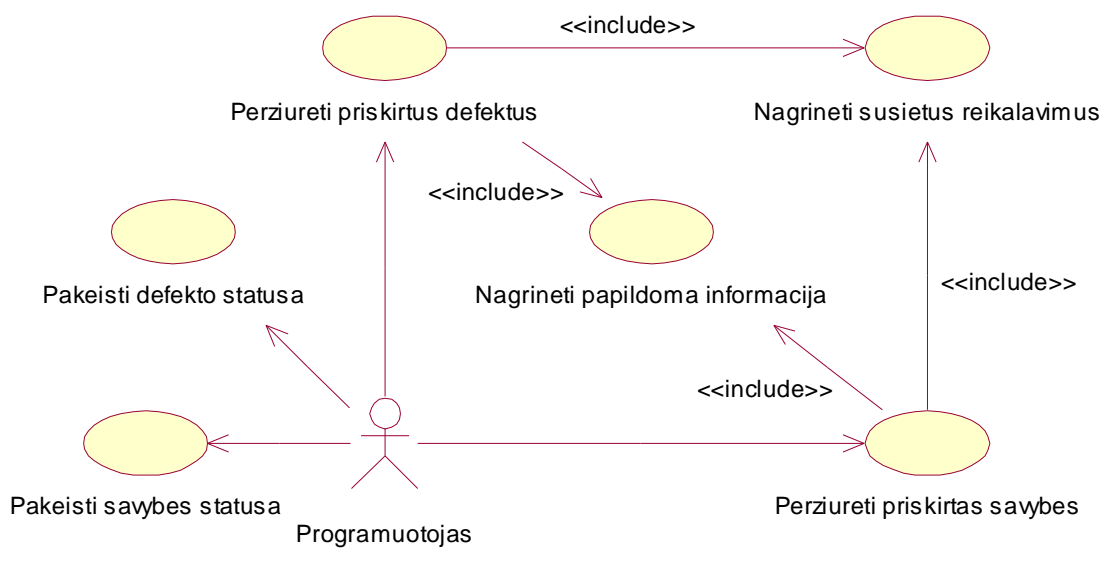
Diagramoje pateiktoje 4.3 paveiksle pavaizduoti testuotojo panaudos atvejai. Pagrindiniai testuotojo veiksmai yra surastų defektų raportavimas projekte ir jų priskyrimas atsakingiems programuotojams. Taip pat numatyta galimybė kiekvienam defektui suteikti prioritetą, nurodyti jo esamą būseną bei prisegti papildomą informaciją.



**4.3 pav. Testuotojo panaudos atvejų diagrama**

- **Programuotojas:**

4.4 paveiksle pateiktoje diagramoje pavaizduoti programuotojo panaudos atvejai. Programuotojas gali peržiūrėti bei redaguoti jam testuotojo priskirtus defektus ir savybes, analizuoti su jais pateiktą papildomą informaciją, susietus reikalavimus.

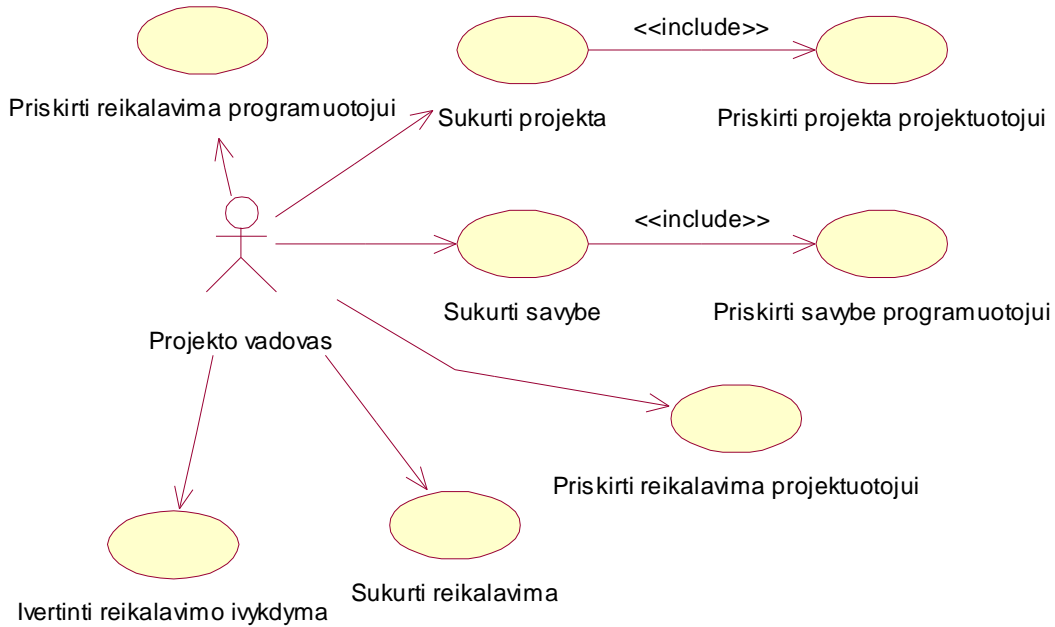


**4.4 pav. Programuotojo panaudos atvejų diagrama**

- **Projekto vadovas:**

Projekto vadovo pagrindinės funkcijos yra naujų projektų kūrimas, pradinių reikalavimų įvedimas bei jų paskirstymas už juos atsakingiems darbuotojams. Projektai ir reikalavimai yra priskiriami projektuotojams, kurie atlieka jų modeliavimą, sukuria kitą programuotojams svarbią informaciją.

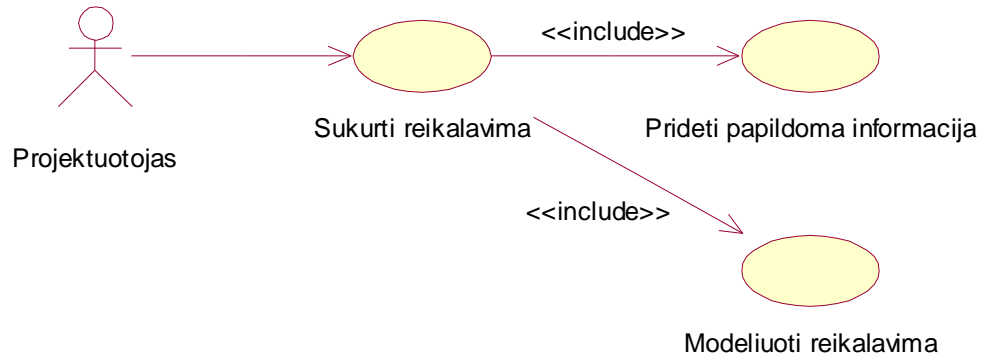
Projekto vadovo panaudos atvejų diagrama pateikta šioje diagramoje (4.5 pav.):



**4.5 pav. Projekto vadovo panaudos atvejų diagrama**

- **Projektotojas:**

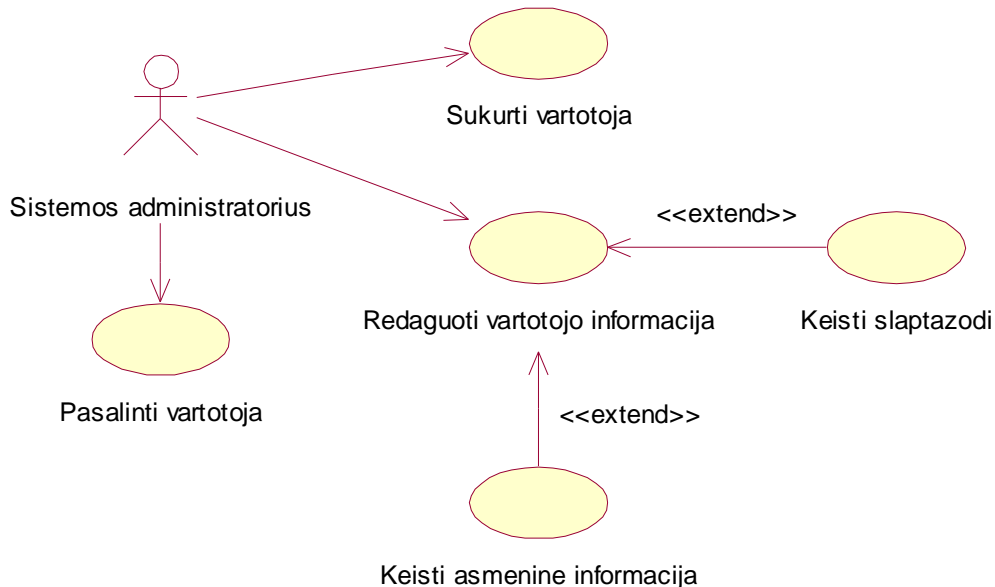
Projektotojas yra mažiausiai funkcijų padengiantis informacinės sistemos vartotojas. Jis yra atsakingas už jam priskirto projekto modelių kūrimą ir pateikimą. Projektotojo panaudos atvejai pavaizduoti 4.6 paveiksle pateiktoje panaudos atvejų diagramoje.



**4.6 pav. Projektotojo panaudos atvejų diagrama**

- **Sistemos administratorius:**

Sistemos administratorius papildomai gali atlikti vartotojų administravimo funkcijas. Šios funkcijos pavaizduotos 4.7 paveiksle pateiktoje panaudos atvejų diagramoje.



**4.7 pav. Sistemos administratoriaus panaudos atvejų diagrama**

## 4.1.2 Panaudojimo atvejų specifikacijos

<b>Panaudojimo atvejis</b>	<b>Sukurti / Redaguoti defekto informaciją</b>
Numeris	PA01
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą ir defektą
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas sukuria / redaguoja defekto informaciją	1.1 Atidaryti defektų langą 1.2 Pasirinkti projektą iš projektų medžio 1.3 Pasirinkti defektą ir atidaryti jo sukūrimo / redagavimo langą
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	1.2. Nėra nei vieno sukurto projekto
Vykdyto variantai	Vartotojas pradeda naudotis IS
Veiklos taisyklės	Vartotojas turi teisingai naudotis IS
Specialūs reikalavimai (nefunkciniai)	Defekto sukūrimas / redagavimas turi būti atliekamas atskirame lange.
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 03

<b>Panaudojimo atvejis</b>	<b>Peržiūrėti projektų informaciją</b>
Numeris	PA02
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti prisijungęs prie IS
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas peržiūri projektų medį	1.1 Vartotojas atidaro defektų, savybių arba reikalavimų langą 1.2 Vartotojas išskleidžia / suskleidžia projektus (medžio šakas)
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	1.2. Projektų nėra arba projektas neturi subprojektų
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai	1. Projekto informacijos redagavimo langus turi būti galima iškviešti iš kontekstinio meniu (paspaudus dešinį pelės klavišą); 2. Medžio lapai (projektai) turi turėti ikonas
Pastabos	Vienas projektas atitinka vieną medžio lapą
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 03



<b>Panaudojimo atvejis</b>	<b>Peržiūrėti defektų informacija</b>
Numeris	PA03
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti prisijungęs prie IS
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas peržiūri projekto defektų informaciją	1.1 Vartotojas atidaro defektų langą 1.2 Vartotojas projektą medyje pasirenka projektą 1.3 Sistema pateikia projekto defektų sąrašą 1.4 Vartotojas peržiūri projekto defektus
Alternatyvos (nesėkmės atvejai)	1.2. Projektų nėra 1.4 Projektui nėra įvesta nei vieno defekto
Vykdyto variantai	1. Vartotojas pasirenka defektą iš pilno defektų sąrašo 2. Vartotojas defektų sąrašė atlieka filtravimą ir pasirenka defektą iš rezultatų
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Defektai turi būti pateikti gražiai atrodančioje lentelėje
Pastabos	Defektų lentelės apačioje turėtų būti pateikta defekto svarbiausios informacijos santrauka
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Peržiūrėti savybių informacija</b>
Numeris	PA04
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti prisijungęs prie IS
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas peržiūri projekto savybių informaciją	1.1 Vartotojas atidaro savybių langą 1.2 Vartotojas projektą medyje pasirenka projektą 1.3 Sistema pateikia projekto savybių sąrašą 1.4 Vartotojas peržiūri projekto savybes
Alternatyvos (nesėkmės atvejai)	1.2. Projektų nėra 1.4 Projektui nėra įvesta nei vienos savybės
Vykdyto variantai	1. Vartotojas pasirenka savybę iš pilno savybių sąrašo 2. Vartotojas savybių sąrašė atlieka filtravimą ir pasirenka defektą iš rezultatų
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Savybės turi būti pateiktos gražiai atrodančioje lentelėje
Pastabos	Savybių lentelės apačioje turėtų būti pateikta savybės svarbiausios informacijos santrauka
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Peržiūrėti reikalavimų informaciją</b>
Numeris	PA05
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti prisijungęs prie IS
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas peržiūri projekto reikalavimų informaciją	1.1 Vartotojas atidaro reikalavimų langą 1.2 Vartotojas projektą medyje pasirenka projektą 1.3 Sistema pateikia projekto reikalavimų sąrašą 1.4 Vartotojas peržiūri projekto reikalavimus
Alternatyvos (nesėkmės atvejai)	1.2. Projektų nėra 1.4 Projektui nėra įvesta nei vieno reikalavimo
Vykdyto variantai	1. Vartotojas pasirenka reikalavimą iš pilno savybių sąrašo 2. Vartotojas reikalavimų sąrašė atlieka filtravimą ir pasirenka reikalavimą iš rezultatų
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Reikalavimai turi būti pateikti gražiai atrodančioje lentelėje
Pastabos	Reikalavimų lentelės apačioje turėtų būti pateikta reikalavimo svarbiausios informacijos santrauka
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Filtruoti informaciją</b>
Numeris	PA06
Dalyvis	Sistemos vartotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti atsidaręs defektų, savybių ar reikalavimų langą ir pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas atlieka informacijos filtravimą	1.1 Įvesti filtravimo kriterijus 1.2 Atlikti filtravimą
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	Filtravimą turi būti galima atlikti dviem būdais: 1. <i>Greituoju</i> (pasirenkant vieną filtravimo kriterijų) 2. <i>Išplėstiniu</i> (filtravimo kriterijų kombinacija)
Veiklos taisyklės	Vartotojas turi įvesti teisingus filtravimo kriterijus
Specialūs reikalavimai (nefunkciniai)	Filtrą turi būti galima patogiai įjungti ir išjungti
Pastabos	Filtravimas gali būti atliekamas paslepiant / parodant lentelės eilutes
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Priskirti defektą programuotojui</b>
Numeris	PA07
Dalyvis	Testuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą ir defektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (testuotojas) priskiria defektą programuotojui	1.1 Vartotojas įveda defekto informaciją 1.2 Vartotojas pasirenka programuotoją iš programuotojų sąrašo
Po sąlyga	Uždarius defekto įrašo redagavimo langą, informacija iškart turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	1.2a Duomenų bazėje nėra vartotojų turinčių programuotojo teises
Vykdyto variantai	1.1a Programuotojui priskiriamas naujas defektas 1.1b Programuotojui priskiriamas egzistuojantis defektas
Veiklos taisyklės	Defektas gali būti priskirtas tik vartotojui turinčiam programuotojo teises
Specialūs reikalavimai (nefunkciniai)	Turi būti pateikta tokia programuotojo informacija: <i>Vardas Pavardė</i>
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Susieti defektą su reikalavimu</b>
Numeris	PA08
Dalyvis	Testuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą ir defektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (testuotojas) priskiria reikalavimą defektui	1.1 Vartotojas įveda / redaguoja defekto informaciją 1.2 Vartotojas pasirenka reikalavimą iš reikalavimų medžio
Po sąlyga	Uždarius defekto įrašo redagavimo langą, informacija iškart turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	1.2a Duomenų bazėje nėra projekto reikalavimų
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Reikalavimų medis turėtų būti „įmontuotas“ į „ <i>combobox</i> “ tipo komponentą
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Kurti subprojektus</b>
Numeris	PA09
Dalyvis	Testuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (testuotojas) sukuria subprojektą	1.1 Vartotojas pasirenka projektą 1.2 Vartotojas įveda subprojekto informaciją
Po sąlyga	Uždarius subprojekto įrašo redagavimo langą, informacija iškart turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	1.1a Nėra sukurtų projektų
Vykdyimo variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Subprojektui galima sukurti neribotą subprojektų kiekį
Neišspręstos problemos	Ar projektai ir subprojektai turėtų būti vaizduojami skirtingai (skirtingomis ikonomis) projektų medyje?
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Redaguoti defekto informaciją (a. priskirti prioritetą; b. Pridėti papildomą informaciją; c. Nustatyti defekto statusą)</b>
Numeris	PA10
Dalyvis	Testuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs defektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (testuotojas) redaguoja defekto informaciją	1.1 Vartotojas pasirenka defektą 1.2 Vartotojas atidaro defekto redagavimo langą 1.3 Vartotojas keičia defekto informaciją
Po sąlyga	Uždarius subprojekto įrašo redagavimo langą, informacija iškart turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	1.1a Duomenų bazėje „defektų“ nėra
Vykdyimo variantai	1.3a Vartotojas prisega papildomą informaciją (failus) 1.3b Vartotojas priskiria prioritetą defektui 1.3c Vartotojas priskiria statusą (priskirtas, užšaldytas, t.t.) 1.3d Vartotojas keičia kitą defekto informaciją
Veiklos taisyklės	
Pastabos	Subprojektui galima sukurti neribotą subprojektų kiekį
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 05

<b>Panaudojimo atvejis</b>	<b>Peržiūrėti priskirtus defektus / savybes</b>
Numeris	PA11
Dalyvis	Programuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas peržiūri jam priskirtus defektus / savybes	1.1 Vartotojas įjungia jam priskirtų defektų / savybių filtrą 1.2 Vartotojas nagrinėja jam priskirtų defektų / savybių informaciją
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	1.2 Vartotojui priskirtų defektų / savybių nėra
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	
Neišspręstos problemos	Ar turėtų būti atskiras mygtukas vartotojo defektų / savybių filtrui įjungti, ar tai būtų realizuojama pačiam vartotojui susidarius filtrą pagal jo prisijungimo vardą?
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Pakeisti defekto / savybės statusą</b>
Numeris	PA12
Dalyvis	Programuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas pakeičia pasirinkto defekto / savybės statusą	1.1 Vartotojas pasirenka projektą 1.2 Vartotojas pasirenka ir atsidaro savybės sukūrimo / redagavimo langą 1.3 Vartotojas pakeičia defekto / savybės statusą
Po sąlyga	Informacija turi būti įrašyta į duomenų bazę uždarius defekto / savybės redagavimo langą
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Savybės būsenos (statuso įrašai) turi būti saugomi duomenų bazėje
Pastabos	Programuotojo vartotojui galėtų būti sudaryta aibė būsenų kuriomis jis galėtų manipuluoti
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Nagrinėti papildomą informaciją</b>
Numeris	PA13
Dalyvis	Programuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą / savybę / defektą
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas nagrinėja papildomą informaciją	1.1 Vartotojas pasirenka projektą / savybę / defektą 1.2 Vartotojas peržiūri papildomą informaciją
Po sąlyga	Papildomą medžiagą (failą) turi atpažinti operacinė sistema
Alternatyvos (nesėkmės atvejai)	1.2 Papildomos informacijos nėra
Vykdyto variantai	1.2a Vartotojas atidaro papildomą informaciją iš peržiūros lango ekrano apačioje 1.2b Vartotojas atidaro papildomą informaciją iš informacijos redagavimo lango
Veiklos taisyklės	Vartotojo kompiuteryje turi būti įdiegta programinė įranga galinti atidaryti papildomos informacijos failus
Specialūs reikalavimai (nefunkciniai)	
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Nagrinėti susietus reikalavimus</b>
Numeris	PA14
Dalyvis	Programuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas nagrinėja defektui ar savybei priskirtus reikalavimus	1.1 Vartotojas pasirenka projektą 1.2 Vartotojas pasirenka defektą ar savybę 1.3 Vartotojas peržiūri su defektu ar savybe susietus reikalavimus
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	1.4 Nėra susietų reikalavimų
Vykdyto variantai	1.3a Vartotojas peržiūri susietus reikalavimus iš peržiūros lango ekrano apačioje 1.3b Vartotojas peržiūri susietus reikalavimus iš informacijos (defekto, savybės) redagavimo lango
Veiklos taisyklės	
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Priskirti reikalavimą programuotojui / projektuotojui</b>
Numeris	PA15
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą ir defektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas priskiria reikalavimą programuotojui / projektuotojui	1.1 Vartotojas įveda reikalavimo informaciją 1.2 Vartotojas pasirenka programuotoją / projektuotoją iš programuotojų / projektuotojų sąrašo
Po sąlyga	Uždarius reikalavimo įrašo redagavimo langą, informacija turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	1.2a Duomenų bazėje nėra vartotojų turinčių programuotojo / projektuotojo teises
Vykdyto variantai	1.1a Programuotojui / Projektuotojui priskiriamas naujas reikalavimas 1.1b Programuotojui / Projektuotojui priskiriamas egzistuojantis reikalavimas
Veiklos taisyklės	Reikalavimas gali būti priskirtas tik vartotojui turinčiam programuotojo arba projektuotojo teises
Specialūs reikalavimai (nefunkciniai)	Pasirinkimo laukeliuose turi būti pateikta tokia programuotojo / projektuotojo informacija: Vardas Pavardė
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Sukurti projektą</b>
Numeris	PA16
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas sukuria projektą	1.1 Pasirenka pagrindinę projektų medžio šaką 1.2 Sukuria naują projektą
Po sąlyga	Uždarius projekto įrašo sukūrimo langą, informacija turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Projektų medis turi būti patogus naudoti: turi būti įrankiai medžio šakos pilnam išskleidimui / suskleidimui
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 09

<b>Panaudojimo atvejis</b>	<b>Priskirti projektą projektuotojui</b>
Numeris	PA17
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi sukurti arba pasirinkti projektą
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas priskiria projektą projektuotojui	1.1 Vartotojas įveda projekto informaciją / atidaro projektą 1.2 Vartotojas pasirenka vartotoją turintį projektuotojo teises
Po sąlyga	Informacija užsaugoma uždarius projekto langą
Alternatyvos (nesėkmės atvejai)	1.2a Vartotojo su projektuotojo teisėmis sąrašė nėra
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Priskyrimas vykdomas projekto sukūrimo lange
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Priskirti savybę programuotojui</b>
Numeris	PA18
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi sukurti arba pasirinkti projektą
Pagrindinis įvykių šrautas	IS reakcija ir sprendimai
Vartotojas priskiria savybę projektuotojui	1.3 Vartotojas įveda projekto informaciją / atidaro projektą 1.4 Vartotojas pasirenka vartotoją turintį projektuotojo teises
Po sąlyga	Informacija užsaugoma uždarius projekto redagavimo langą
Alternatyvos (nesėkmės atvejai)	1.2a Vartotojo su projektuotojo teisėmis sąrašė nėra
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Priskyrimas vykdomas projekto redagavimo lange
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10



<b>Panaudojimo atvejis</b>	<b>Priskirti reikalavimą projektuotojui</b>
Numeris	PA19
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi sukurti arba pasirinkti projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas priskiria reikalavimą projektuotojui	1.1 Vartotojas įveda reikalavimo informaciją / atidaro reikalavimo redagavimo langą 1.2 Vartotojas pasirenka vartotoją turintį projektuotojo teises
Po sąlyga	Informacija užsaugoma uždarius reikalavimo redagavimo langą
Alternatyvos (nesėkmės atvejai)	1.2a Vartotojo su projektuotojo teisėmis sąrašė nėra
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Priskyrimas vykdomas reikalavimo sukūrimo lange
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Įvertinti reikalavimo įvykdymą</b>
Numeris	PA20
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi pasirinkti reikalavimą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas reikalavimo realizaciją	1.1 Vartotojas įveda reikalavimo atidaro reikalavimo redagavimo langą 1.2 Vartotojas nustato reikalavimo atlikimo statusą ir parašo įvertinimą procentais
Po sąlyga	Informacija užsaugoma uždarius reikalavimo redagavimo langą
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	Reikalavimas turi būti realizuotas
Specialūs reikalavimai (nefunkciniai)	
Pastabos	Priskyrimas vykdomas reikalavimo sukūrimo lange
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Sukurti savybę</b>
Numeris	PA21
Dalyvis	Projektų vadovas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas sukuria savybę	1.1 Atidaro naujos savybės sukūrimo langą 1.2 Įveda savybės informaciją, prisega papildomą informaciją
Po sąlyga	Uždarius savybės įrašo sukūrimo langą, informacija turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Savybės sukūrimo langas turi būti panašus į defekto sukūrimo langą
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Sukurti reikalavimą</b>
Numeris	PA22
Dalyvis	Projektų vadovas / Projektuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi būti pasirinkęs projektą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas sukuria reikalavimą	1.1 Atidaro naujo reikalavimo sukūrimo langą 1.2 Įveda reikalavimo informaciją, prisega papildomą informaciją
Po sąlyga	Uždarius reikalavimo įrašo sukūrimo langą, informacija turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	
Pastabos	
Neišspręstos problemos	Reikalavimai galėtų būti grupuojami į biznio srities reikalavimus ir techninius reikalavimus. Taip pat tarp biznio ir techninių reikalavimų turėtų būti galimybė sudaryti „reikalavimų sekamumo matricą“.
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Modeliuoti reikalavimą</b>
Numeris	PA23
Dalyvis	Projektuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti išorinį modeliavimo įrankį
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas modeliuoja reikalavimą	1.1 Analizuoja reikalavimo informaciją ir dokumentus 1.2 Modeliuoja reikalavimą su modeliavimo įrankiu 1.3 Prisega modelio failą prie reikalavimo specifikacijos
Po sąlyga	Uždarius reikalavimo įrašo sukūrimo langą, informacija turi būti įrašyta į duomenų bazę
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	Modeliavimas turėtų būti atliekamas su įrankiu kuriuo sukurtus failus gali atidaryti
Pastabos	
Neišspręstos problemos	Ar reikalavimų modeliavimas turėtų būti palaikomas sistemos? Kol kas tai yra apibrėžiama kaip sistemos funkcionalumo plėtojimo galimybė.
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Pridėti papildomą medžiagą (prie reikalavimo)</b>
Numeris	PA24
Dalyvis	Projektuotojas
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti papildomos informacijos arba būti atlikęs reikalavimo modeliavimą
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas prisega papildomą medžiagą (reikalavimo modelį) prie reikalavimo	1.1 Projektuotojas atlieka reikalavimo modeliavimą 1.2 Vartotojas atidaro reikalavimo redagavimo langą 1.3 Vartotojas prisega papildomą informaciją prie reikalavimo
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Papildoma informacija turėtų būti sukurta su plačiai naudojamu įrankiu (organizacijoje), kad tą informaciją visi galėtų peržiūrėti
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Sukurti vartotoją</b>
Numeris	PA25
Dalyvis	Sistemos administratorius
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti administratoriaus teises
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (administratorius) sukuria naują sistemos vartotoją	1.1 Vartotojas atidaro sistemos vartotojų valdymo langą 1.2 Vartotojas atidaro vartotojo sukūrimo / redagavimo langą 1.3 Vartotojas įveda vartotojo asmeninę ir kitą informaciją (priskiria teises ir t.t.)
Po sąlyga	Vartotojas turi būti įrašytas į duomenų bazę uždarius vartotojo sukūrimo / redagavimo langą
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	Sukuriant vartotoją būtinai turi būti nurodytas jo prisijungimo vardas, slaptažodis ir priskirta bent viena rolė
Specialūs reikalavimai (nefunkciniai)	Vartotojo slaptažodis duomenų bazėje turi būti išsaugotas užšifruotame ir neatkoduojamame pavidale
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

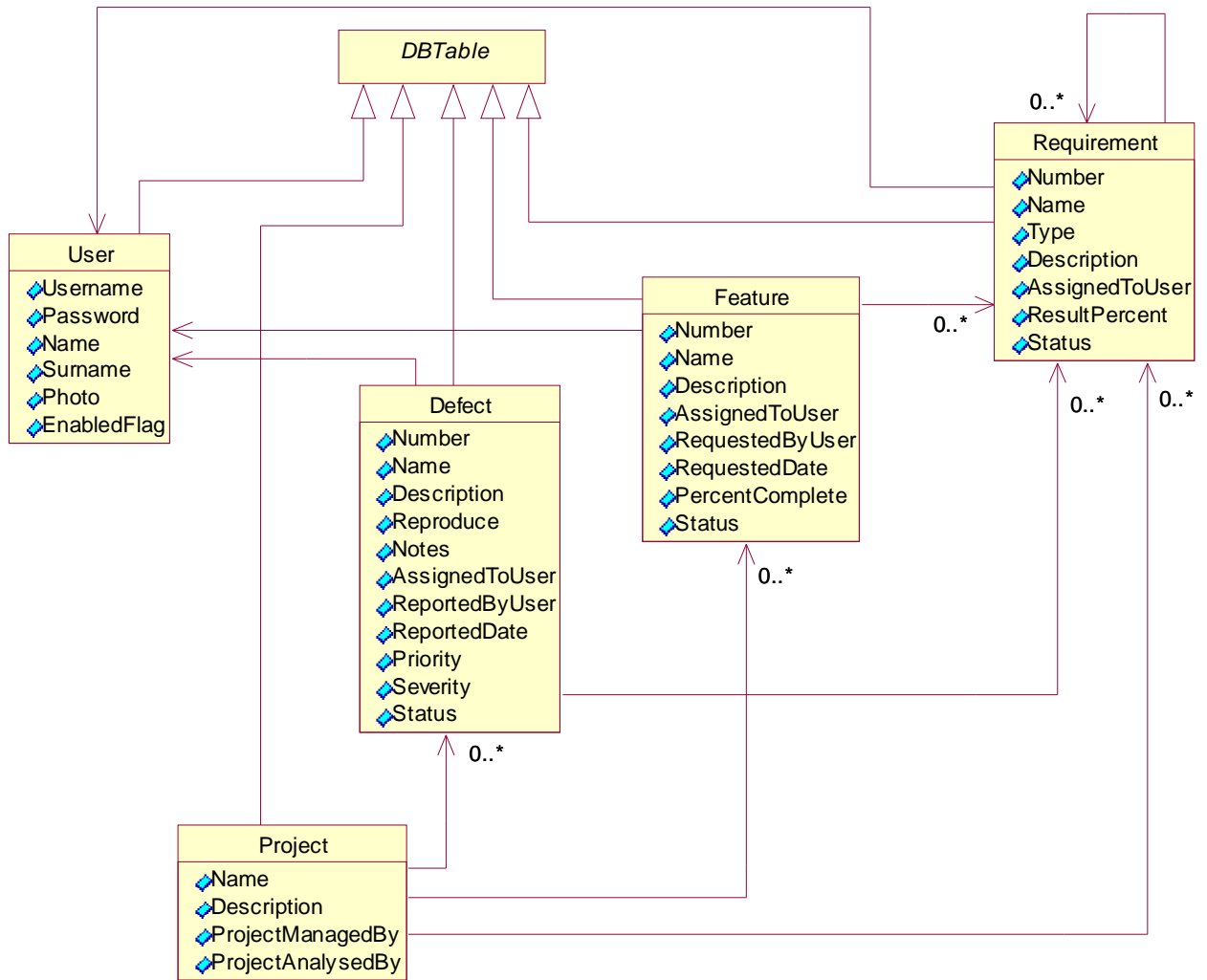
<b>Panaudojimo atvejis</b>	<b>Pašalinti vartotoją</b>
Numeris	PA26
Dalyvis	Sistemos administratorius
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti administratoriaus teises
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (administratorius) pašalina sistemos vartotoją	1.1 Vartotojas pasirenka pašalinamąjį sistemos vartotoją iš vartotojų sąrašo 1.2 Vartotojas pašalina vartotoją (patį save)
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	1.2a Šiuo metu prisijungęs vartotojas (administratorius) bando pašalinti savo vartotoją 1.2b Kitas administratorius pirmas pašalino pasirinktąjį vartotoją (vartotojas jau neegzistuoja)
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	Vartotojo slaptažodis duomenų bazėje turi būti išsaugotas užšifruotame ir neatkoduojamame pavidale
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Redaguoti vartotojo informaciją / Keisti asmeninę vartotojo informaciją</b>
Numeris	PA27
Dalyvis	Sistemos administratorius
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti administratoriaus teises
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (administratorius) redaguoja pasirinkto sistemos vartotojo informaciją	1.1 Vartotojas pasirenka sistemos vartotoją iš vartotojų sąrašo 1.2 Vartotojas atidaro vartotojo redagavimo langą 1.3 Vartotojas keičia informaciją
Po sąlyga	Pakeista informacija į duomenų bazę įrašoma uždarius vartotojo redagavimo langą
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	
Veiklos taisyklės	
Specialūs reikalavimai	Turi būti saugoma vartotojo sukūrimo / redagavimo data
Pastabos	
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

<b>Panaudojimo atvejis</b>	<b>Keisti vartotojo slaptažodį</b>
Numeris	PA28
Dalyvis	Sistemos administratorius
Sistema	Defektų, savybių ir reikalavimų valdymo IS
Prieš sąlyga	Vartotojas turi turėti administratoriaus teises
Pagrindinis įvykių srautas	IS reakcija ir sprendimai
Vartotojas (administratorius) keičia pasirinkto sistemos vartotojo slaptažodį	1.1 Vartotojas pasirenka sistemos vartotoją iš vartotojų sąrašo 1.2 Vartotojas keičia vartotojo slaptažodį
Po sąlyga	Senasis vartotojo slaptažodis pakeičiamas nauju
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	1.2a Administratorius keičia vartotojo slaptažodį iškviėdamas slaptažodžio keitimo langą 1.2b Administratorius keičia vartotojo slaptažodį iš vartotojo redagavimo lango
Veiklos taisyklės	Administratorius neturi sužinoti senojo vartotojo slaptažodžio
Specialūs reikalavimai	
Pastabos	Slaptažodis galėtų būti užkoduojamas MD5 algoritmu
Neišspręstos problemos	
Sudarė	P. Indriūnas
Sudarymo data	2005 05 10

### 4.1.3 Dalykinės srities klasių diagrama

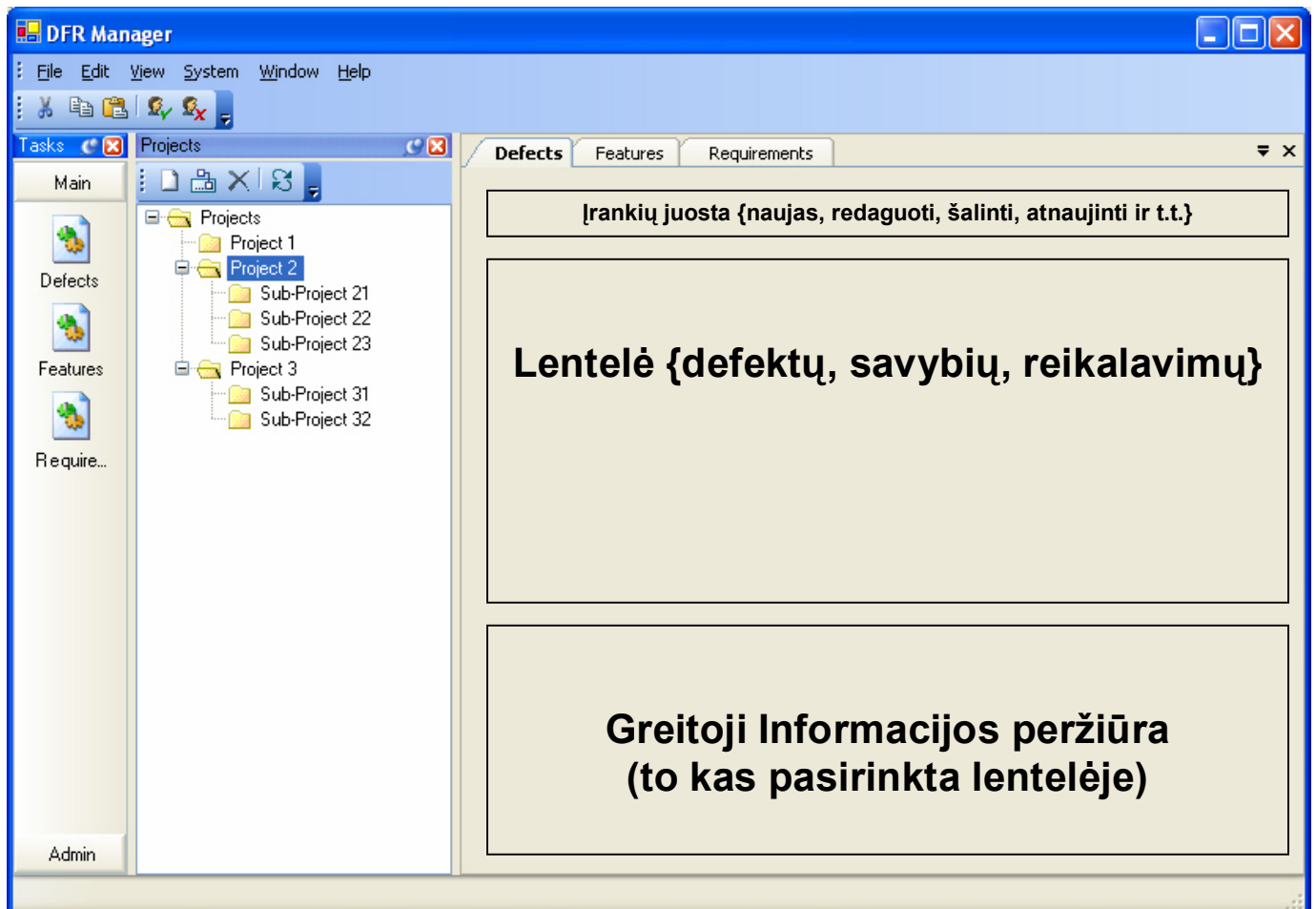
Diagramoje pateiktoje 4.8 paveiksle, pavaizduotos nagrinėjamosios dalykinės srities klasės. Klasės atvaizduoja pagrindinius kompiuterizuojamus dalykinės srities objektus. Visos klasės paveldi abstrakčią duomenų bazės esybės klasę *DBTable*. Kai kurie ryšiai tarp klasių yra pavaizduoti su kardinalumu *nulis- arba-daug*. Tai reiškia, kad klasė gali turėti daug susijusių kitos klasės objektų. Ši dalykinės srities klasių diagrama, tai yra abstraktus žvilgsnis į sistemos loginę architektūrą.



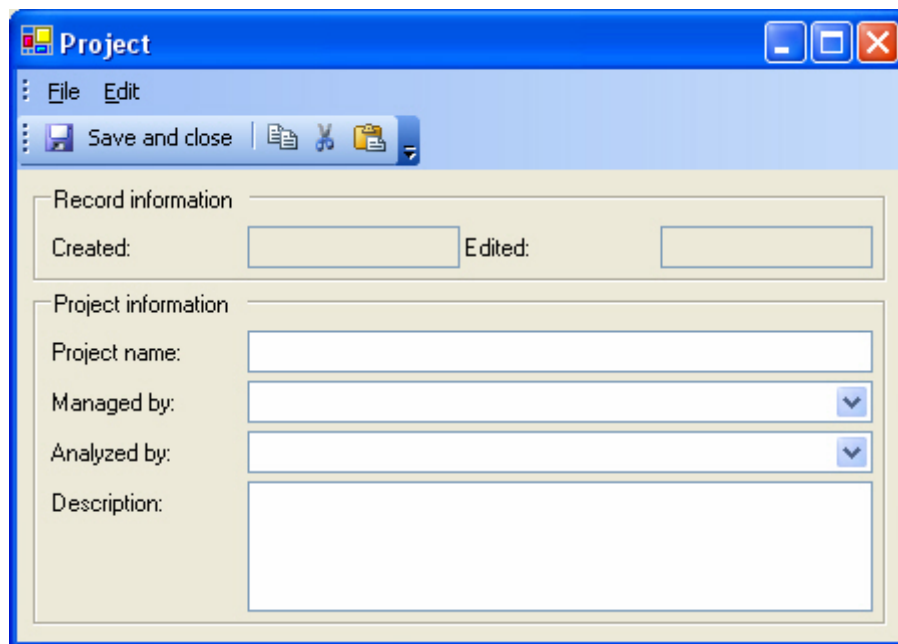
4.8 pav. Defektų, savybių ir reikalavimų dalykinės srities klasės

#### 4.1.4 Vartotojų interfeiso modelis

Vartotojo interfeiso modelyje (4.9; 4.10; 4.11; 4.12 pav.) pateikiama sistemos vartotojo sąsajos architektūra. Vartotojo sąsajos modeliui sudaryti, buvo pasirinktas prototipo kūrimo būdas. Šis būdas buvo pasirinktas todėl, kad programinės įrangos kūrimo įrankiu (*MS Visual Studio .net*) pasinaudojant papildomais trečiųjų šalių komponentais (*Infragistics*) yra patogiau kurti langus ir sudėtingos struktūros vartotojo sąsajos komponentus. Be to, juos bus galima panaudoti kuriant galutinį sistemos vartotojo sąsajos variantą.



4.9 pav. Pagrindinė programos forma



**Project**

File Edit

Save and close

Record information

Created: Edited:

Project information

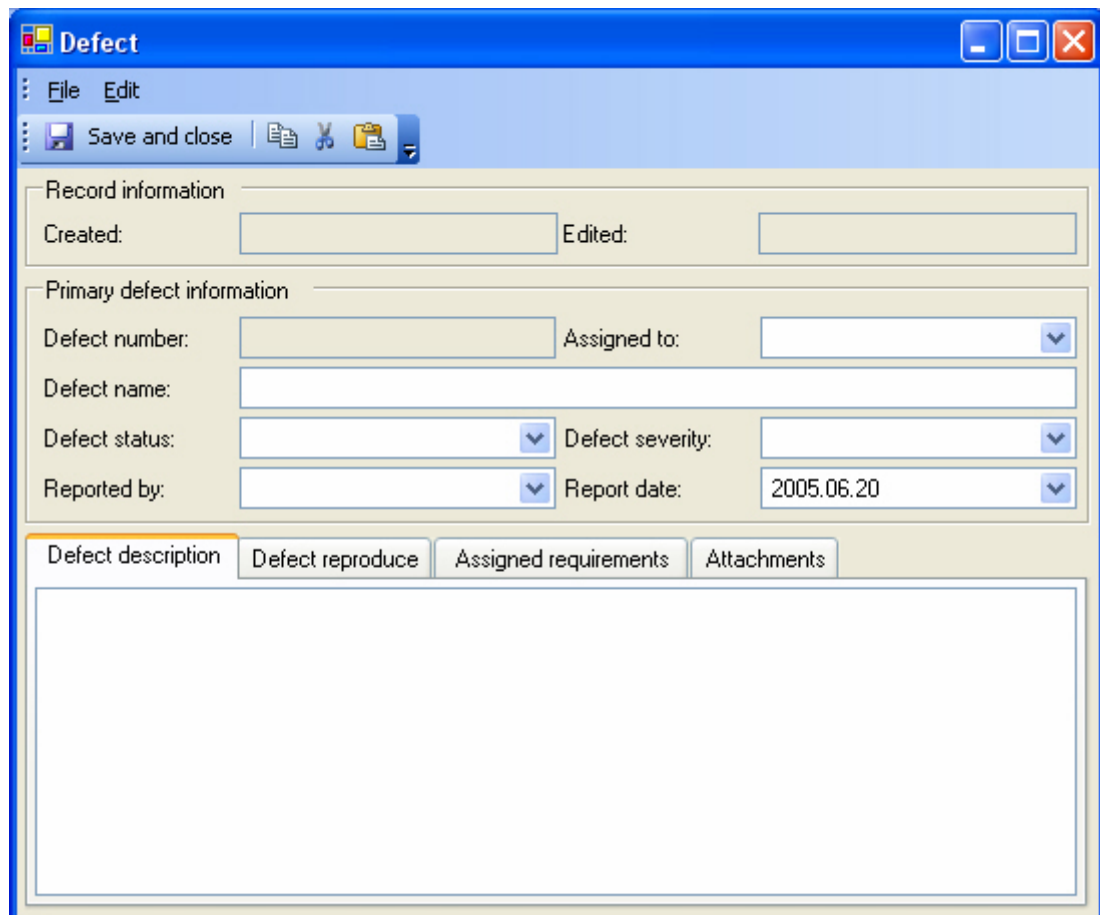
Project name:

Managed by:

Analyzed by:

Description:

4.10 pav. Projekto sukūrimo / redagavimo forma



**Defect**

File Edit

Save and close

Record information

Created: Edited:

Primary defect information

Defect number: Assigned to:

Defect name:

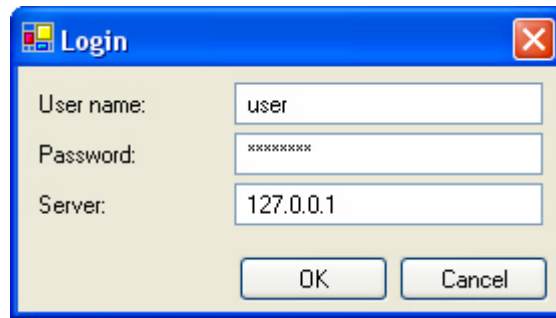
Defect status: Defect severity:

Reported by: Report date: 2005.06.20

Defect description Defect reproduce Assigned requirements Attachments

4.11 pav. Defekto sukūrimo / redagavimo forma





4.12 pav. Prisijungimo prie sistemos langas

#### 4.1.5 Nefunkciniai reikalavimai

Kuriama defektų savybių ir reikalavimų sistema bus naudojama programinę įrangą kuriančioje organizacijoje. Tokio tipo įmonėse dirba įvairūs informacinių technologijų profesionalai, kurie gerai supranta kokios yra geros programinės įrangos savybės. Siekiant kad programa būtų plačiai naudojama, o ne atmesta, ji turi būti patraukli ir patogi naudoti.

Svarbiausi nefunkciniai reikalavimai, keliami kuriamai sistemai, pateikti šioje lentelėje:

Nr.	Nefunkcinis reikalavimas	Reikalavimo paaiškinimas
1	Programos grafinė vartotojo sąsaja turi būti preciziškai tiksli ir taisyklinga	Vartotojo sąsajos išvaizda ir patrauklumas yra vienas iš pačių svarbiausių kriterijų, vartotojui sprendžiant apie programos kokybę. Visi langų komponentai turi būti taisyklingai ir vienodai išlygiuoti nuo kraštų paliekant 6-8 pikselius; Spalvos turi būti suderintos ir ne per daug ryškios
2	Programos grafinės vartotojo sąsajos langai turi būti suprojektuoti pagal Microsoft Outlook 2003 stilių	Informacijos sukūrimo ir redagavimo dialogai turi būti sudaryti pagal Microsoft Outlook stilių. Tai reiškia, kad dialoguose vietoj tradicinių OK ir Cancel mygtukų, naudojami mygtukai <i>Save</i> ; <i>Save And Close</i> ; <i>Close</i> ir pan.
3	Programa turi būti galima nesunkiai dirbti, nesinaudojant pele.	Visi langų komponentai turi turėti teisingą TAB klavišo paspaudimo sekos numerį. Vartotojas naudodamasis TAB klavišu, turi turėti galimybę patekti į visus vartotojo sąsajos komponentus aktyviame lange.
4	Sistema turi stabiliai veikti dirbant keliems vartotojams vienu metu	Dirbant su sistema, neturi kilti problemų dėl tuo pačiu metu prisijungusių prie sistemos vartotojų skaičiaus. Sistema turi išlikti stabili net jeigu 2 arba daugiau vartotojų redaguoja tą patį objektą vienu metu.
5	Programa turi veikti stabiliai, gaunant didelius duomenų kiekius iš duomenų bazės	Esant prastam ryšiui, gaunant didelį duomenų kiekį programa neturi „užšalti“. Tam tikslui pasiekti reikia naudoti asinchroninius kreipinius.
6	Programa turi veikti stabiliai dirbant iki 16 valandų	Programa, paleista vieną kartą, turi veikti efektyviai ir stabiliai dirbant visą darbo dieną (t.y. „neužsiukšlinti“ kompiuterio atminties, nenaudoti didesnio kompiuterio resursų kiekio nei yra būtina)
7	Programa turi būti pritaikyta darbui Microsoft Windows aplinkoje	Programa turi veikti Microsoft Windows (pradedant 2000) operacinėje sistemoje. Taip pat turi naudoti Microsoft .net technologijas.
8	Klaidos turi būti greitai surandamos	Sistemos klaidos turi būti surandamos ir greitai ištaisomos. Tam tikslui pasiekti programa turi būti sudaryta iš griežtai specifines funkcijas atliekančių komponentų.

4.1 lentelė. Nefunkcinių reikalavimų lentelė

## 4.2 Projekto modelis

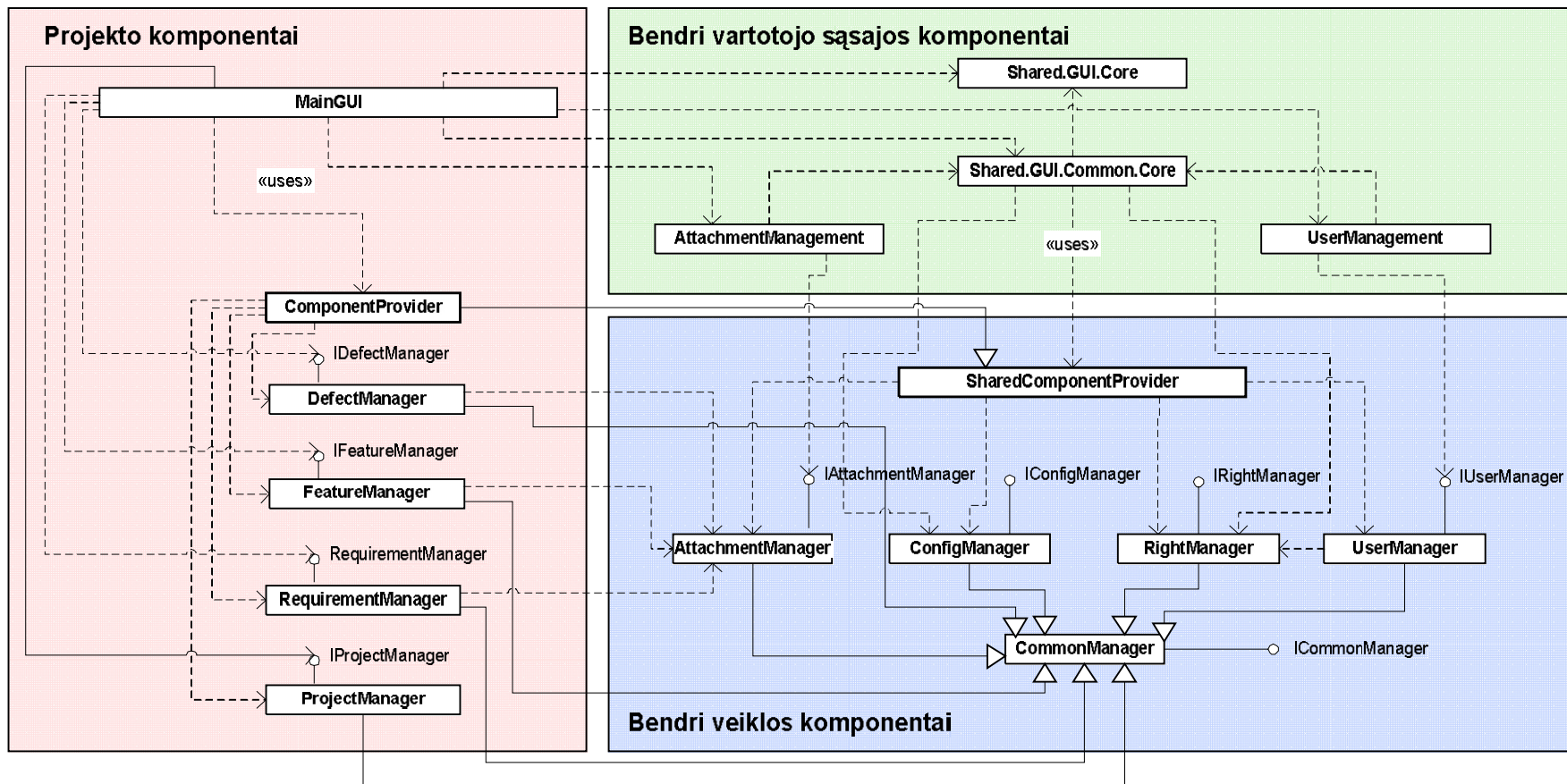
### 4.2.1 Projekto tikslas

Šios informacinės sistemos kūrimo projekto tikslas yra nuodugniai išnagrinėti šios sistemos vartotojų grupes, panaudojimo atvejus, jų realizacijas, sudaryti loginę programos architektūrą, atlikti duomenų bazės projektavimą, apmastyti kokios galėtų būti realizacijos alternatyvos. Taip pat suprojektuoti sistemos vartotojo interfeisą, sudaryti langų navigavimo planą.

### 4.2.2 Sistemos architektūra

4.13 paveiksle pateiktoje diagramoje pavaizduota sistemos statinė architektūra. Diagramoje pavaizduotos 3 pagrindiniai programos architektūros sluoksniai: *pakartotinai panaudotini veiklos komponentai*, *pakartotinai panaudotini vartotojo sąsajos komponentai* ir *realizuojamos sistemos veiklos ir vartotojo sąsajos komponentai*. Veiklos objektus sukuria taip vadinami komponentų tiekėjai (*Component Providers*), kuriuos naudoja vartotojo sąsajos komponentai per jų interfeisus. Kiekvieną esybių grupę valdo specialiai tai esybių grupei sukurtas komponentas (pavyzdžiui sistemos vartotojus ir roles valdo vartotojų valdymo komponentas - *UserManager*).

Komponentai – tai operacinės sistemos Windows bibliotekos (*dll failai*) ir vykdomieji moduliai (*exe failai*).



4.13 pav. Defektų, savybių ir reikalavimų valdymo sistemos komponentų modelis

- **Pakartotinai panaudotinių veiklos komponentų sluoksnis („Bendri veiklos komponentai“):**

Komponento pavadinimas	Paskirtis
CommonManager (ICommonManager)	Šis komponentas yra paties žemiausio lygio duomenų bazių operacijų komponentas. Jis yra atsakingas už duomenų konteinerių informacijos gavimo, įrašymo, pašalinimo, atnaujinimo operacijas. Šį komponentą paveldi visi kiti veiklos komponentai dirbantys su duomenų baze.
UserManager (IUserManager)	Šis komponentas atsakingas už sistemos vartotojų, rolių valdymą, autentifikacijos ir autorizacijos veiksmus.
RightManager (IRightManager)	Šis komponentas yra susijęs su vartotojų valdymo komponentu. Jis teikia vartotojų teisių atlikti tam tikrus veiksmus, patikrinimo paslaugas.
ConfigManager (IConfigManager)	Konfigūracijos komponentas atsakingas už konfigūracijos lentelėse esančios informacijos (nekintančios) teikimą ir valdymą.
AttachmentManager (IAttachmentManager)	Priedų komponentas atlieka papildomos informacijos saugyklos valdymo operacijas. Priedai gali būti priskirti bet kokioms esybėms per specialiai joms sukurtą ryšio duomenų bazės lentelę.
SharedComponentProvider	Šis komponentais yra atsakingas už šiame sluoksnyje esančių komponentų sukūrimą. Jis atlieka komponentų fabriko vaidmenį. Komponento sukūrimas atliekamas pagal pateikto komponento interfeiso tipą.

**4.2 lentelė. Pakartotinai panaudotini veiklos komponentai**

- **Pakartotinai panaudotinių sąsajos komponentų sluoksnis („Bendri vartotojo sąsajos komponentai“):**

Komponento pavadinimas	Paskirtis
UserManagement	Šiame komponente saugomi su sistemos vartotojų valdymu susiję grafinio vartotojo interfeiso klasės ir komponentai. Šis komponentas tiesiogiai naudoja veiklos vartotojų valdymo komponentą – <i>UserManagement</i> visus veiksmus atlikdamas per šio komponento interfeisą <i>IUserManagement</i> .
AttachmentManagement	Šiame komponente saugomi su priedais susiję grafinio vartotojo interfeiso klasės ir komponentai. Šis komponentas tiesiogiai naudoja veiklos priedų valdymo komponentą – <i>AttachmentManager</i> visus veiksmus atlikdamas per šio komponento interfeisą <i>IAttachmentManager</i> .
Shared.GUI.Common.Core	Tai visuose programinės įrangos projektuose naudotinas grafines vartotojo sąsajos komponentas. Šiame komponente yra klasės veiklos komponentų sukūrimo, vartotojo prisijungimo prie sistemos valdymui, taip pat langai ir panelės dažnai naudojamos daugelyje programinės įrangos projektų.
Shared.GUI.Core	Šis komponentas yra paties žemiausio lygio vartotojo sąsajos valdymo komponentas. Jame saugomos visos žemiausio hierarchijos lygio langų, panelių klasės ir resursai. Taip pat įvairios klasės jokiais ryšiais nesusiję su veiklos srities komponentais. Šį komponentą galima panaudoti bet kuriame programinės įrangos vartotojo sąsajos projekte.

**4.3 lentelė. Pakartotinai panaudotini vartotojo sąsajos komponentai**

- **Projekto komponentų sluoksnis („Projekto komponentai“):**

<b>Komponento pavadinimas</b>	<b>Paskirtis</b>
DefectManager (IDefectManager)	Šios komponentas yra atsakingas už defektų informacijos valdymą. Pagrindiniai šio komponento veiksmai yra defektų informacijos iš duomenų bazės gražinimas lentelių (DataSet) arba esybių masyvo pavidalu, taip pat defektų įrašymo, šalinimo bei informacijos atnaujinimo operacijos.
FeatureManager (IFeatureManager)	Šios komponentas yra atsakingas už savybių informacijos valdymą. Pagrindiniai šio komponento veiksmai yra savybių informacijos iš duomenų bazės gražinimas lentelių (DataSet) arba esybių masyvo pavidalu, taip pat savybių įrašymo, šalinimo bei informacijos atnaujinimo operacijos.
RequirementManager (IRequirementManager)	Šios komponentas yra atsakingas už reikalavimų informacijos valdymą. Pagrindiniai šio komponento veiksmai yra reikalavimų informacijos iš duomenų bazės gražinimas lentelių (DataSet) arba esybių masyvo pavidalu, taip pat reikalavimų įrašymo, šalinimo informacijos atnaujinimo, reikalavimų modelių valdymo operacijos.
ProjectManager (IProjectManager)	Projektų valdymo komponentas atsakingas už projekto esybių valdymą ir projektų medžio suformavimą.
ComponentProvider	Šis komponentais yra atsakingas už projekto sluoksnyje esančių komponentų sukūrimą. Jis atlieka komponentų fabriko vaidmenį. Komponento sukūrimas atliekamas pagal pateikto komponento interfeiso tipą.
MainGUI	Tai yra pagrindinis vykdomasis programos vartotojo sąsajos komponentas. Jame saugomi projekto grafinio vartotojo interfeiso klasės ir komponentai. Taip pat valdoma programos darbo logika. Šis komponentas gali naudoti bet kurį egzistuojantį komponentą, kurio tik gali prireikti.

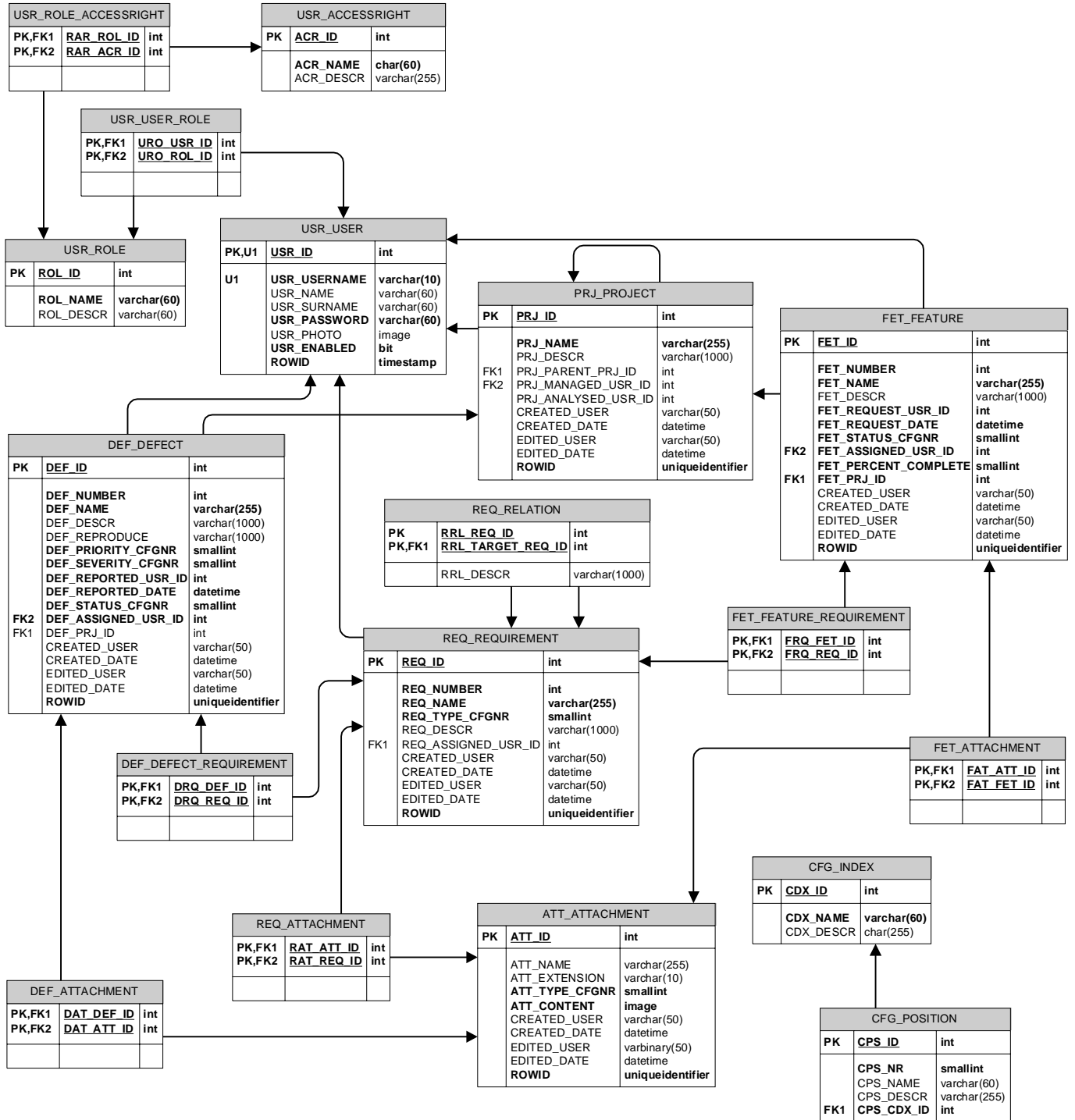
#### 4.4 lentelė. Projekto komponentai

Pagal šį sistemos architektūros modelį (4.13 pav.) galima kurti atskirus komponentus, kurie yra pakankamai mažai susiję tarpusavyje. Sudarius komponentų interfeisus, komponentų realizavimą labai lengvai būtų galima paskirstyti keliems programuotojams, o atsiradus klaidoms būtų labai paprasta jas surasti.

Sistemos skaidymas į komponentus yra labai svarbus, siekiant sukurti patikimą, ir lengvai plečiamą produktą. Komponentų bendravimas per interfeisus pagerina komponentų versijavimo galimybes, taip pat sukuriama griežtesnės ribos tarp komponentų funkcijų.

### 4.2.3 Duomenų bazės modelis

4.14 paveiksle pateikta defektų, savybių ir reikalavimų valdymo sistemos duomenų bazės schema.



4.14 pav. Defektų, savybių ir reikalavimų valdymo sistemos duomenų bazės modelis

### Vartotojų lentelių grupė:

Vartotojų lentelių grupė yra naudojama sistemos vartotojų, rolių ir teisių saugojimui.

- **Lentelė „USR\_USER“:**

Šioje lentelėje saugoma sistemos vartotojų informacija. Vartotojai saugomi šioje lentelėje turi teisę jungtis ir dirbti su informacine sistema. Būtina vartotojo informacija yra jo prisijungimo vardas, slaptažodis ir aktyvumo vėlevėlė. Vardas ir pavardė nėra būtina informacija. Vartotojus sukuria vartotojas turintis informacinės sistemos administratoriaus teises.

USR_USER		
<b>PK,U1</b>	<b>USR_ID</b>	int
<b>U1</b>	<b>USR_USERNAME</b> USR_NAME USR_SURNAME <b>USR_PASSWORD</b> USR_PHOTO <b>USR_ENABLED</b> <b>ROWID</b>	varchar(10) varchar(60) varchar(60) <b>varchar(60)</b> image bit timestamp

<i>Atributas</i>	<i>Atributo paskirtis</i>
USR_ID	Vartotojo identifikacijos numeris
USR_USERNAME	Sistemos vartotojo vardas.
USR_NAME	Sistemos vartotojo asmens vardas
USR_SURNAME	Sistemos vartotojo asmens pavardė
USR_PASSWORD	Slaptažodis
USR_PHOTO	Sistemos vartotojo asmens nuotrauka
USR_ENABLED	Vėlevėlė nurodanti ar šis vartotojas yra įjungtas / išjungtas
ROWID	Unikalus įrašo identifikatorius

- **Lentelė „USR\_ROLE“:**

Šioje lentelėje saugomos sistemos vartotojų rolės. Rolės apibrėžia sistemos vartotojų grupes, kurioms priskiriamos tam tikros naudojimosi sistema teisės (apribojimai). Roles sukuria programuotojas, arba duomenų bazės administratorius. Tai yra atliekama prieš pradėdant naudotis informacine sistema.

USR_ROLE		
<b>PK</b>	<b>ROL_ID</b>	int
	<b>ROL_NAME</b> ROL_DESCR	varchar(60) varchar(60)

<i>Atributas</i>	<i>Atributo paskirtis</i>
ROL_ID	Rolės identifikacijos numeris
ROL_NAME	Rolės pavadinimas
ROL_DESCR	Rolės apibudimas (kokia tai vartotojų grupė)

- **Lentelė „USR\_USER\_ROLE“:**

Šioje lentelėje saugomi vartotojų ir rolių ryšiai. Vienam vartotojui gali būti priskirtos kelios rolės. Informaciją į šią lentelę įveda sistemos administratorius, atlikdamas rolės priskyrimą vartotojui.

USR_USER_ROLE		
<b>PK,FK1</b>	<b>URO_USR_ID</b>	int
<b>PK,FK2</b>	<b>URO_ROL_ID</b>	int

<i>Atributas</i>	<i>Atributo paskirtis</i>
URO_USR_ID	Vartotojo identifikacijos numeris (išorinis raktas)
URO_ROL_ID	Rolės identifikacijos numeris (išorinis raktas)

- **Lentelė „USR\_ACCESSRIGHT“:**

Šioje lentelėje saugoma sistemos „teisių“ informacija. Teisės yra priskiriamos rolėms, o rolės – vartotojams. Vartotojo priskirto rolei, kuri neturi tam tikros „teisės“, naudojamas sistema yra apribojamas. „Teisės“ rolėms priskiria programuotojas arba duomenų bazės administratorius.

USR_ACCESSRIGHT		
<b>PK</b>	<b>ACR_ID</b>	int
	<b>ACR_NAME</b> ACR_DESCR	char(60) varchar(255)

Atributas	Atributo paskirtis
ACR_ID	„Teisės“ identifikacijos numeris
ACR_NAME	„Teisės“ pavadinimas. Pagal jį atliekamas teisių tikrinimas
ACR_DESCR	„Teisės“ apibūdinimas, teisės apribojimų paaiškinimas

- **Lentelė „USR\_ROLE\_ACCESSRIGHT“:**

Šioje lentelėje saugoma rolių ir teisių ryšiai. Informaciją į šią lentelę įveda programuotojas arba duomenų bazės administratorius.

USR_ROLE_ACCESSRIGHT		
<b>PK,FK1</b>	<b>RAR_ROL_ID</b>	int
<b>PK,FK2</b>	<b>RAR_ACR_ID</b>	int

Atributas	Atributo paskirtis
RAR_ROL_ID	„Teisės“ identifikacijos numeris
RAR_ACR_ID	„Teisės“ pavadinimas. Pagal jį atliekamas teisių tikrinimas

### **Sistemos konfigūracijos lentelių grupė:**

Ši lentelių grupė yra naudojama informacijos iš griežtai apibrėžtos ir nekintančios aibės saugojimui. Naudojant šią lentelių grupę, nereikia kurti daug nereikalingų lentelių (nes informacija jose nekinta), todėl supaprastėja duomenų bazės schema. Informaciją į šias lenteles įveda programuotojas.

- **Lentelė „CFG\_INDEX“:**

Šioje lentelėje saugoma konfigūracijos indeksų informacija. Konfigūracijos indeksas, tai įrašas grupuojantis konfigūracijos pozicijas (įrašus iš CFG\_POS lentelės). Jis – tai tam tikros nekintamos aibės pavadinimas

CFG_INDEX		
<b>PK</b>	<b>CDX_ID</b>	int
	<b>CDX_NAME</b> CDX_DESCR	varchar(60) char(255)

Atributas	Atributo paskirtis
CDX_ID	Konfigūracijos indekso identifikacijos numeris
CDX_NAME	Konfigūracijos indekso pavadinimas
CDX_DESCR	Konfigūracijos indekso apibūdinimas



- **Lentelė „CFG\_POS“:**

Šioje lentelėje saugoma konfigūracijos pozicijų informacija. Konfigūracijos pozicija, tai aibė, kurią apibrėžia konfigūracijos indeksas, elementas. Vienas konfigūracijos indeksas, gali turėti daug konfigūracijos pozicijų (elementų).

CFG_POSITION		
<b>PK</b>	<b>CPS_ID</b>	int
<b>FK1</b>	<b>CPS_NR</b>	smallint
	<b>CPS_NAME</b>	varchar(60)
	<b>CPS_DESCR</b>	varchar(255)
	<b>CPS_CDX_ID</b>	int

<i>Atributas</i>	<i>Atributo paskirtis</i>
CPS_ID	Konfigūracijos pozicijos identifikacijos numeris
CPS_NR	Konfigūracijos pozicijos eilės numeris konfigūracijos indekse
CPS_NAME	Konfigūracijos pozicijos pavadinimas
CPS_DESCR	Konfigūracijos pozicijos apibūdinimas
CPS_CDX_ID	Konfigūracijos indekso identifikacijos numeris (išorinis raktas)

**Defektų, savybių ir reikalavimų lentelių grupė:**

Šioje lentelių grupėje saugomi visi sistemos veiklos duomenys (*angl. Business scope data*), t.y. duomenys apie defektus, savybes, reikalavimus ir kita su jais susijusi informacija.

- **Lentelė „PRJ\_PROJECT“:**

Šioje lentelėje saugoma programinės įrangos projektų informacija. *Projektas* tai esybė, kuriai priklauso *defektai, savybės ir reikalavimai*. Ji yra šių keturių esybių hierarchijos viršuje. Projektus sistemos pagalba sukuria ir redaguoja sistemos vartotojai, turintys projekto vadovo teises.

PRJ_PROJECT		
<b>PK</b>	<b>PRJ_ID</b>	int
<b>FK1</b> <b>FK2</b>	<b>PRJ_NAME</b>	varchar(255)
	<b>PRJ_DESCR</b>	varchar(1000)
	<b>PRJ_PARENT_PRJ_ID</b>	int
	<b>PRJ_MANAGED_USR_ID</b>	int
	<b>PRJ_ANALYSED_USR_ID</b>	int
	<b>CREATED_USER</b>	varchar(50)
	<b>CREATED_DATE</b>	datetime
	<b>EDITED_USER</b>	varchar(50)
	<b>EDITED_DATE</b>	datetime
	<b>ROWID</b>	uniqueidentifier

<i>Atributas</i>	<i>Atributo paskirtis</i>
PRJ_ID	Projekto identifikacijos numeris
PRJ_NAME	Projekto pavadinimas
PRJ_DESCR	Projekto apibūdinimas
PRJ_PARENT_PRJ_ID	Projekto tėvinio projekto identifikacijos numeris
PRJ_MANAGED_USR_ID	Projekto vadovo vartotojo identifikacijos numeris
PRJ_ANALYZED_USR_ID	Projekto projektuotojo identifikacijos numeris
CREATED_USER	Vartotojo, sukūrusio įrašą, informacija
CREATED_DATE	Įrašo sukūrimo data
EDITED_USER	Vartotojo, kuris paskutinis, redagavo įrašą, informacija
EDITED_DATE	Paskutinio redagavimo data
ROWID	Unikalus įrašo identifikatorius

- Lentelė „DEF\_DEFECT“:

Šioje lentelėje yra saugoma programinės įrangos projekto defektų informacija. Defektai, tai įvairios programinės įrangos realizavimo klaidos ir neatitikimai užsakovo pateiktiems reikalavimams. Defektus kuria vartotojai turintys programuotojo ir testuotojo teises.

DEF_DEFECT		
PK	DEF_ID	int
FK2 FK1	DEF_NUMBER	int
	DEF_NAME	varchar(255)
	DEF_DESCR	varchar(1000)
	DEF_REPRODUCE	varchar(1000)
	DEF_PRIORITY_CFGNR	smallint
	DEF_SEVERITY_CFGNR	smallint
	DEF_REPORTED_USR_ID	int
	DEF_REPORTED_DATE	datetime
	DEF_STATUS_CFGNR	smallint
	DEF_ASSIGNED_USR_ID	int
	DEF_PRJ_ID	int
	CREATED_USER	varchar(50)
	CREATED_DATE	datetime
	EDITED_USER	varchar(50)
	EDITED_DATE	datetime
ROWID	uniqueidentifier	

Atributas	Atributo paskirtis
DEF_ID	Defekto identifikacijos numeris
DEF_NUMBER	Defekto informacinis identifikacijos numeris
DEF_NAME	Defekto pavadinimas
DEF_DESCR	Defekto apibūdinimas
DEF_REPRODUCE	Defekto atkūrimo informacija
DEF_PRIORITY_CFGNR	Defekto prioriteto konfigūracijos pozicijos numeris
DEF_SEVERITY_CFGNR	Defekto svarbos konfigūracijos pozicijos numeris
DEF_REPORTED_USR_ID	Vartotojo, raportavusio defekto pasirodymą numeris
DEF_REPORTED_DATE	Raportavimo data
DEF_STATUS_CFGNR	Defekto statusas (Priskirtas, Pataisytas ir t.t.)
DEF_ASSIGNED_USR_ID	Vartotojo, atsakingo už ištaisymą, numeris
DEF_PRJ_ID	Projekto, kuriam defektas priklauso, numeris
CREATED_USER	Vartotojo, sukūrusio įrašą, informacija
CREATED_DATE	Įrašo sukūrimo data
EDITED_USER	Vartotojo redagavusio įrašą, informacija
EDITED_DATE	Paskutinio redagavimo data
ROWID	Unikalus įrašo identifikatorius

- Lentelė „FET\_FEATURE“:

Šioje lentelėje saugoma programinės įrangos projekto siūlomų papildomų savybių informacija. Savybės tai užsakovo, ar kito sistemos vartotojo pateikiami papildomo funkcionalumo pageidavimai projektui. Savybė nėra reikalavimas.

FET_FEATURE		
PK	FET_ID	int
FK2 FK1	FET_NUMBER	int
	FET_NAME	varchar(255)
	FET_DESCR	varchar(1000)
	FET_REQUEST_USR_ID	int
	FET_REQUEST_DATE	datetime
	FET_STATUS_CFGNR	smallint
	FET_ASSIGNED_USR_ID	int
	FET_PERCENT_COMPLETE	smallint
	FET_PRJ_ID	int
	CREATED_USER	varchar(50)
	CREATED_DATE	datetime
	EDITED_USER	varchar(50)
	EDITED_DATE	datetime
	ROWID	uniqueidentifier

Atributas	Atributo paskirtis
FET_ID	Savybės identifikacijos numeris
FET_NUMBER	Savybės informacinis identifikacijos numeris
FET_NAME	Savybės pavadinimas
FET_DESCR	Savybės apibūdinimas
FET_REQUEST_USR_ID	Vartotojo, raportavusio defekto pasirodymą numeris
FET_REQUEST_DATE	Raportavimo data
FET_STATUS_CFGNR	Savybės statusas (Priskirtas, Realizuojama ir t.t.)
FET_ASSIGNED_USR_ID	Vartotojo, atsakingo už realizavimą, numeris
FET_PERCENT_COMPLETE	Procentali savybės realizavimo būsenos išraiška
FET_PRJ_ID	Projekto, kuriam defektas priklauso, numeris
CREATED_USER	Vartotojo, sukūrusio įrašą, informacija
CREATED_DATE	Įrašo sukūrimo data
EDITED_USER	Vartotojo redagavusio įrašą, informacija
EDITED_DATE	Paskutinio redagavimo data
ROWID	Unikalus įrašo identifikatorius

- **Lentelė „REQ\_REQUIREMENT“:**

Šioje lentelėje saugoma projektui keliamų reikalavimų informacija. Reikalavimai yra įvairūs užsakovo pateikti funkcionalumo ir patogumo pageidavimai ir pastabos. Informaciją į šią lentelę įveda vartotojas turintis projektuotojo teises.

REQ_REQUIREMENT		
PK	REQ_ID	int
FK1	REQ_NUMBER	int
	REQ_NAME	varchar(255)
	REQ_TYPE_CFGNR	smallint
	REQ_DESCR	varchar(1000)
	REQ_ASSIGNED_USR_ID	int
	CREATED_USER	varchar(50)
	CREATED_DATE	datetime
	EDITED_USER	varchar(50)
	EDITED_DATE	datetime
	ROWID	uniqueidentifier

Atributas	Atributo paskirtis
REQ_ID	Reikalavimo identifikacijos numeris
REQ_NUMBER	Reikalavimo informacinis identifikacijos numeris
REQ_NAME	Reikalavimo pavadinimas
REQ_TYPE_CFGNR	Reikalavimo apibūdinimas
REQ_DESCR	Reikalavimo apibūdinimas
REQ_ASSIGNED_USR_ID	Vartotojo atsakingo už reikalavimo realizavimą numeris
CREATED_USER	Vartotojo, sukūrusio įrašą, informacija
CREATED_DATE	Įrašo sukūrimo data
EDITED_USER	Vartotojo redagavusio įrašą, informacija
EDITED_DATE	Paskutinio redagavimo data
ROWID	Unikalus įrašo identifikatorius

- **Lentelė „REQ\_RELATION“:**

Šioje lentelėje saugoma reikalavimų tarpusavio ryšių informacija. Tam tikri vartotojo, veiklos, sistemai ir kiti reikalavimai gali būti tarpusavyje susiję. Šią informaciją įveda vartotojas turintis projektuotojo teises.

REQ_RELATION		
PK	RRL_REQ_ID	int
PK,FK1	RRL_TARGET_REQ_ID	int
	RRL_DESCR	varchar(1000)

Atributas	Atributo paskirtis
RRL_REQ_ID	Pirmasis ryšio reikalavimas
RRL_TARGET_REQ_ID	Antrasis ryšio reikalavimas
RRL_DESCR	Konfigūracijos indekso apibūdinimas

- **Lentelė „DEF\_DEFECT\_REQUIREMENT“:**

Šioje lentelėje saugoma defektų ir reikalavimų ryšių informacija. Defektui gali būti priskiriami keli reikalavimai, tokiu būdu siekiant parodyti, kad defektas yra reikalavimų neatitikimo pasekmė, arba suteikti papildomos informacijos.

DEF_DEFECT_REQUIREMENT		
PK,FK1	<u>DRQ_DEF_ID</u>	int
PK,FK2	<u>DRQ_REQ_ID</u>	int

Atributas	Atributo paskirtis
DRQ_DEF_ID	Defekto identifikacijos numeris (išorinis raktas)
DRQ_REQ_ID	Reikalavimo identifikacijos numeris (išorinis raktas)

- **Lentelė „FET\_FEATURE\_REQUIREMENT“:**

Šioje lentelėje saugoma savybių ir reikalavimų ryšių informacija. Savybei gali būti priskiriami keli reikalavimai, tokiu būdu siekiant pabrėžti, kad realizuojant savybę turėtų būti vadovojamasi nurodytais reikalavimais.

FET_FEATURE_REQUIREMENT		
PK,FK1	<u>FRQ_FET_ID</u>	int
PK,FK2	<u>FRQ_REQ_ID</u>	int

Atributas	Atributo paskirtis
FRQ_FET_ID	Savybės identifikacijos numeris (išorinis raktas)
FRQ_REQ_ID	Reikalavimo identifikacijos numeris (išorinis raktas)

- **Lentelė „ATT\_ATTACHMENT“:**

Ši lentelė yra papildomos informacijos, arba priedų (*angl. Attachment*), prisegamos prie tam tikrų esybių (*Defekto, Savybės, Reikalavimo*), saugykla. Joje saugoma bet kokio formato informacija, tokia kaip paveikslėliai, dokumentai, ekrano nuotraukos ir t.t.

ATT_ATTACHMENT		
PK	<u>ATT_ID</u>	int
	ATT_NAME	varchar(255)
	ATT_EXTENSION	varchar(10)
	ATT_TYPE_CFGNR	smallint
	ATT_CONTENT	image
	CREATED_USER	varchar(50)
	CREATED_DATE	datetime
	EDITED_USER	varbinary(50)
	EDITED_DATE	datetime
	ROWID	uniqueidentifier

Atributas	Atributo paskirtis
ATT_ID	Priedo identifikacijos numeris
ATT_NAME	Priedo pavadinimas
ATT_EXTENSION	Priedo plėtinys
ATT_TYPE_CFGNR	Priedo tipo konfigūracijos pozicijos numeris
ATT_CONTENT	Priedo turinys (bet kokia dvejetainė informacija)
CREATED_USER	Vartotojo, sukūrusio įrašą, informacija
CREATED_DATE	Įrašo sukūrimo data
EDITED_USER	Vartotojo, kuris paskutinis, redagavo įrašą, informacija
EDITED_DATE	Paskutinio redagavimo data
ROWID	Unikalus įrašo identifikatorius

- **Lentelė „DEF\_ATTACHMENT“:**

Šioje lentelėje saugoma defektų ir priedų ryšių informacija. Defektui priskiriami priedai, siekiant palengvinti defekto taisymo procesą, suteikiant papildomos vaizdines ar kitokios informacijos.

DEF_ATTACHMENT		
PK,FK1	<u>DAT_DEF_ID</u>	int
PK,FK2	<u>DAT_ATT_ID</u>	int

Atributas	Atributo paskirtis
DAT_DEF_ID	Defekto identifikacijos numeris (išorinis raktas)
DAT_ATT_ID	Priedo identifikacijos numeris (išorinis raktas)

- **Lentelė „FET\_ATTACHMENT“:**

Šioje lentelėje saugoma savybių ir priedų ryšių informacija. Savybei priskiriami priedai, siekiant palengvinti savybės realizavimo procesą, suteikiant papildomos vaizdines ar kitokios informacijos.

FET_ATTACHMENT		
PK,FK1	<u>FAT_ATT_ID</u>	int
PK,FK2	<u>FAT_FET_ID</u>	int

Atributas	Atributo paskirtis
FAT_ATT_ID	Priedo identifikacijos numeris (išorinis raktas)
FAT_FET_ID	Savybės identifikacijos numeris (išorinis raktas)

- **Lentelė „REQ\_ATTACHMENT“:**

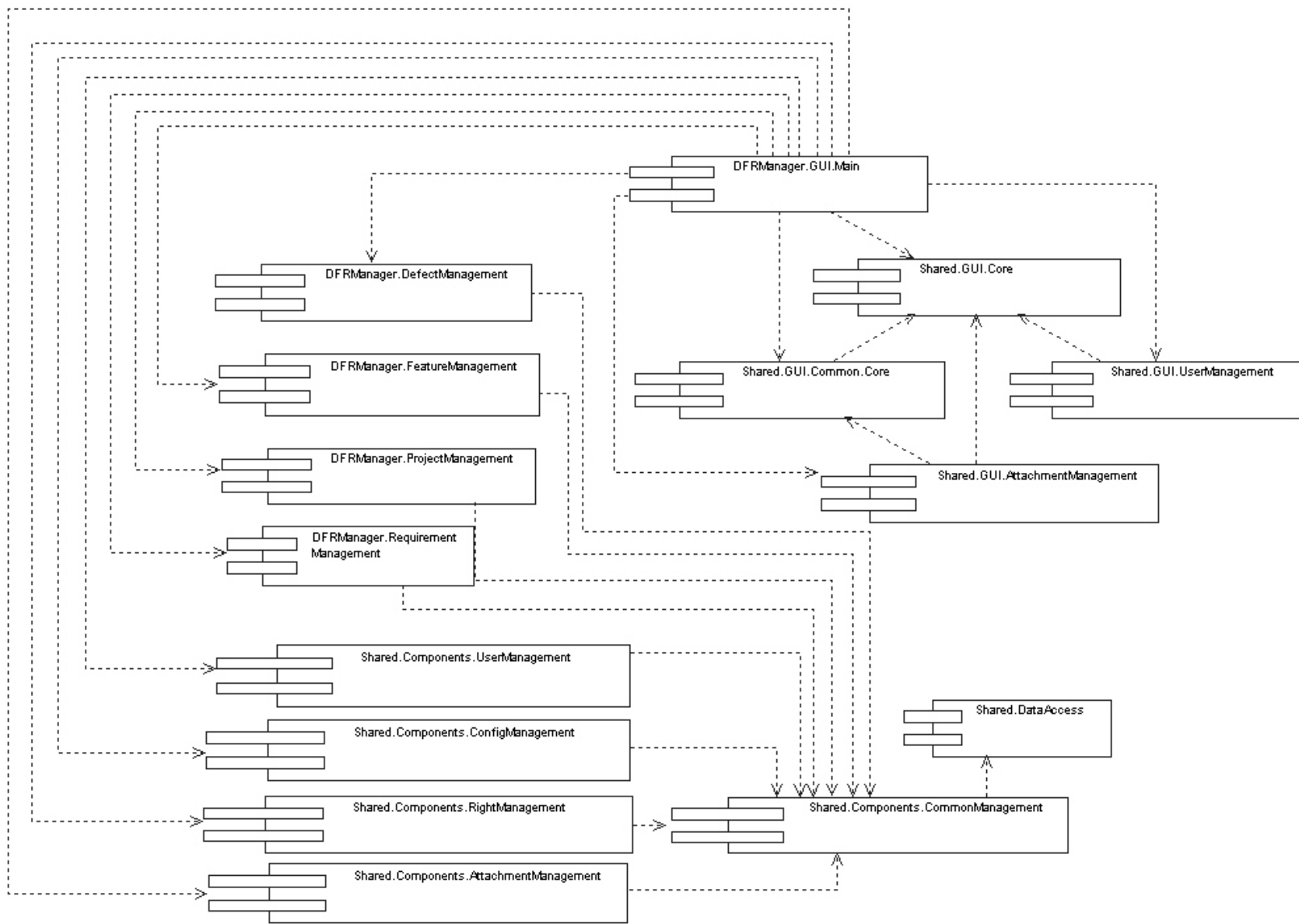
Šioje lentelėje saugoma reikalavimų ir priedų ryšių informacija. Reikalavimui priskiriami priedai, siekiant suteikti papildomos informacijos apie reikalavimą. Tai gali būti įvairiais CASE įrankiais sukurti modeliai, įvairios formalios specifikacijos, dokumentai, vaizdinė medžiaga ir panašiai.

REQ_ATTACHMENT		
PK,FK1	<u>RAT_ATT_ID</u>	int
PK,FK2	<u>RAT_REQ_ID</u>	int

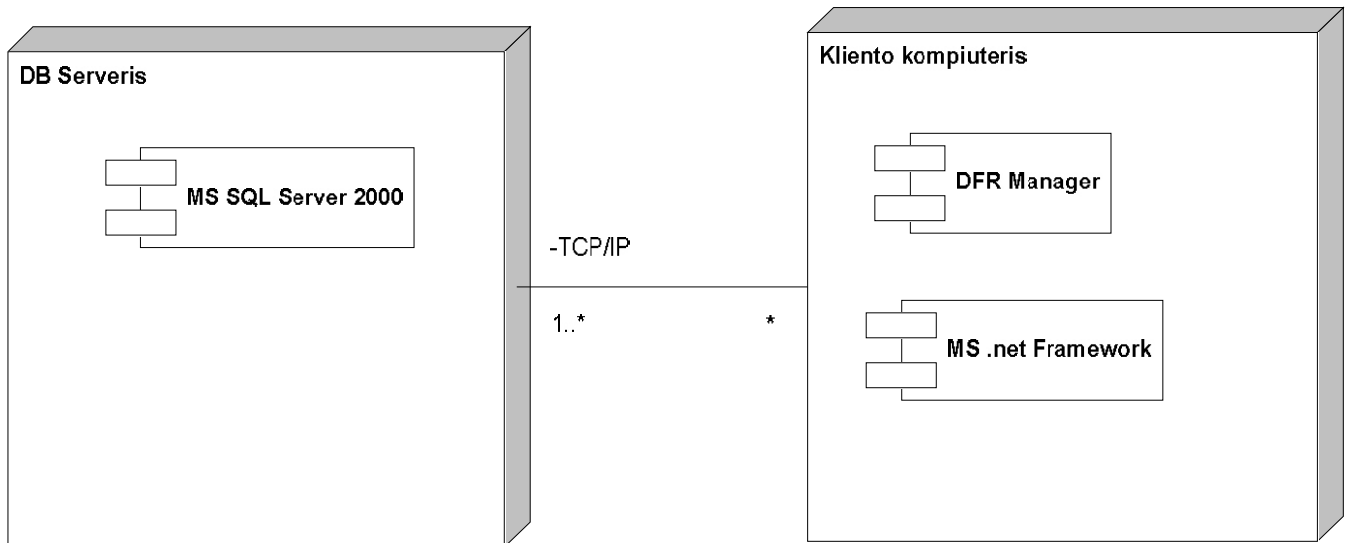
Atributas	Atributo paskirtis
RAT_ATT_ID	Priedo identifikacijos numeris (išorinis raktas)
RAT_REQ_ID	Reikalavimo identifikacijos numeris (išorinis raktas)

#### 4.2.4 Realizacijos modelis

Realizacijos modelyje pateikiamos sistemos komponentų ir sistemos diegimo diagramos (pav. 4.15 ir pav. 4.16).



**4.15 pav. Defektų, savybių ir reikalavimų valdymo sistemos komponentų modelis**



**4.16 pav. Defektų, savybių ir reikalavimų sistemos „Softrack“ diegimo diagrama**

Šiose sistemos komponentų ir diegimo diagramose gerai atsispindi visa sistemos architektūra. Tiek iš programinės, tiek iš aparatinės pusės. Ši informacija yra naudinga sistemos administratoriams.

#### 4.2.5 Reikalavimai sistemos funkcionavimo palaikymui

Lentelėje 4.5 pateikti reikalavimai sistemos „Softrack“ funkcionavimui palaikyti.

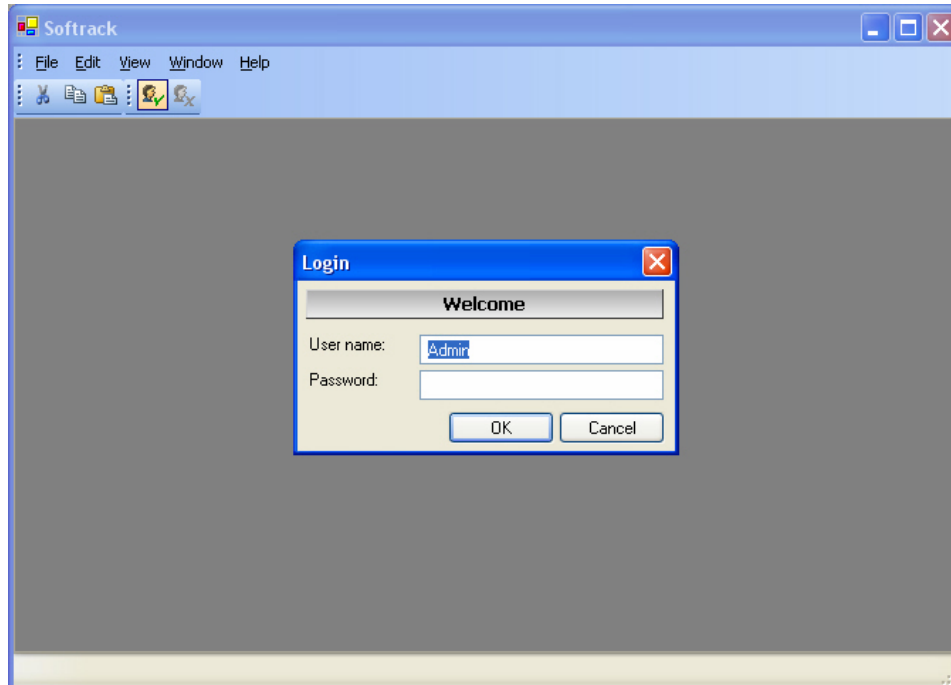
<p>Reikalavimai techninei įrangai</p>	<p><i>Klientas:</i>          Kompiuteris turintis ne lėtesnį nei 300 MHz (rekomenduojama <math>\geq</math> 800MHz) procesorių, ne mažiau nei 64MB (rekomenduojama <math>\geq</math> 256MB) operatyviosios atminties, 50 MB laisvos vietos diske ir su įdiegta 10Mbps (rekomenduojama 100Mbps) spartos tinklo plokšte; Ryšys: LAN arba Internet; TPC/IP protokolas;</p> <p><i>Serveris:</i>          Kompiuteris turintis ne lėtesnį nei 166MHz (rekomenduojama <math>\geq</math> 500MHz) procesorių, ne mažiau nei 64MB (rekomenduojama <math>\geq</math> 256MB) operatyviosios atminties, 270Mb laisvos disko vietos MS SQL Server 2000 instaliacijai; 130Mb įvairiems servisams; + papildoma disko vieta duomenims; Įdiegta 100Mbps spartos tinklo plokštė; Ryšys: LAN arba Internet; TPC/IP protokolas;</p>
<p>Reikalavimai programinei įrangai</p>	<p><i>Klientas:</i></p> <ol style="list-style-type: none"> <li>2. Microsoft Windows (2000; XP; arba vėlesnė)</li> <li>3. Microsoft .net framework v1.0</li> <li>4. Infragistics v5.1 vartotojo sąsajos komponentai (pateikiami kartu su sistema)</li> </ol> <p><i>Serveris:</i></p> <ol style="list-style-type: none"> <li>3. Microsoft Windows (2000; XP; arba vėlesnė) operacinė sistema</li> <li>4. Microsoft SQL Server 2000 DBVS</li> </ol>

4.5 lentelė. Reikalavimai sistemos funkcionalumo palaikymui

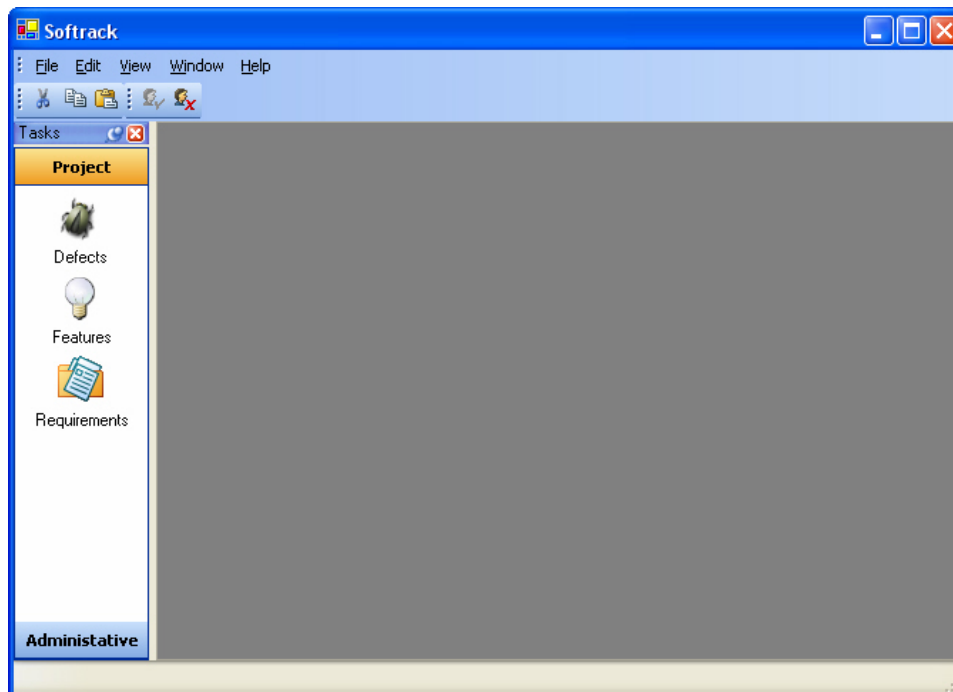


#### 4.2.6 Sistemos naudojimo instrukcija

Ijungus programą „Softrack“ automatiškai atidaromas prisijungimo prie sistemos dialogas (4.17 pav.). Prisijungimui paspartinti, automatiškai užpildomas sistemos vartotojo vardo laukas, panaudojant duomenis iš paskutinio prisijungimo.



4.17 pav. Prisijungimo langas prie sistemos „Softrack“

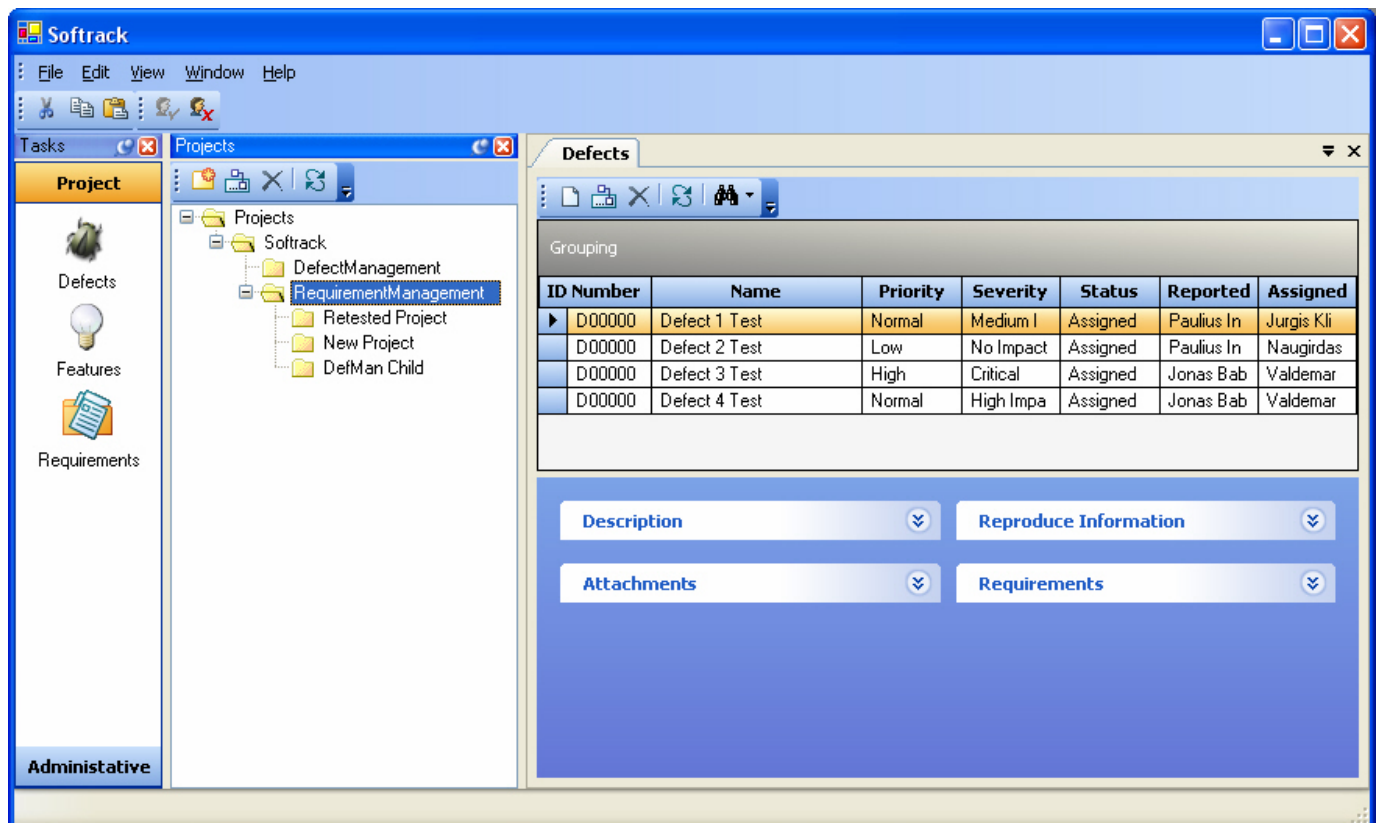


4.18 pav. Pagrindinis programos langas po prisijungimo

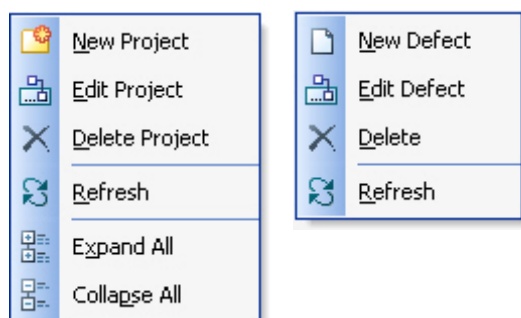
4.18 paveikslėlyje pateiktas pagrindinis programos langas. Kairėje lango pusėje yra programos langų valdymo juosta. Administratoriaus teises turintiems sistemos vartotojams, langų valdymo juostoje rodoma administravimo grupė „Administrative“, kurioje yra sistemos vartotojų informacijos langų valdymo įrankiai (mygtukas „Users“). Administratoriaus teisių neturintiems sistemos vartotojams rodoma tik projekto informacijos langų valdymo grupė, kurioje yra trys mygtukai:

1. „Defects“ – paspaudus šį mygtuką atidaromas defektų valdymo langas.
2. „Features“ – paspaudus šį mygtuką atidaromas savybių valdymo langas.
3. „Requirements“ – paspaudus šį mygtuką atidaromas reikalavimų valdymo langas.

Programos „Softrack“ langų valdymo juostoje paspaudus mygtuką „Defects“ atidaromas paveikslėlyje 4.19 pavaizduotas projektų medis ir defektų valdymo langas. Projektų medyje pasirinkus konkretų projektą, parodomi visi tam projektui priklausantys defektai. Defektų lentelės apačioje yra defekto pagrindinės informacijos peržiūros juosta. Joje vartotojas gali matyti pasirinkto defekto aprašymą („Description“), defekto atkūrimo informaciją („Reproduce Information“), prisegtą papildomą informaciją („Attachments“) bei susijusius reikalavimus („Requirements“).



4.19 pav. Programos „Softrack“ defektų valdymo langas



4.20 pav. Projektų medžio ir defektų valdymo kontekstiniai meniu

Paveikslėlyje 4.20 pavaizduoti projektų medžio ir defektų lentelės valdymo kontekstiniai meniu. Šių kontekstinių meniu pagalba, galima atlikti tokius veiksmus:

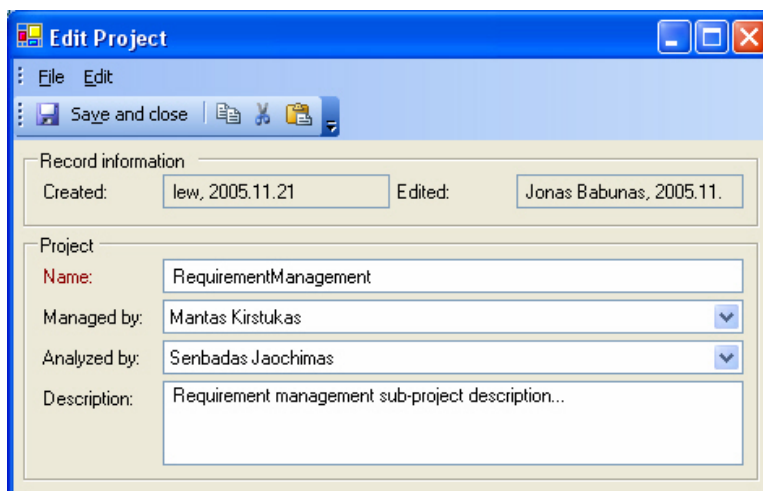
- **Projektų medžio valdymo meniu:**

Komandos pavadinimas	Komandos paaiškinimas
New Project	„Naujas projektas“ – įterpia naują projektą į projektų medį
Edit Project	„Redaguoti projektą“ – redaguoti medyje pasirinktą projektą
Delete Project	„Ištrinti projektą“ – pašalinti pasirinktą projektą iš projektų medžio
Refresh	„Atnaujinti“ – atnaujinti informaciją projektų medyje
Expand All	„Išskleisti viską“ – išplečia visas projektų medžio šakas
Collapse All	„Suskleisti viską“ – suskleidžia visas projektų medžio šakas

- **Defektų valdymo meniu:**

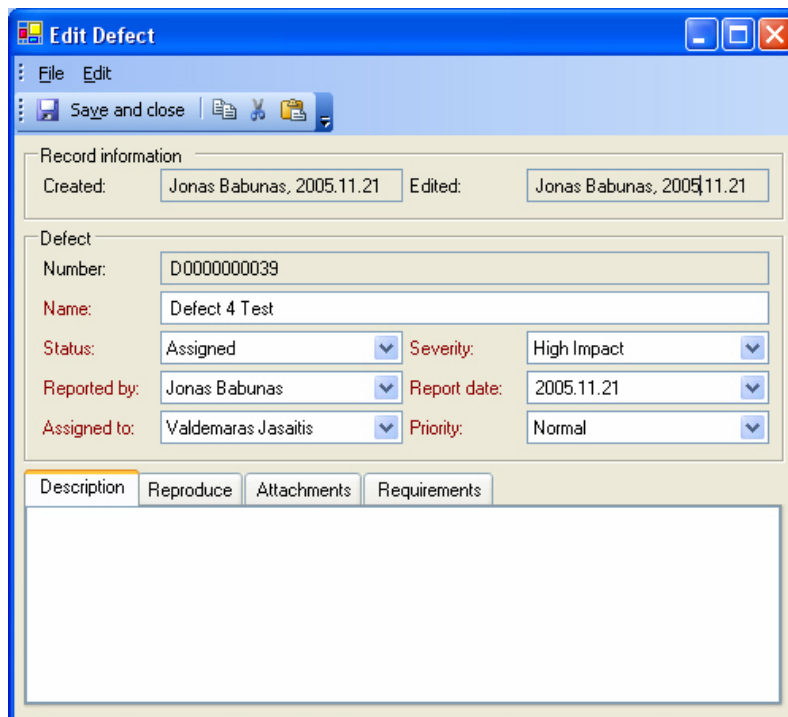
Komandos pavadinimas	Komandos paaiškinimas
New Defect	„Naujas defektas“ – įterpia naują defektą į defektų lentelę
Edit Defect	„Redaguoti defektą“ – redaguoti defektų lentelėje pasirinktą defektą
Delete	„Ištrinti“ – pašalinti pasirinktą defektą iš defektų lentelės
Refresh	„Atnaujinti“ – atnaujinti informaciją defektų lentelėje

Paveikslėlyje 4.21 pateiktas projekto sukūrimo / redagavimo langas. Lange įvedama informacija apie projektą („Name“, „Description“) ir projektą valdančius įmonės darbuotojus („Managed by“, „Analyzed by“). Redaguojant projektą, lango viršuje taip pat pateikiama projekto įrašo sukūrimo ir redagavimo informacija. Projekto informacija išsaugoma duomenų bazėje paspaudus mygtuką „Save and Close“ arba „Save“.



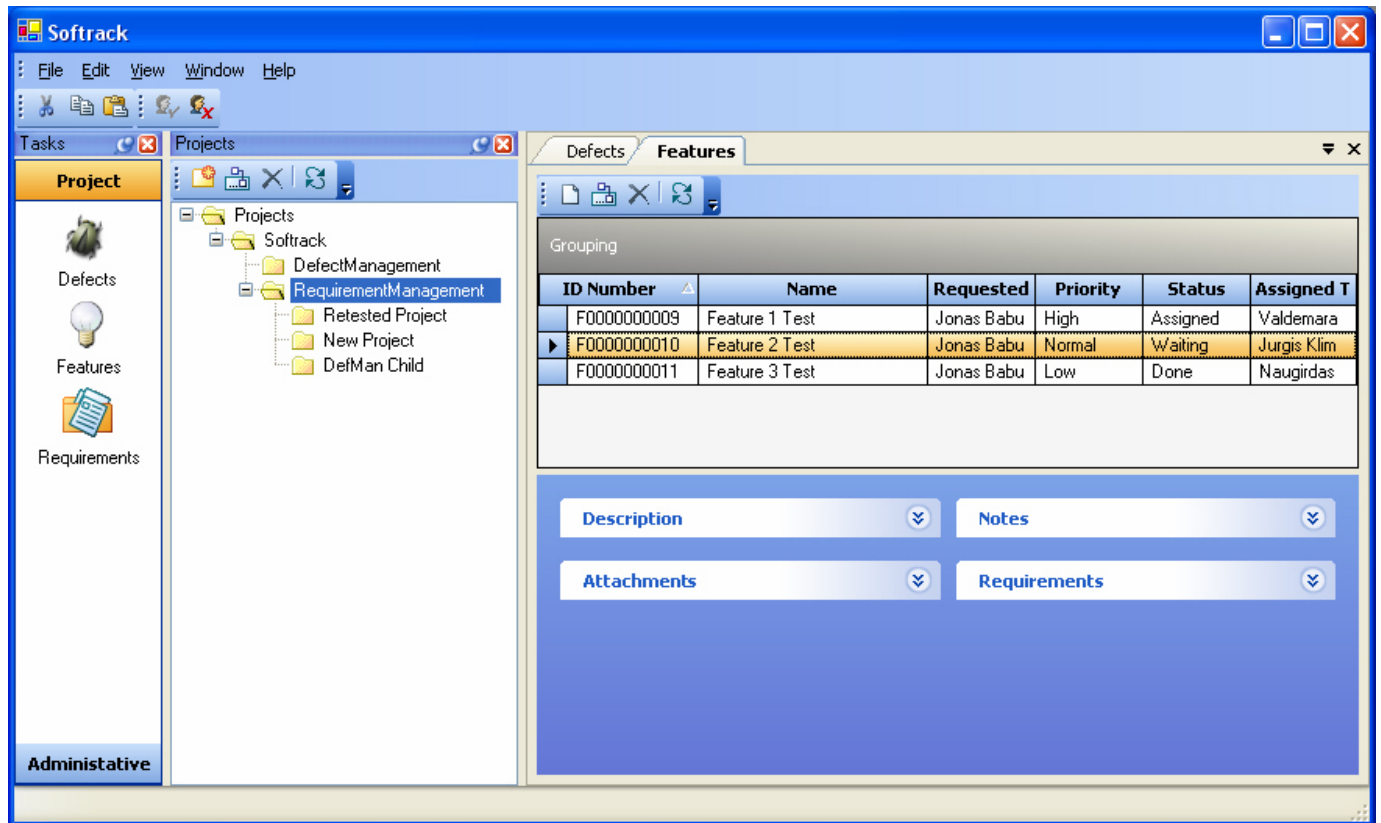
**4.21 pav. Projekto sukūrimo / redagavimo dialogas**

Paveikslėlyje 4.22 pateiktas defekto sukūrimo / redagavimo langas. Lange įvedama informacija apie defektą, jį suradusius ir už pašalinimą atsakingus įmonės darbuotojus, bei kita įvairi defektui būdinga informacija. Redaguojant defektą, lango viršuje taip pat pateikiama defekto įrašo sukūrimo ir redagavimo informacija. Defekto informacija išsaugoma duomenų bazėje paspaudus mygtuką „*Save and Close*“ arba „*Save*“.

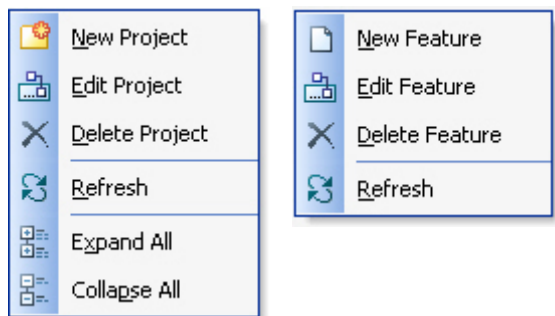


**4.22 pav. Projekto sukūrimo / redagavimo dialogas**

Langų valdymo juostoje paspaudus mygtuką „Features“ atidaromas paveikslėlyje 4.23 pavaizduotas projektų medis ir savybių valdymo langas. Projektų medyje pasirinkus konkretų projektą, parodomos visos tam projektui priklausančios savybės. Savybių lentelės apačioje yra savybės pagrindinės informacijos peržiūros juosta. Joje vartotojas gali matyti pasirinktos savybės aprašymą („Description“), pastabas („Notes“), prisegtą papildomą informaciją („Attachments“) bei susijusius reikalavimus („Requirements“).



4.23 pav. Programos „Softrack“ savybių valdymo langas



4.24 pav. Projektų medžio ir savybių valdymo kontekstiniai meniu

Paveikslėlyje 4.24 pavaizduoti projektų medžio ir savybių lentelės valdymo kontekstiniai meniu. Šių kontekstinių meniu pagalba, galima atlikti tokius veiksmus:

- **Projektų medžio valdymo meniu:**

Komandos pavadinimas	Komandos paaiškinimas
New Project	„Naujas projektas“ – įterpia naują projektą į projektų medį
Edit Project	„Redaguoti projektą“ – redaguoti medyje pasirinktą projektą
Delete Project	„Ištrinti projektą“ – pašalinti pasirinktą projektą iš projektų medžio
Refresh	„Atnaujinti“ – atnaujinti informaciją projektų medyje
Expand All	„Išskleisti viską“ – išplečia visas projektų medžio šakas
Collapse All	„Suskleisti viską“ – suskleidžia visas projektų medžio šakas

- **Savybių valdymo meniu:**

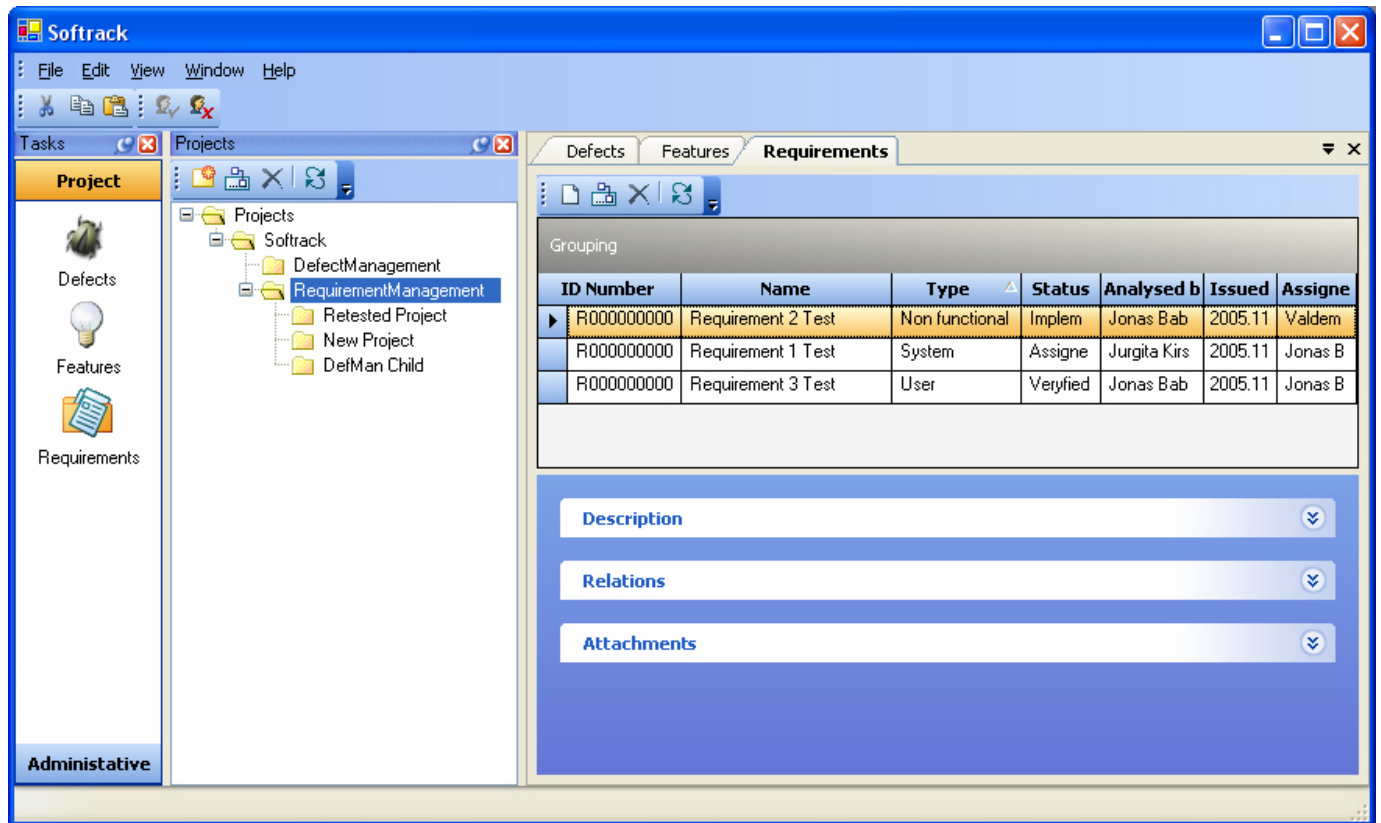
Komandos pavadinimas	Komandos paaiškinimas
New Feature	„Nauja savybė“ – įterpia naują savybę į savybių lentelę
Edit Feature	„Redaguoti savybę“ – redaguoti savybių lentelėje pasirinktą savybę
Delete	„Ištrinti“ – pašalinti pasirinktą savybę iš savybių lentelės
Refresh	„Atnaujinti“ – atnaujinti informaciją savybių lentelėje

**4.25 pav. Savybės sukūrimo / redagavimo dialogas**

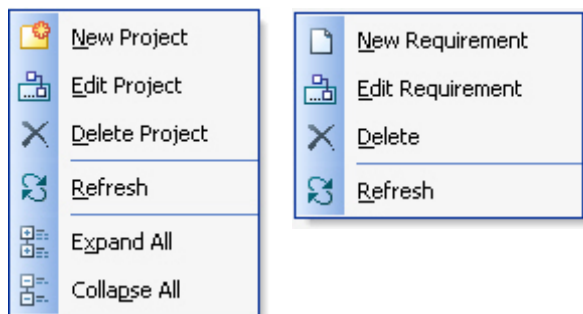
Paveikslėlyje 4.25 pateiktas savybės sukūrimo / redagavimo langas. Lange įvedama informacija apie savybę, ją sukūrusius ir už realizavimą atsakingus įmonės darbuotojus bei kita įvairi savybei būdinga informacija. Redaguojant savybę, lango viršuje taip pat pateikiama įrašo sukūrimo ir redagavimo

informacija. Savybės informacija išsaugoma duomenų bazėje paspaudus mygtuką „*Save and Close*“ arba „*Save*“.

Langų valdymo juostoje paspaudus mygtuką „*Requirements*“ atidaromas paveikslėlyje 4.26 pavaizduotas projektų medis ir reikalavimų valdymo langas. Projektų medyje pasirinkus konkretų projektą, parodomi visi tam projektui priklausantys reikalavimai. Reikalavimų lentelės apačioje yra reikalavimo pagrindinės informacijos peržiūros juosta. Joje vartotojas gali matyti pasirinkto *reikalavimo aprašymą* („*Description*“), *reikalavimų ryšius* („*Relations*“) bei prisegtą *papildomą informaciją* („*Attachments*“).



4.26 pav. Programos „Softrack“ reikalavimų valdymo langas



4.27 pav. Projektų medžio ir reikalavimų valdymo kontekstiniai meniu

Paveikslėlyje 4.27 pavaizduoti projektų medžio ir reikalavimų lentelės valdymo kontekstiniai meniu.

Šių kontekstinių meniu pagalba, galima atlikti tokius veiksmus:

- **Projektų medžio valdymo meniu:**

Komandos pavadinimas	Komandos paaiškinimas
New Project	„Naujas projektas“ – įterpia naują projektą į projektų medį
Edit Project	„Redaguoti projektą“ – redaguoti medyje pasirinktą projektą
Delete Project	„Ištrinti projektą“ – pašalinti pasirinktą projektą iš projektų medžio
Refresh	„Atnaujinti“ – atnaujinti informaciją projektų medyje
Expand All	„Išskleisti viską“ – išplečia visas projektų medžio šakas
Collapse All	„Suskleisti viską“ – suskleidžia visas projektų medžio šakas

- **Reikalavimų valdymo meniu:**

Komandos pavadinimas	Komandos paaiškinimas
New Requirement	„Naujas reikalavimas“ – įterpia naują reikalavimą į reikalavimų lentelę
Edit Requirement	„Redaguoti reikalavimą“ – redaguoti reikalavimų lentelėje pasirinktą reikalavimą
Delete	„Ištrinti“ – pašalinti pasirinktą reikalavimą iš reikalavimų lentelės
Refresh	„Atnaujinti“ – atnaujinti informaciją reikalavimų lentelėje

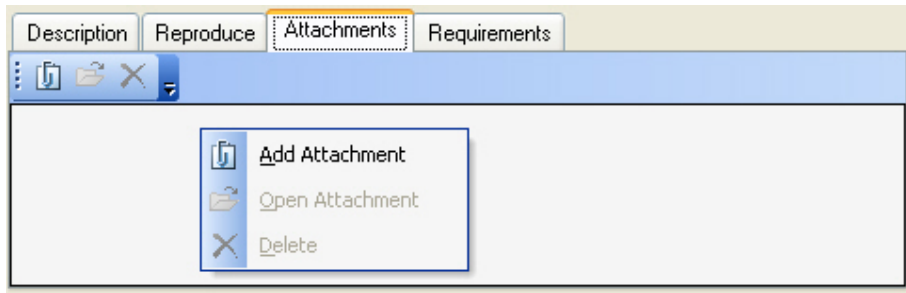
4.28 pav. Reikalavimo sukūrimo / redagavimo dialogas

Paveikslėlyje 4.28 pateiktas reikalavimo sukūrimo / redagavimo langas. Lange įvedama informacija apie reikalavimą, jį analizavusius, modeliavusius ir už realizavimą atsakingus įmonės darbuotojus bei kita įvairi reikalavimui būdinga informacija. Redaguojant reikalavimą, lango viršuje taip pat pateikiama įrašo



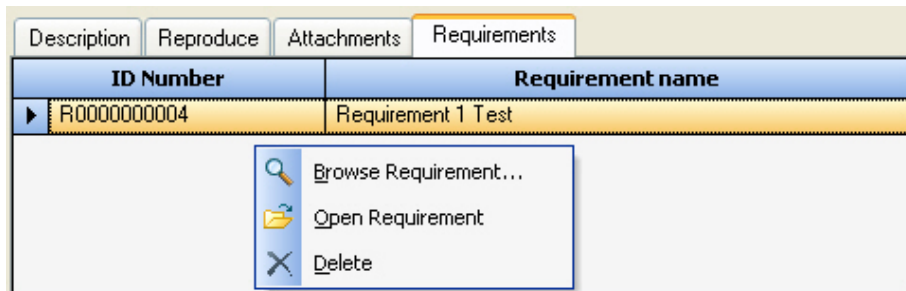
sukūrimo ir redagavimo informacija. Reikalavimo informacija išsaugoma duomenų bazėje paspaudus mygtuką „*Save and Close*“ arba „*Save*“.

Paveikslėlyje 4.29 pateikta papildomos informacijos prisegimo prie objekto (pvz. Defekto, savybės, reikalavimo) lentelė. Failas prisegamas paspaudus „*Add Attachment*“ mygtuką. Failo atidarymas ar šalinimas atliekamas paspaudus atitinkamus mygtukus „*Open Attachment*“ ir „*Delete*“. Lentelė naudojama defektų, savybių ir reikalavimų sukūrimo / redagavimo dialoguose.



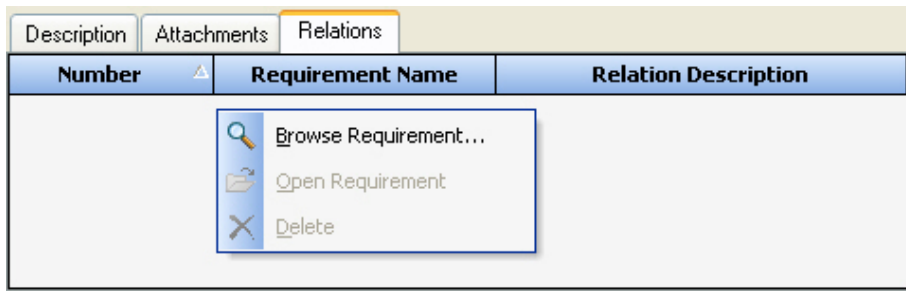
**4.29 pav. Papildomos informacijos (failo) prisegimo prie redaguojamo objekto lentelė**

Paveikslėlyje 4.30 pateikta reikalavimo susiejimo su objektu (pvz. Defektu, savybe, reikalavimu) lentelė. Reikalavimas susiejamas paspaudus „*Browse Requirement*“ mygtuką. Reikalavimo atidarymas ar šalinimas atliekamas paspaudus atitinkamus mygtukus „*Open Requirement*“ ir „*Delete*“. Lentelė naudojama defektų ir savybių sukūrimo / redagavimo dialoguose.



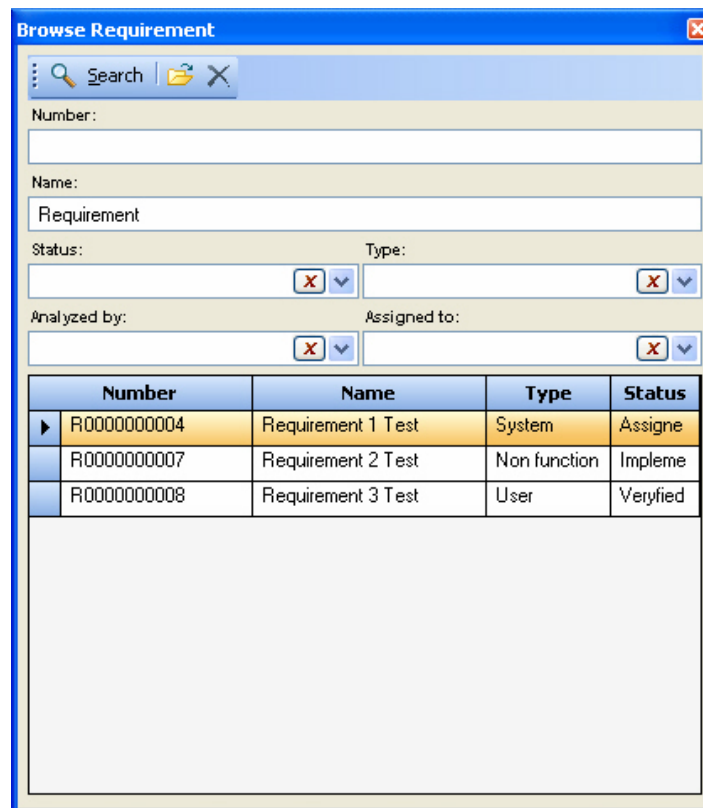
**4.30 pav. Reikalavimo susiejimo su objektu lentelė**

Paveikslėlyje 4.31 pateikta reikalavimų ryšių lentelė. Ryšys su reikalavimu sukuriamas paspaudus „*Browse Requirement*“ mygtuką. Susieto atidarymas ar šalinimas atliekamas paspaudus atitinkamus mygtukus „*Open Requirement*“ ir „*Delete*“. Lentelė naudojama reikalavimo sukūrimo / redagavimo dialoge.



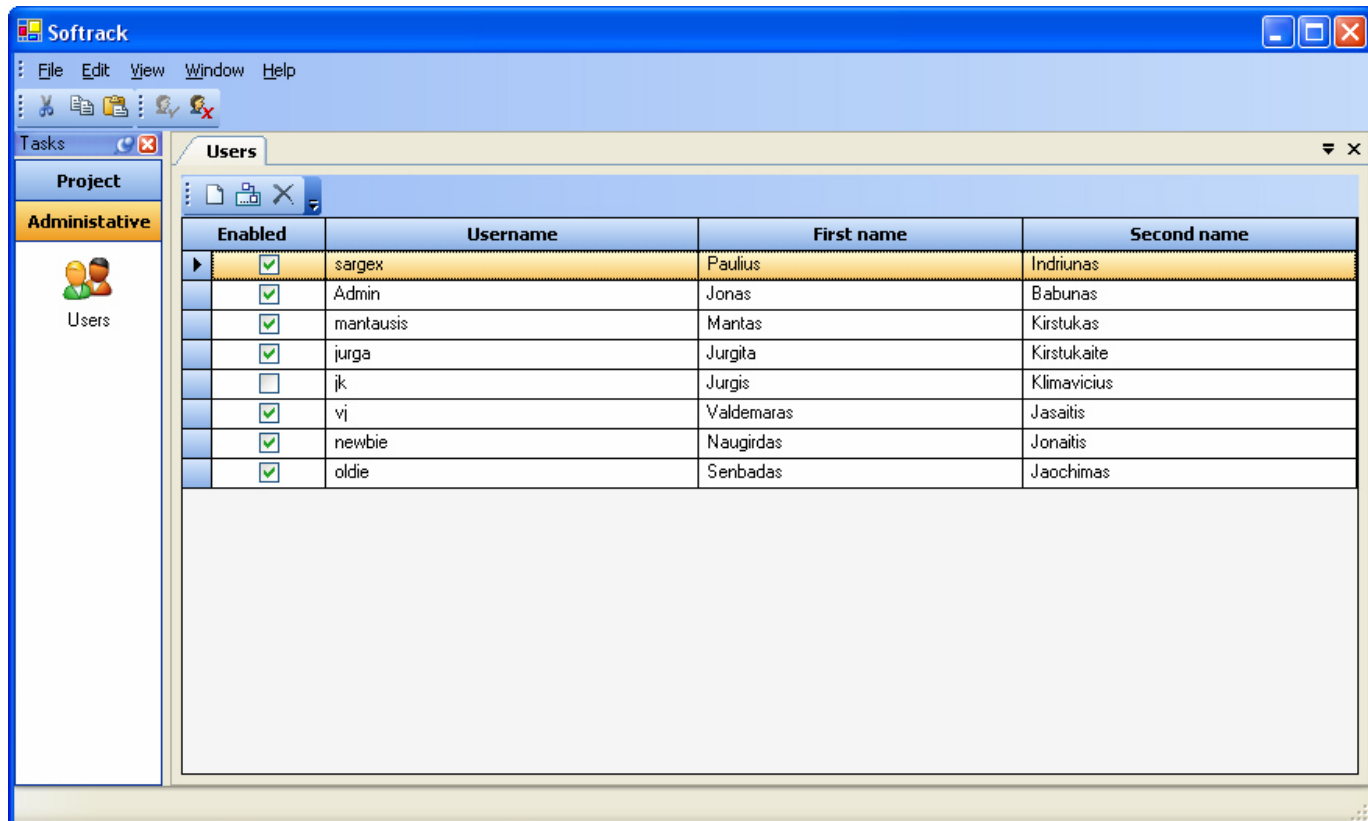
4.31 pav. Reikalavimų ryšių lentelė

4.32 paveikslėlyje pateiktas reikalavimų paieškos dialogas. Jo pagalba galima atlikti reikalavimų paiešką pagal reikalavimo informacinius laukus. Paieška atliekama paspaudus mygtuką „Search“. Visi paieškos kriterijus atitinkantys reikalavimai išvedami paieškos rezultatų lentelėje. Joje rodomas reikalavimo identifikacijos numeris, pavadinimas, tipas ir būseną. Reikalavimo rezultato pasirinkimas atliekamas du kartus spragtelėjus ant reikalavimo įrašo, rezultatų lentelėje arba paspaudus mygtuką „Open“.



4.32 pav. Reikalavimo paieškos langas

Langų valdymo juostoje paspaudus mygtuką „Users“ atidaromas paveikslėlyje 4.33 pavaizduotas sistemos vartotojų valdymo langas. Šiame lange esančioje vartotojų valdymo lentelėje rodoma visų sistemos vartotojų informacija. Lentelėje rodomas sistemos vartotojo vardas („Username“), asmens informacija („First name“, „Second name“) ir vartotojo įjungimo žymė („Enabled“).



4.33 pav. Sistemos vartotojų valdymo langas



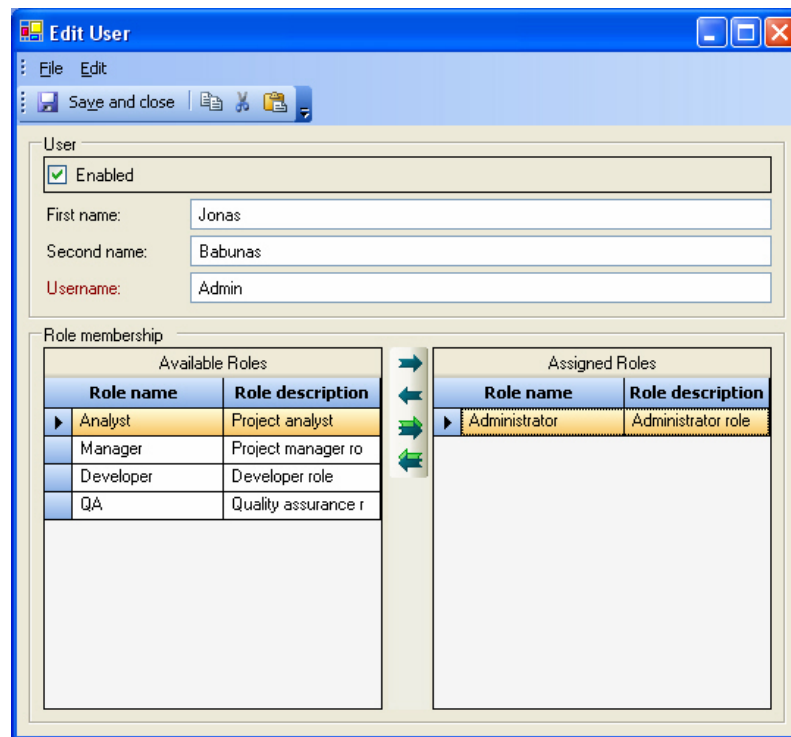
4.34 pav. Vartotojų valdymo kontekstinis meniu

Paveikslėlyje 4.34 pavaizduotas reikalavimų lentelės valdymo kontekstinis meniu. Šio kontekstinio meniu pagalba, galima atlikti tokius veiksmus:

- *Sistemos vartotojų valdymo meniu:*

Komandos pavadinimas	Komandos paaiškinimas
New User	„Naujas vartotojas“ – įterpia naują vartotoją į vartotojų valdymo lentelę
Edit User	„Redaguoti vartotoją“ – redaguoti vartotojų lentelėje pasirinktą vartotoją
Delete	„Ištrinti“ – pašalinti pasirinktą vartotoją iš vartotojų lentelės

Paveikslėlyje 4.35 pateiktas sistemos vartotojo sukūrimo / redagavimo langas. Lange įvedama vartotojo informacija, priskiriamos rolės. Priskirtos rolės rodomos dešiniojoje rolių lentelėje, o nepriskirtos – kairiojoje. Vartotojo informacija išsaugoma duomenų bazėje paspaudus mygtuką „Save and Close“ arba „Save“.



4.35 pav. Sistemos vartotojo sukūrimo / redagavimo dialogas

### 4.3 Projekto išvados

Atlikus defektų, savybių ir reikalavimų sistemos projektavimą padarytos tokios išvados:

- Siekiant padaryti programą plačiai naudojamą, grafinę vartotojo sąsają tikslinga sukurti gražią, patogią ir preciziškai tikslią. Iš sistemos vartotojo perspektyvos, grafinė sąsaja yra vienas iš pagrindinių programos kokybės vertinimo kriterijų.
- Sistemą yra tikslinga realizuoti padalinant ją į daugelį atskirų ir griežtai apibrėžto funkcionalumo komponentų. Naudojant griežtą komponentizavimą, bus lengviau surandamos klaidos, tarp komponentų nebus glaudžių ryšių, tuos pačius komponentus bus galima panaudoti kuriant kitas informacines sistemas;

## IŠVADOS

Atlikus tiriamąjį ir praktinį darbą, programinės įrangos defektų, savybių ir reikalavimų valdymo srityje, išspręstos darbo pradžioje iškeltos problemos bei uždaviniai ir padarytos tokios darba apibendrinančios išvados:

1. Išanalizavus programinę įrangą kuriančios organizacijos veiklą ir pagrindinius joje vykstančius procesus nustatyta, kad organizacijoje yra tikslinga įdiegti kompiuterizuotą projekto informacijos valdymo sistemą.
2. Atlikta analogiškų pasaulinio lygio sprendimų lyginamoji analizė. Pagal nustatytus palyginimo kriterijus, kiekvienai lyginamai sistemai apskaičiuotas bendro sistemos tinkamumo koeficientas. Pagal šį koeficientą pasirinktos dvi didžiausius koeficientus turinčios sistemos, kurių geriausių savybių pagrindu buvo projektuojamas naujas programinės įrangos defektų, savybių ir reikalavimų valdymo metodas.
3. Sukurtas programinės įrangos defektų, savybių ir reikalavimų valdymo metodas, palengvinsiantis įmonės darbuotojų darbą projektavimo, realizavimo ir kokybės užtikrinimo projekto kūrimo fazėse.
4. Naujasis metodas nuo egzistuojančių metodų išsiskiria tuo, kad reikalavimai šiame metode glaudžiai susiejami su programinės įrangos defektais ir savybėmis. Tai padidina informacijos susietumą, užtikrina, kad su defektais ar savybėmis dirbantys darbuotojai galės greitai surasti reikiamus reikalavimus. Dauguma defektų sekimo sistemų (*angl. bugtracking systems*) modelių neatsižvelgia į reikalavimų valdymo problemas. Pagrindinis naujojo metodo trūkumas yra toks, kad jis nepadengia daug reikalavimo valdymo funkcijų, kurios yra siūlomos specializuotose reikalavimų valdymo sistemose (sistemose kurios nevaldo defektų ir savybių).
5. Suprojektuota ir realizuota programinės įrangos defektų, savybių ir reikalavimų valdymo metodo informacijos srautų valdymo sistema, palengvinanti įmonės darbuotojų darbą projektavimo, realizavimo ir kokybės užtikrinimo fazėse, tokiu būdu paspartinant ir patobulinant programinės įrangos kūrimo procesą.

## LITERATŪRA:

- [1] MICROSOFT COORPORATION *Analyzing Requirements and Defining Microsoft .NET Solution Architectures*. Microsoft Press, 2003. 490 p. ISBN 0-7356-1894-1
- [2] COMPUWARE COORPORATION *Requirements Traceability for quality management*. [žiūrėta 2004 m. lapkričio 30 d.]. Prieiga per Internetą:  
<[http://www.bitpipe.com/data/detail?id=1044887018\\_760&type=RES&src=FEATURE\\_SPOTLIGHT&x=353313773](http://www.bitpipe.com/data/detail?id=1044887018_760&type=RES&src=FEATURE_SPOTLIGHT&x=353313773)>
- [3] E. WELLER, *Defect Management in Developement and Test*. 1999. [žiūrėta 2004 m. gruodžio 4 d.]. Prieiga per Internetą:  
<<http://www.stickyminds.com/sitewide.asp?ObjectId=2152&Function=DETAILBROWSE&ObjectType=ART>>
- [4] Karl E. WIEGERS, *Automating requirements management*. [žiūrėta 2005 m. spalio 15 d.]. Prieiga per Internetą: <[http://www.processimpact.com/articlesrm\\_tools.pdf](http://www.processimpact.com/articlesrm_tools.pdf)>
- [5] *PRTracker Tour 2003*. [žiūrėta 2004 m. gruodžio 19 d.]. Prieiga per Internetą: <<http://www.prtracker.com/Tour/>>
- [6] „OnTime“ *Defect and feature management system*. [žiūrėta 2004 m. gruodžio 10 d.]. Prieiga per Internetą: <<http://www.axosoft.com/products/downloads.aspx>>
- [7] C. FETTERS, T. HSU, B. SMEED *Requirements Tracing* [žiūrėta 2005 m. kovo 04 d.]. Prieiga per Internetą: <<http://www.cis.ksu.edu/~hankley/d841/Fa99/chap2.htm>>
- [8] D. LE VIE *Writing Software Requirements Specifications* [žiūrėta 2005 m. gegužės 18 d.]. Prieiga per Internetą: <<http://www.techwr-l.com/techwhirl/magazine/writing/softwarerequirementspecs.html>>
- [9] S. W. BANNERMAN *A Middleweight Requirements Management Framework* [žiūrėta 2005 m. gegužės 29 d.]. Prieiga per Internetą: <<http://reqs.tigris.org/pdfs/mrmf.pdf>>
- [10] M. PALO *Requirements Traceability* [žiūrėta 2005 m. gegužės 14 d.]. Prieiga per Internetą: <[www.cs.helsinki.fi/u/paakki/palo.pdf](http://www.cs.helsinki.fi/u/paakki/palo.pdf)>