



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Michail Mazo

**Reikalavimų specifikavimo
pasinaudojant šablonais
tyrimas**

Magistro darbas

Darbo vadovas
doc. R. Butleris

Kaunas, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas
doc. dr. R. Butleris

.....

**Reikalavimų specifikavimo
pasinaudojant šablonais
tyrimas**

Magistro darbas

Vadovas

doc. dr. R. Butleris

.....

Recenzentas

dr. B. Tamulynas

.....

Atliko

IFM-1/4 gr. stud.

M. Mazo

.....

Kaunas, 2005

TURINYS

1. Įvadas	5
2. Reikalavimų specifikavimo procesas	6
3. Reikalavimų specifikavimo dokumentas	7
3.1. IEEE standartai	7
3.2. SRS dokumento turinys	7
3.3. SRS dokumento charakteristikos	8
3.4. Skirtumas tarp SRS is SS	8
3.5. Skirtumas tarp SRS ir projektavimo reikalavimų dokumento	9
4. Reikalavimų specifikavimo šablonų ir įrankių analizė	9
4.1. Šablonų analizė	9
4.1.1. IEEE STD 830-1998	9
4.1.2. Volere	10
4.1.3. Šablonų palyginimas ir analizės išvados	11
4.2. Įrankių analizė	11
4.2.1. AnalystPRO	11
4.2.2. RequisitePRO	13
4.2.3. Analizės išvados	14
5. SRS automatizavimo įrankio konceptualus modelis	15
5.1. SRS automatizavimo įrankio konteksto specifikavimas	15
5.1.1. Komponentinė sistema	15
5.1.2. Vieninga sistema	15
5.1.3. Darbo tikslas ir siejami privalumai	16
5.1.4. Kompouterizuojamos sistemos funkcijos	17
5.1.5. Rezultato kokybės kriterijai	17
5.2. Reikalavimų specifikacija	18
5.2.1. Apribojimai reikalavimams	18
5.2.2. Veiklos konteksto diagrama	18
5.2.3. Panaudojimo atvejų diagrama	19
5.2.4. Panaudojimo atvejų sąrašas	19
5.2.5. Funkciniai reikalavimai	22
5.2.6. Nefunkciniai reikalavimai	22
5.3. Apibendrintas architektūros modelis	23
5.4. Šablono modelio parinkimas	24
5.5. Duomenų struktūros	24
5.5.1. Volere siūloma reikalavimų formų uniifikavimo strategija	24
5.5.2. Volere šablono padalinimas pagal reikalavimų formatus	26
5.5.3. Duomenų struktūra ‘Requirement Shell’	30
5.5.4. Duomenų struktūra ‘Event / Use Case List’	31
5.5.5. Duomenų struktūra ‘User / Stakeholder Profile’	31
5.5.6. Duomenų struktūra ‘Illustrations’	32
5.5.7. Duomenų struktūra ‘Glossary’	32

5.5.8. Sistemos pagalbinės duomenų struktūros	33
5.5.8.1. Duomenų struktūra ‘Template Table’	33
5.5.8.2. Duomenų struktūra ‘System Users’	33
5.5.8.3. Duomenų struktūra ‘Change Log’	34
5.5.8.4. Duomenų struktūra ‘Modify List’	34
5.5.8.5. Duomenų struktūra ‘System Settings’	35
5.5.8.6. Duomenų struktūra ‘Requirement Shell Versions’	35
5.5.9. Bendra sistemos duomenų saugyklos loginė schema	36
5.6. Duomenų apdorojimo procesai	38
5.6.1. Duomenų apdorojimo konceptas	38
5.6.2. Sistemos parametrų nustatymas	38
5.6.3. Vartotojo sukūrimas / panaikinimas	38
5.6.4. Vartotojo prisijungimas / atsijungimas	39
5.6.5. Vartotojo duomenų redagavimas	39
5.6.6. Sistemos vartotojų peržiūra	40
5.6.7. Informacija apie seansą	40
5.6.8. Reikalavimo sukūrimas / redagavimas / panaikinimas	41
5.6.9. Reikalavimo paieška	41
5.6.10. Reikalavimo užrakinimas / atrakinimas	43
5.6.11. Pakeitimų žurnalo papildymas	44
5.7. Apibendrintas sistemos klasių vaizdas	44
6. SRSTemplate eksperimentinis tyrimas	45
6.1. Sistemos panaudojamumo įvertinimas	45
6.1.1. “Constructive Interaction” testavimo metodas	45
6.1.2. MORAE testavimo paketas	46
6.1.3. Testavimo procesas	46
6.1.4. Testavimo rezultatai pagal ISO 9241	48
6.2. Sistemos palyginimas su AnalystPRO ir RequisitePRO	51
7. Išvados	53
8. Literatūra	55
9. Terminų ir sutrumpinimų žodynas	56
10. Priedai	57
10.1. Apibendrinta klasių diagrama	57
10.2. Panaudojimo atvejų sėkų diagramos	58
10.3. Sistemos vaizdai	63
10.4. Darbo CD	72
10.5. Recenzija	73
10.6. Mokslinis straipsnis	74

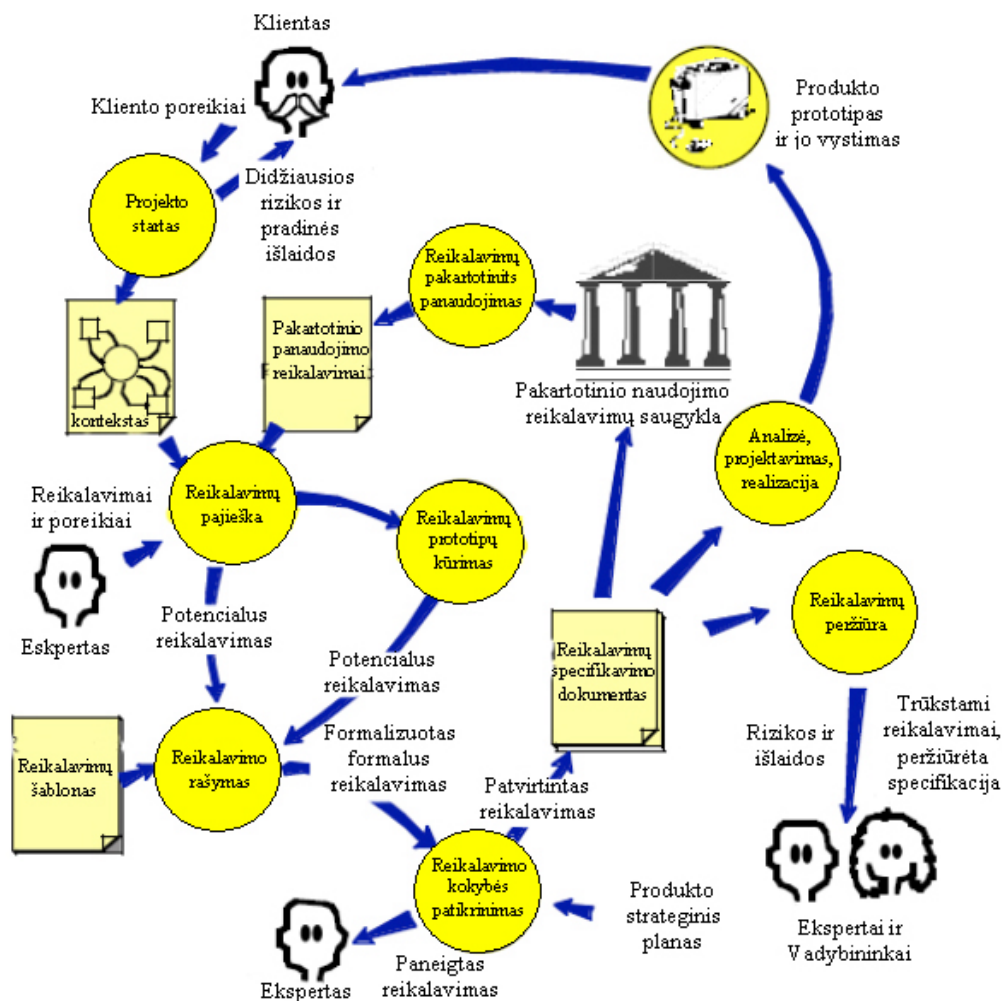
1. Įvadas

Viribus Unitis
(bendromis pastangomis)

Kiekvienas darbas ar projektas prasideda nuo balto lapo ir projektavimas visada yra didžiausia ir sunkiausia gero produkto gamybos dalis. Programinės įrangos kūryme viskas prasideda nuo reikalavimų analizės ir specifikavimo, kurie yra pastoviai tobulinami ir vėliau sudaro produkto dokumentacijos pagrindą. Egzistuoja daug įvairių šablonų, reikalavimų ir pavyzdžių, kaip reikia rašyti reikalavimų specifikavimo dokumentą. Tai didelis ir kruopštus darbas ir dažnai tam yra skyriamas atskiras atsakingas žmogus ar net darbuotojų grupė. Tačiau kaip parinkti geriausią reikalavimų specifikavimo šabloną ir efektyviausiai organizuoti reikalavimų specifikavimo rašymo eigą? Šis darbas yra skirtas išaiškinti ir ištyrti skirtingus reikalavimo specifikavimo šablonus ir sukūrti programinio produkto prototipą, kuris maksimaliai automatizuotų ir optimizuotų specifikavimo dokumento kūrymą ir leistų prie jo vienu metu efektyviai ir bendromis pastangomis dirbti keliems sistemos analitikams.

2. Reikalavimų specifیکavimo procesas

Reikalavimų programiniai įrangai specifیکavimas yra sudėtingas procesas, kuris tęsiasi viso programinio produkto kūrimo laiko metu. Volere, reikalavimų specifیکavimo šablono lideris, siūlo “Mastering the requirements process” kursus projekto vadovams, sistemos analitikams ir kitiems darbuotojams, dirbantiems prie reikalavimų programiniai įrangai specifیکavimo. Kurso apžvalgoje [1] pateikiama tokia ši procesą aprašanti diagrama (pav. 1).



Pav. 1: Reikalavimų specifیکavimo procesas

Klientas pateikia savo reikalavimus ir poreikius. Su juo yra aptariamos didžiausios įgyvendinimo problemos ir pradinės išlaidos. Projektas startuoja. Sudaromas projekto kontekstas – bendri tikslai, pagrindinės strateginės funkcijos ir esminiai reikalavimai. Toliau yra jiejiami reikalavimai kūriamai sistemai. Vyksta konsultacija su ekspertais, kurie pasiūlo reikalavimus ir poreikius. Kūriami reikalavimų prototipai. Potencialūs reikalavimai yra įnešami į reikalavimų

specifikavimo dokumento šabloną. Toliau yra patikrinami šie formalizuoti reikalavimai. Jų kokybę ir prasmingumą tikrina ekspertai. Patvirtinti reikalavimai yra įrašomi į aktualų reikalavimų specifikavimo dokumentą. Toliau jie yra pastoviai peržiūrimi ir papildomi ekspertų ir vadybininkų, kurie lygina jų atitikimą ir įgyvendinimo galimybę su projekto strateginiu planu. Patvirtinti reikalavimai yra taip pat išsaugomi pakartotinio panaudojimo reikalavimų saugykloje. Šie reikalavimai gali būti toliau dar kartą panaudoti kituose ar tame pačiame projekte. Vyksta reikalavimų analizė, produkto projektavimas atsižvelgiant į sukeltus reikalavimus ir produkto realizacija. Realizuotas produkto prototipas yra parodomas klientui ir aptariami papildomi reikalavimai produktui, kurie vėliau vėl pereina visą aukščiau nurodytą ciklą.

Reikalavimų specifikavimo procesas yra sudėtingas ir iteratyvus, tačiau šiame darbe pasiribuosime užbriežtais tikslais – šablonų analize ir reikalavimų specifikavimo automatizuotos sistemos prototipo kūrimu. Visų pirmą reikia ištirti esamus reikalavimų specifikavimo šablonus ir jų ypatybes.

3. Reikalavimų specifikavimo dokumentas

Pradedant šablonų analizę, reikėtų pradėti nuo pat pradžių – apibrėžimų ir pagrindinių aspektų. Tam yra skirtas Roberto Japengos (Robert Japenga) straipsnis “Kaip rašyti programinės įrangos reikalavimų specifikaciją” (“How to write a software requirements specification”) [2]. Remiantis šiuo straipsniu sudarysime bendrą reikalavimų specifikavimo šablonų viziją.

3.1. IEEE standartai

IEEE [3] yra puikus programinės įrangos reikalavimų specifikavimo (toliau SRS – Software Requirements Specification) šablono dokumento apibrėžimų šaltinis. Kaip SRS šablonas yra plačiai naudojamas standartas IEEE STD 830-1998. Aišku galimos jo modifikacijos atsižvelgiant į projekto ypatumus. Svarbu apart SRS turėti ir sistemos reikalavimų specifikavimo dokumentą (toliau SS – System Specification). Populiariausias SS dokumento standartas yra IEEE STD 1233-1998. Tačiau dažnai projektuojant nedidelėms sistemoms yra naudojami vienodi šablonai abiemis dokumentams.

3.2. SRS dokumento turinys

Pagal IEEE standartą SRS dokumento kūrėjas turi detaliai perduoti:

- a) *Funkcionalumą*. Ką turi daryti programinė įranga?

- b) *Išorines sąsajos*. Kaip programa sąveikauja su vartotoju, sistemos techninę įrangą, kita programinę įrangą
- c) *Eksplotacijos parametrai*. Koks yra atskirų programinės įrangos funkcijų greitis, naudingumas, reakcijos laikas, atstajimo laikas, etc
- d) *Atributus*. Kokie yra portatyvumo, teisingumo, palaikomumo, saugos ir kiti sprendimai
- e) *Projektavimo apribojimai, kūriant produktą*. Ar yra nustatyti papildomi reikalavimai ir apribojimai atsižvelgiant į realizavimo programavimo kalbą, duomenų bazes, resursus ar aplinkos kintamuosius, etc.

3.3. SRS dokumento charakteristikos

Pagal IEEE standartą SRS dokumentas turi būti:

- a) *Korektiškas*. Tai reškia, kad dokumentas turi būti visą laiką aktualus, su visais paskutintais pakeitimais.
- b) *Nedviprasmiškas*. Reikalavimas yra ndviprasmiškas tik tada, kada įmanoma tik viena vienintelė jo interpretacija
- c) *Pilnas*. Dokumentas turi turėti viską, kas reikalinga sistemos analitikams
- d) *Nuoseklus*. SRS dokumentas turi būti nuoseklus pats ir nurodomų dokumentų atžvilgiu. Jeigu komanda yra vadinama „Start and Stop“ vienoje dokumento dalyje ji negali vadintis „Start/Stop“ kitoje dalyje.
- e) *Išdėstytas pagal reikšmingumą*. Reikalavimai turi būti išdėstyti pagal jų svarbą, nes dažnai neįmanoma ar nepakanka lėšų įgyvendinti visus reikalavimus. Tai turi būtų irgi nurodyta.
- f) *Griežtas ir pagrįstas*. Negalimos tokios frazės ir paaiškinimai, kaip „Sistema turėtų dirbti patikimai“ arba „Sistema turi pateikti greitą atsakymą“. Teisingas pavizdys būtų: „Sistema turi pateikti atsakymą per 10 milisekunžių“
- g) *Koreguojamas*. Turi būti galimybė greitai ir paprastai papildyti ar koreguoti reikalavimus.
- h) *Atsekamas*. Dokumento struktūra turi būti grįžta ir kiekviena jo dalis turi būti lengvai atdekama.

3.4. Skirtumas tarp SRS ir SS

Sistemos reikalavimų dokumentas (SS) yra bendras viso produkto reikalavimų dokumentas, kuris aprašo visas jo funkcijas ir reikalavimus. Tai gali būti reikalavimai programinei įrangai, techninei įrangai ar mechaninei įrangai. Tačiau dažnai programinės įrangos reikalavimų

specifikavimo dokumentas (SRS) yra labai didelis ir yra išskiriamas į atskirą dokumentą. Kartais, kai projektas yra nedidelis, yra rašomas vienas bendras dokumentas visiems reikalavimams ir tada visi dokumentai yra apjungiami į vieną.

3.5. Skirtumas tarp SRS ir projektavimo reikalavimų specifikavimo dokumento

Reikalavimų programinei įrangai specifikavimo dokumentas nustato reikalavimus būsimam produktui, jo galimybėms ir funkcijoms programinės dalies atžvilgiu. Projektavimo reikalavimų specifikavimo dokumentas aptaria kaip tai turi būti įgyvendinta – pvz. kokia programavimo kalba turi būti panaudota, ar koku programavimo metodu ar koncepcija reikia vadovautis ir kodėl.

Galima padaryti išvadą, kad egzistuoja daug skirtingų reikalavimų specifikavimo dokumentų rūšių ir šablonų. Todėl sekančiame skyriuje aptarsime svarbiausius iš jų.

4. Reikalavimų specifikavimo šablonų ir įrankių analizė

4.1. Šablonų analizė

4.1.1. IEEE STD 830-1998

Institute of Electrical and Electronics Engineers (IEEE) (JAV) siūlo vieną populiariausių ir pilniausių programinės įrangos Reikalavimų Specifikavimo šablonų. Jų naujaisias šablonas yra IEEE STD 830-1998 [4], kurio paskutinė versija buvo išleista 1998 metais. Tai yra detalus dokumentas, susidedantis iš 5 dalių. Pirmoji dalis išaiškina šio šablono panaudojimo sferą ir galimybes. Antroji dalis pateikia panaudotos literatūros ir kitų sandartų sąrašą, kurie kontribavo naujausios šablono versijos kūrimui. Trečiasis skyrius išvardina apibrėžimus ir specifinius terminus bei sutrumpinimus, panauduotus dokumente. Ketvirtas skyrius pataria kaip rašyti gerą SRS ir į kokius klausimus šis dokumentas turėtų atsakyti. Penktoji dalis yra SRS skyrių aprašymas su detaliais paaiškinimais kiekvienam skyriui ir poskyriui. Paaiškinimai yra aiškūs ir nedviprasmiški. Viskas aprašyta paprasta kalba, kuri nevargina ypatingais terminais ir nereikalauja aukštų žinių iš vartotojo. IEEE šablonas taip pat turi du priedus. Vienas jų pateikia šablonus kiekvienam SRS skyriui, bei jų skirtingas galimas versijas. Antrasis priedas nurodo kaip galima suderinti šį šabloną su IEEE/EIA 12207.1-1997 standartu. IEEE/EIA 12207.1-1997 standartas nustato bendrą programinės įrangos kūrimo procesų struktūrą (common framework for software life cycle processes). IEEE STD 830-1998 šablono turinys yra pateiktas pav. 2.

1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, acronyms, and abbreviations
1.4 References
1.5 Overview
2. Overall description
2.1 Product perspective
2.2 Product functions
2.3 User characteristics
2.4 Constraints
2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)
Appendixes
Index

Pav. 2: IEEE STD 830-1998 šablono turinys

4.1.2. VOLERE

Volere Requirements Specification Template yra sukūrtas Atlantic Systems Guild Limited Didžiojoje Britanijoje [5]. Paskutintoji versija (Edition 10.1) buvo išleista 2004 metais, todėl Volere galima laikyti naujaisiu reikalavimų specifikavimo šablonu. Pirmasis šablonų variantas pasirodė 1995 metais. Kaip teigia autoriai, Volere yra sukaupta didelė reikalavimų specifikavimo konsultavimo ir tyrimo patirtis. Apie Volere reikalavimų procesą yra parašyta atskira knyga [6]. Volere SRS šablono dokumentą sudaro įvadinis skyrius ir pats šablonas. Įvadinis skyrius pateikia bendrą informaciją apie Volere, supažindina su reikalavimų tipais (funkciniais, nefunkciniais, projekto pribojimais, etc.). Tai pat aptariamas reikalavimų testavimas. Svarbus punktas yra patarimai ir reikalavimų specifikavimo ypatumų aprašymai. Įvadinis skyrius taip pat pateikia sutrumpinimų ir apibrėžimų sąrašą, kurie yra naudojami šablone. Antroji dalis susideda iš detalaus ir pilno reikalavimų specifikavimo

PROJECT DRIVERS
1. The Purpose of the Project
2. Client, Customer and other Stakeholders
3. Users of the Product
PROJECT CONSTRAINTS
4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions
FUNCTIONAL REQUIREMENTS
7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements
NON-FUNCTIONAL REQUIREMENTS
10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements
PROJECT ISSUES
18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Pav. 3: Volere šablono turinys

dokumento šablono. Kiekvienas skyrius, poskyris ir pastraipa yra krupščiai aprašytos ir pateikiami panaudojimo pavyzdžiai. Volere šablono turinys pateiktas pav. 3.

4.1.3. Šablonų palyginimas ir analizės išvados

Abu aukščiau aptarti šablonai suteikia pakia galimybę sukūrti gerą reikalavimų specifikavimo dokumentą. Nors jie ir yra panašūs iš pirmo žvilgsnio, bet turi esminius skirtumus: IEEE siūlo SRS – programinės įrangos reikalavimų specifikavimo šabloną, o Volere siūlo bendrą šabloną, kuris gali būti panauduotas kaip bendrai SS (sistemos reikalavimams), taip ir SRS. Volere autoriai siūlo tiesiog ištrinti tuos skyrius, kurie yra nereikalingi konkrečiame projekte. Volere yra didesnis už IEEE šabloną (62 puslapiai prieš 37). Kūriant reikalavimų specifikavimo automatizavimo sistemą yra svarbu nuspręsti koku reikalavimų specifikavimo šablonu apsiriboti – ar paimti Volere, kaip pilniausią ir universalų šabloną, ar fokusuotis į programinės įrangos reikalavimų specifikavimo šabloną IEEE. Kūriant tokios sistemos prototipą, būtina užtikrinti jos principinį funkcionavimą, ir universalumą, todėl lankstesnis SRS šablonas (šiuo atvjeju Volere) tiktų šiems tikslams labiau. Tačiau net ir Volere standartą reikėtų dar labiau patikslinti, kad jis būtų supaprastintas ir nevalgintų vartotojo užklaomis ir pasirinkimo variantais, bet ir neprarastų savo lankstūmo ir universalumo.

4.2. Įrankių analizė

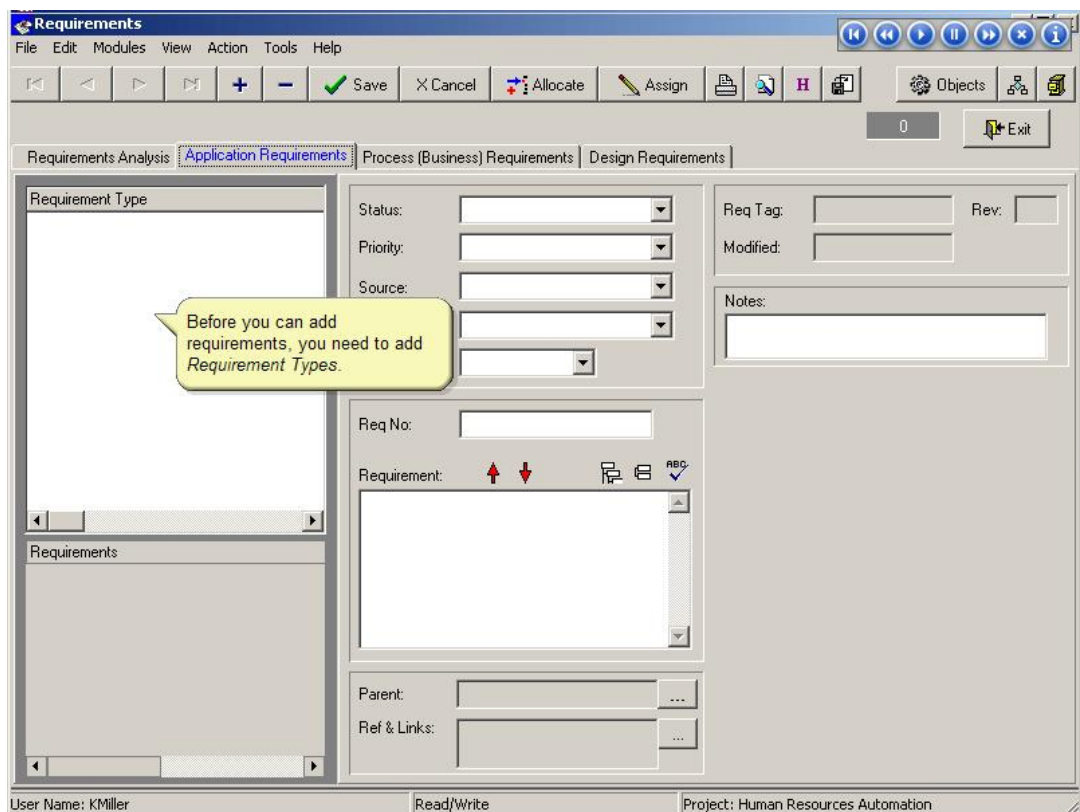
4.2.1. Analyst PRO

Programinę įrangą gaminanti JAV firma “Goda Software” siūlo specializuotą reikalavimų specifikavimo dokumento rašymo ir sistemos projektavimo produktą “AnalystPRO” [7]. Jos savybės yra pateiktos sistemos aprašyme [8] (Pav. 4).

AnalystPRO turi daugybę naudingų savybių. Tarp jų tokios kaip automatinė dokumento versijos kontrolė ir kelių analitikų darbo palaikymas. Tačiau ši sistema nepateikia reikalavimų specifikavimo dokumento šabloną – visus reikalavimus reikia suvesti pačiam vartotojui ir tai žymiai apsunkina darbą. Daug laiko užima sistemos paruošimas pačiam darbui (Pav. 5).

	Analyst Pro Feature	Use
Requirements	<ul style="list-style-type: none"> • Requirements Classification • Unique Identification of Requirements • Requirements Allocation and Linking • Requirements Sharing • Custom Attributes • Automatic Version Control • Requirements Work flow • Requirements Import • Requirements Sorting and Prioritization 	<ul style="list-style-type: none"> • Track user, system requirements to design, testing and maintenance • Share requirements across projects • Revert to any requirement version • Maintain requirements versions without external tools • Assign requirements to team members and instantly communicate with auto email • Customize Analyst Pro project by project, per your own company standards, for continuous internal use across projects • Import requirements from documents • Analyze requirements
Repository	<ul style="list-style-type: none"> • Customizable Structured Storage • Automatic Version Control • File Check In/Check Out • Locking/Unlocking Files • Quick Access to Document Change History • One-Click Open Document 	<ul style="list-style-type: none"> • Quickly locate project files including source code files, reports, flow charts and more • Link repository items with requirements • Share documents/files among team members • Configure manage your files/documents without the set-up complexities
Diagrams	<ul style="list-style-type: none"> • Flow Chart • Network Diagrams 	<ul style="list-style-type: none"> • Create process, organizational diagrams • Specify systems with UML and use case diagrams
Traceability Analysis	<ul style="list-style-type: none"> • Create Traceability Views • Create Traceability Reports • Impact Analysis 	<ul style="list-style-type: none"> • Study the impact of requirements or use case change on design and test cases • Generate traceability reports to identify inconsistencies
Output	<ul style="list-style-type: none"> • Documents • Reports • Export • Graphs 	<ul style="list-style-type: none"> • Generate requirements documents for your customer sign off • Generate graphs for project briefing • Generate requirements change reports • Export to excel for custom requirements analysis
Concurrency	<ul style="list-style-type: none"> • Multi User Support • Records Locking 	<ul style="list-style-type: none"> • Work collaboratively and concurrently on systems development without update conflicts
Access Control	<ul style="list-style-type: none"> • Create User Groups with different set of privileges • Assign users to User Groups 	<ul style="list-style-type: none"> • Control the viewing and modifying of different modules • Manage access control easily within user groups
Customization	<ul style="list-style-type: none"> • Customize Attributes • Customize GUI/Reports 	<ul style="list-style-type: none"> • Customize Analyst Proto your needs, project type and size

Pav. 4: AnalystPRO savybės



Pav. 5: Reikalavimų specifikuojimas AnalystPRO

Kad efektyviai naudoti AnalystPRO reikia tam tikslams specialiai išskyrto projekto laiko, kas visada yra problema esant tankiam darbo grafikui. Tai žymiai atsveria visas teigiamas ir naudingas AnalystPRO savybes.

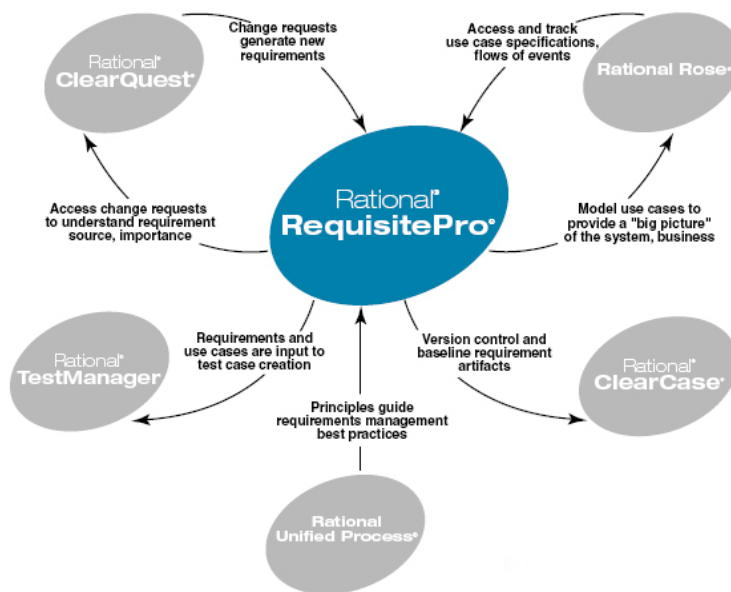
4.2.2. RequisitePRO

Vieną seniausių reikalavimų specifikuojimo produktų siūlo IBM. [13] Tai Rational RequisitePRO. Šis įrankis yra garsiosios Rational šeimos narys, į kurią įeina ir toks žymus UML įrankis kaip Rational Rose. Rational RequisitePRO suteikia galimybes keliems arba kelių sistemos analitikų grupei kooperatyviai dirbti prie reikalavimų specifikuojimo dokumento. Šis įrankis turi visas reikiamas pagrindines savybes: reikalavimų versijos kontrolę, pakeitimų atsekamumą.

Rational RequisitePRO architektūra yra ypač įdomi. Šis įrankis integruosasi į Microsoft Office įrankius Word ir Access, bei naudoja pastaruosius kaip dokumento kūrimo darbo aplinką ir reikalavimų duomenų bazę. Sistema gali taip pat naudoti kitas duomenų bazines: IBM UDB DB2, Microsoft SQL Server, Oracle.

Kooperatyvus darbas užtikrinamas naudojant Rational RequisiteWeb – RequisitePRO Web vartotojo sąsają. Naudojant šią sąsają, vartotojai gali dirbti prie sistemos reikalavimų dokumento (beveik) nepriklausomai nuo darbo aplinkos. Tačiau RequisiteWeb nesuteikia pilno RequisitePRO funkcionalumo. RequisiteWeb reikalauja Microsoft Windows Server.

Rational RequisitePRO yra Rational paketo dalis ir todėl tai yra komponentinė sistema. Kad užtikrinti pilną sistemos funkcionalumą, pageidautina turėti visus kitus komponentus (pav. 40).



Pav. 40: RequisitePRO kaip Rational paketo dalis [13]

4.2.3. Analizės išvados

Šiuo laiku AnalystPRO ir RequisitePRO yra kone vieninteliai programiniai produktai, skirti kūrėti reikalavimų specifikavimo dokumentą. Rinkoje yra daugybė šablonų ir reikalavimų specifikavimo dokumento pavyzdžių, bet jie visi yra tik tekstinio formato dokumentai, o ne programinė įranga.

Paradoksalu, bet AnalystPRO, kaip jau buvo pabrėžta, nesivadovauja nevienu reikalavimų specifikavimo šablonu, bet yra tarsi bendroji platforma bet kuriam šablonui. Toks lankstumas iš privalumo virto silpniausia AnalystPRO vieta. Vartotojas visų pirma turi paruošti programą darbui. Tai reiškia, kad jis pats pasirenka tinkamiausią šabloną ir suveda jį į AnalystPro aplinką.

Rational RequisitePRO yra originalus pagal savo savybes ir architektūrą programinis įrankis, turintis daugybę priavalmų, tačiau kai kurios jo savybės turi svarbių trūkumų. Visų pirma, RequisitePRO reikalauja papildomos Microsoft programinės įrangos, kuri yra gana brangi. Be to, RequisitePRO tiesiog negali tinkamai dirbti be šių papildomų įrankių ir komponentų. Taip pat yra

pageidautyna turėti visus Rational paketo komponentus, kas irgi reikalauja papildomų pastangų ir pinigų. RequisitePRO yra patogus naudojimui integralumo į MS Word aplinką deka, tačiau nėra nepriklausomas nuo operacinės sistemos ir darbo aplinkos. RequisiteWeb yra IBM pastanga padaryti RequisitePRO dalinai nepriklausoma no vartotojo aplinkos, tačiau siūlomas funkcionalumas nėra pilnas ir RequisiteWeb tera tik papildoma RequisitePRO Web vartotojo sąsaja.

Kūriamas reikalavimų specifikavimo rašymo įrankis turėtų pasižymėti gerosiomis apžvelgtų produktų savybėmis ir išvengti jų trūkumų. Reikalavimų specifikavimo sistema turėtų pati siūlyti reikalavimų specifikavimo šabloną, pateikti jo aprašymą ir jau paruoštą reikalavimų pildymo formą. Toks įrankis turėtų būti nepriklausomas nuo vartotojo aplinkos: operacinės sistemos ir darbo vietos, bei nereiklauti papildomų programų.

5. SRS automatizavimo įrankio konceptualus modelis

5.1. SRS automatizavimo įrankio konteksto specifikavimas

Siekiant sukūrti reikalavimų specifikavimo automatizavimo sistemą, yra keli realizavimo variantai. Tai komponentinė sistema ir vieninga sistema.

5.1.1. Komponentinė sistema

Komponentinė sistema gali susidėti iš kelių jau egzistuojančių programų ir sistemų. Pavyzdžiui kaip komunikavimo priemonė gali būti naudojamas Skype Technologies S.A. “Skype” Peer-to-Peer telefonijos produktas, kaip reikalavimų specifikavimo dokumento tvarkyklė Open Office ir kaip failų perdavimo sistema tarp sistemos analitikų Yahoo Groups. Tokia sistema turi daug privalumų. Visų pirmą – visi komponentai yra nemokami. Antrą – tai senos, visiems žinomos sistemos ir sprendimai, kurie nereikalauja papildomo apmokymo. Tačiau visų programų paruošimas darbui taip pat užima daug laiko. Dar vienas neigiamas bruožas – reikia dirbti su daug programų vienu metu ir yra prarandama centralizacija ir, kaip pasekmė, darbo koncentracija. Be to – ne visas reikiamas funkcijas galima realizuoti. Pavyzdžiui sinchroninis darbas prie dokumento ir versijos kontrolė, o jūk tai yra svarbiausios savybės.

5.1.2. Vieninga sistema

Vieninga sistema – tai viena centralizuota programa, kuri automatizuotų reikalavimų programinei įrangai specifikavimo dokumento rašymą. Ji turi užtikrinti pagrindines funkcijas: SRS

šablonas, šablono sinchroninis redagavimas, automatinė dokumento versijos kontrolė, dokumento versijų išsaugojimas. Viena programa palengvins darbą ir nereiklaus daug laiko pačios savybių išaiškinimui ir nustatymui.

5.1.3. Darbo tikslas ir siejami privalumai

Dažnai reikalavimų specifikavimo dokumentas (ir dokumentacija) yra traktuojami, kaip neišvengiama našta ir atidedami kiek galima ilgiau. Kodėl? Visų pirma dėl to, kad niekas nenori skirti tiek daug projekto laiko “beletristikai”. Jūk “tikras” profesionalas gali sukūrti puikią programą ir be aprašymų! Be abėjo tai įmanoma. Dar įmanoma išlošti milijoną loterijoje, bet kam taip rizikuoti? Visų programuotojų, sistemos analitikų ir apskritai programinės įragos kūrėjų pirmutinis tikslas yra sukūrti gerą produktą. Atvirai pasakius visi supranta, kad be gero reikalavimų specifikavimo ir dokumentacijos tai yra labai sunku. Bet laikas... Todėl tiesiog būtina sumažinti reikalavimų specifikavimo kūrimo laiką. Kaip tai padaryti? Reikalingas vienas universalus dokumento šablonas, kuris būtų lengvai suprantamas ir greitai užpildomas. Sunkiausia yra aišku dirbti prie reikalavimų specifikavimo dokumento. Kaip jau pabriežta anksčiau, didelė problema yra užtikrinti suderintą kelių sistemos analitikų darbą prie dokumento. Beveik neįmanoma susodinti juos visus į vieną vietą ir priversti dirbti, nes tikriausiai šie darbuotojai dirba ir prie kitų projektų ir turi skirtingą darbo grafiką. Reikia užtikrinti dokumento versijos aktualumą ir senų versijų suderintą išsaugojimą. Taip pat visi užimti darbuotojai turi žinoti apie dokumente padarytus pasikeitimus. Kaip taip paskirstyti darbą, kad visi dirbtu tikrai prie naujaisios versijos ir jų darbas nepersidengtų? Visos šios problemos apsunkina reikalavimų specifikavimo darbą ir padaro jį kiekvieno projektuotojo galvos skausmu.

Vienintelė šiandien egzistuojanti reikalavimų programinei įrangai specifikavimo automatizuota sistema yra AnalystPRO. Ši sistema yra parašyta skyriuje 4.2.1. Tačiau AnalystPRO reikalauja daug paruošimo prieš pradėdant dirbti. Bet to tai gana sudėtinga programa ir taip pat reikalauja laiko, kad išmokti dirbti su ją. Būtent todėl daugiau paplitę tiesiog tekstinio formato šablonai, kurie yra užpildomi naudojant paprastą tekstinį redaktoriu. Toks darbo būdas yra efektyvus kai vienas sistemos analitikas dirba prie darbo, bet netinkamas kai daug analitikų dirba prie vienos specifikacijos.

Darbo tikslas sukūrti paprastą ir vartotojui aiškią programą, kuri užtikrintų sistemos analitikų grupės darbą prie reikalavimų programinei įrangai specifikavimo dokumento paruošimo. Nutarta, kad sistema bus sukūrtą pagal vieningos sistemos strategiją. Programa turi būti lengvai įdiegiama ir

neturi reikalauti papildomo laiko ir žinių jos vartojimui. Pagrindiniai jos privalumai – SRS rašymo palengvinimas ir vartojimo lengvumas. Sistema turi būti lengvai prieinama iš bet kurio kompiuterio ir operacinės sistemos reikalavimų specifikavimo automatizavimo įrankį. Tokio įrankio prototipas ir yra SRSTemplate.

5.1.4. Kompiuterizuojamos sistemos funkcijos

Reikalavimų programinei įrangai specifikavimo automatizavimo sistema turi užtikrinti šias funkcijas:

SRS šablono parinkimas. Vartotojui pateikiamas Reikalavimų programinei įrangai specifikavimo dokumento šablonas ir, pagal pegeidavimą, jo aprašymas.

SRS šablono užpildymas ir kooperatyvus redagavimas. Turi būti užtikrinta galimybė vienu metu keliems analitikams dirbti prie reikalavimų dokumento.

SRS dokumento versijos kontrolė. Būtina užtikrinti automatinį visų analitikų informavimą apie padarytus pakeitimus dokumente. Visi pakeitimai turi būti atvaizduoti pakeitimų sąrašę. Vartotojui turi būti visada pateikiama tik naujausia dokumento versija. Jeigu pats analitikas tam tikru metu nedalyvauja vykstančiame dokumento tvarkyme ir nori tvarkyti dokumentą asmeniškai, jam turi būti uždrausta tvarkyti šiuo metu tvakomą dokumentą (konkurencijos kontrolė).

SRS dokumento versijų išsaugojimas saugykloje. Visos dokumento versijos turi būti automatiškai išsaugojamos specialioje dokumento versijų saugykloje. Tačiau išsaugojimo “gylis” gali būti nustatomas vartotojo.

5.1.5. Rezultato kokybės kriterijai

Darbo rezultatas turi būti Reikalavimų programinei įrangai specifikavimo automatizavimo sistemos veikiantis prototipas. Prototipas turi realizuoti visas aukščiau nurodytas funkcijas.

Atliekant sistemos projektavimą bus remiamasi šiais kriterijais :

- efektyvumas – kokių kompiuterio resursų reikia sistemai veikti;
- panaudojamumas – ar sistema pakankamai tiksliai interpretuoja vartotojo įvedamus duomenis, ar yra priimtina vartotojui;

- teisingumas – ar sistema pilnai atitinka specifikaciją ir veikia logiškai;
- patikimumas – ar sistema yra tolerantiška klaidoms;

5.2. Reikalavimų specifikacija

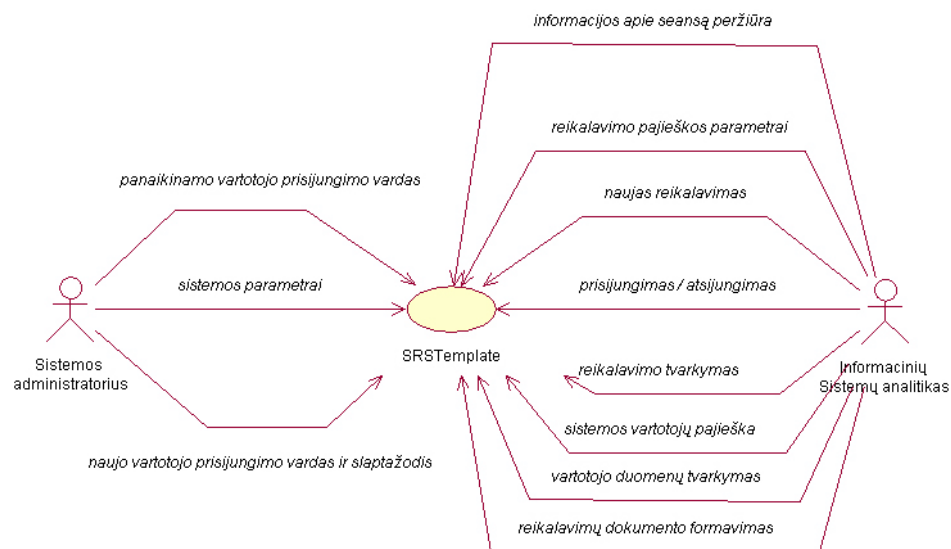
5.2.1. Apribojimai reikalavimams

Sistemos kūrimo eigai ir charakteristikoms keliami tokie apribojimai:

- sistema turi funkcionuoti bet kokios operacinės sistemos pagrindu
- sistema turi veikti nešiojamame kompiuteryje
- sistema neturi reikalauti papildomo įdiegimo kliento kompiuteryje
- sistema turi būti lengvai įdiegiama vartotojų tarnybinėje stotyje
- sistema neturi reikalauti papildomų komponentų vartotojų tarnybinėje stotyje (serveryje)

5.2.2. Veiklos konteksto diagrama

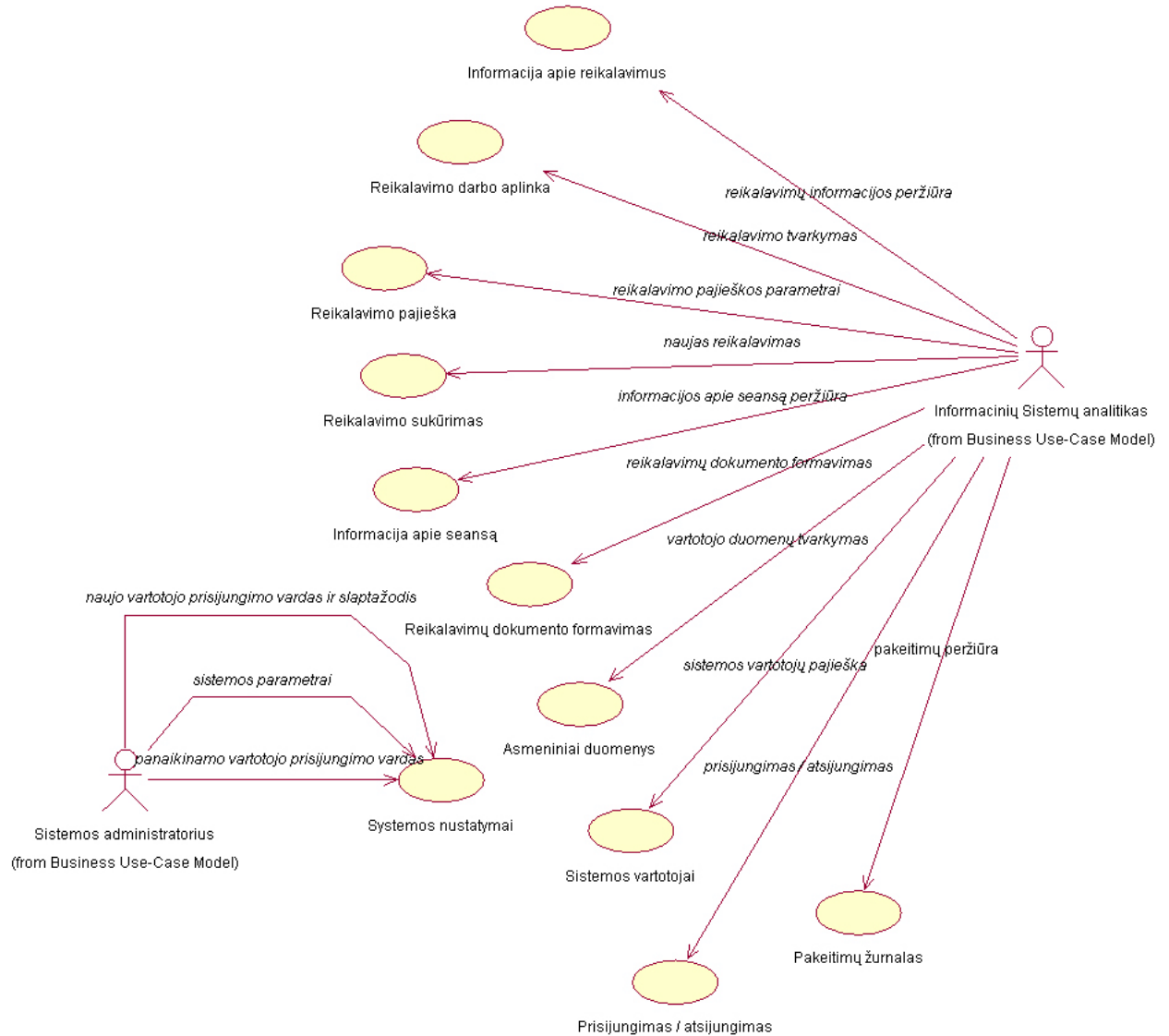
Pateikiama kuriamos sistemos veiklos konteksto diagrama (pav. 32), kuri apibrėžia šios sistemos ribas, išorines esybes, kurios bendrauja su sistema, bei pagrindinius informacijos srautus tarp sistemos ir išorinių esybių.



Pav. 32: SRSTemplate konteksto diagrama

5.2.3. Panaudojimo atvejų diagrama

Veiklos konteksto diagrama parodo produkto sąsają su išorę. Žemiau yra pateikiama SRSTemplate panaudojimo atvejų diagrama, kuri apsprendžia esmines sistemos funkcijas ir jų sąveika su aplinka ir vartotojais. Panaudojimo atvejų diagrama yra parodyta pav. 33.



Pav. 33: SRSTemplate panaudojimo atvejų diagrama

5.2.4. Panaudojimo atvejų sąrašas

Skyriuje 5.2.3. parodyta SRSTemplate panaudojimo atvejų diagrama. Šios diagramos panaudojimo atvejų sąrašas yra pateikiamas žemiau.

1. Panaudojimo atvejis: Sistemos nustatymai

Aktorius: Sistemos administratorius

Aprašymas: SRSTemplate darbo aplinkos parametrų nustatymas (pvz. Reikalavimų versijų gylio parametro nustatymas). Naujo vartotojo įterpimas. Sistemos vartotojo panaikinimas. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 58).

2. Panaudojimo atvejis: Reikalavimo paieška

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Reikalavimo paieška pagal nurodytus pajiškos raktus ir jų kombinaciją. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 58).

3. Panaudojimo atvejis: Reikalavimo sukūrimas

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Naujo reikalavimo sukūrimas. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 59).

4. Panaudojimo atvejis: Reikalavimų dokumento formavimas

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Reikalavimų specifikacijos dokumento formavimas pagal nurodytą dokumento formatą (tik naujausios reikalavimų versijos, visos reikalavimų versijos). Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 59).

5. Panaudojimo atvejis: Pakeitimų žurnalas

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Pakeitimų reikalavimų specifikacijos dokumente peržiūra. Žurnalas parodo vartotojui visus pakeitimus ir užfiksuoja jų atlikimo laiką ir autorių. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 59).

6. Panaudojimo atvejis: Informacija apie reikalavimus

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Pateikiama informacija apie Volere reikalavimų šabloną . Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 60).

7. Panaudojimo atvejis: Sistemos vartotojai

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Pateikiama informacija apie sistemos vartotojus. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 60).

8. Panaudojimo atvejis: Asmeniniai duomenys

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Asmeninių vartotojo duomenų peržiūra ir tvarkymas. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 60).

9. Panaudojimo atvejis: Informacija apie seansą

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Informacijoje apie seansą pateikiama informacija apie vartotoją, kuris šiuo metu yra prisijungęs prie sistemos ir pateikiamas sąrašas kitų sistemos vartotojų, šiuo metu dirbančių prie sistemos. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 61).

10. Panaudojimo atvejis: Prisijungimas / atsijungimas

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Vartotojo prisijungimas prie sistemos ir atsijungimas nuo sistemos. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 61).

11. Panaudojimo atvejis: Reikalavimo darbo aplinka

Aktorius: Informacinių sistemų analitikas / projektuotojas

Aprašymas: Darbo su reikalavimu procesai: reikalavimų koregavimas, panaikinimas, atspausdinimas. Panaudojimo atvejo funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl.62).

5.2.5. Funkciniai reikalavimai

Reikalavimai reikalavimų specifikavimo dokumento sudarymui:

Reikalavimas 1: Sistema turi dirbti pagal Volere reikalavimų specifikavimo šabloną.

Reikalavimas 2: Turi būti užtikrinta informacijos apie šablonus peržiūra.

Reikalavimas 3: Sistema turi leisti sukūrti naują reikalavimą

Reikalavimas 4: Sistema turi užtikrinti reikalavimo versijų išsaugojimą.

Reikalavimas 5: Reikalavimų versijų gylis turi būti reguliojamas.

Reikalavimas 6: Sistema turi užtikrinti reikalavimų pajiešką pagal raktažodžius.

Reikalavimas 7: Sistema turi užtikrinti kiekvieno reikalavimo atspausdinimo galimybę.

Reikalavimas 8: Sistema turi sudaryti reikalavimų specifikavimo dokumentą ir leisti jį atspausdinti / išsaugoti.

Reikalavimai kelių analitikų darbo prie reikalavimų specifikavimo dokumento užtikrinimui:

Reikalavimas 9: Sistema turi leisti dirbti prie reikalavimų sudarymo dokumento iškart keliems analitikams.

Reikalavimas 10: Sistema turi užtikrinti konkurencijos kontrolę, t.y. organizuoti kelių analitikų darbą taip, kad jų darbas nepersidengtų.

Reikalavimas 11: Sistema turi užtikrinti, kad kiekvienas vartotojas turi išsamią informaciją apie tai kas šiuo metu vyksta sistemoje (kas yra šiuo metu prisijungęs ir kas ką daro).

Reikalavimas 12: Konkurencijos kontrolė turi būti užtikrinta užrakto būdu.

Reikalavimai darbui su sistema:

Reikalavimas 13: Sistemos administratorius turi turėti galimybę keisti versijų išsaugojimo gylį.

Reikalavimas 14: Sistemos administratorius turi turėti galimybę sukūrti naują sistemos vartotoją.

Reikalavimas 15: Sistemos administratorius turi turėti galimybę panaikinti sistemos vartotoją.

Reikalavimas 16: Kiekvienas sistemos vartotojas turi turėti galimybę tvarkyti savo asmeninius duomenis.

Reikalavimas 17: Turi būti pateikiama ir prieinama informacija apie visus sistemos vartotojus.

Reikalavimas 18: Turi būti parodoma kas iš sistemos vartotojų šiuo metu yra prisijungęs prie sistemos.

Reikalavimai duomenims:

Reikalavimas 19: Turi būti naudojama MySQL duomenų bazė. Duomenų saugyklos schema parodyta skyriuje 5.5.9. pav. 31.

5.2.6. Nefunkciniai reikalavimai

Reikalavimai sistemos išvaizdai

- Turi būti naudojama grafinė vartotojo sąsaja.
- Sąsaja turi būti lengvai skaitoma, suprantama.

Reikalavimai panaudojamumui

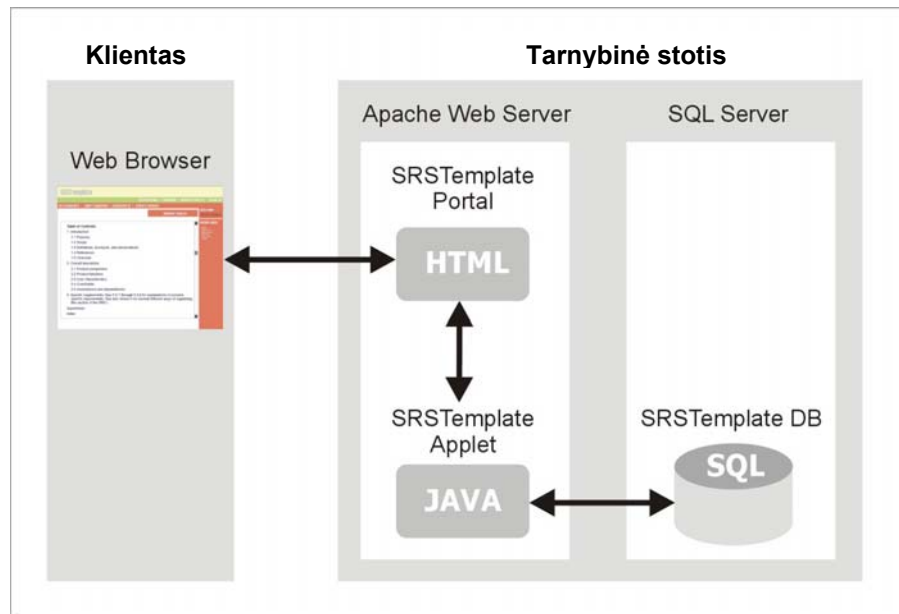
- Paprasta naudotis IS projektuotojams- analitikams.
- Turi būti lengvai išmokstama dirbti su sistema.
- Sistema turi funkcionuoti bet kokios operacinės sistemos pagrindu
- Sistema turi veikti nešiojamame kompiuteryje
- Sistema neturi reikalauti papildomo įdiegimo kliento kompiuteryje
- Sistema turi būti lengvai įdiegiama vartotojų tarnybinėje stotyje
- Sistema neturi reikalauti papildomų komponentų vartotojų tarnybinėje stotyje (serveryje)

Reikalavimai vykdymo charakteristikoms

- Interaktyvaus bendravimo su vartotoju metu programa turi veikti pakankamai greitai, užtikrinti efektyvų darbą.

5.3. Apibendrintas architektūros modelis

SRSTemplate sistema turi būti lengvai įdėgiama ir neturi kelti daug reikalavimų vartotojui. Todėl architektūros modelis turi būti nesudėtingas. Nutarta naudoti Apache Web Serverį ir SQL duomenų bazių serverį. Sistema bus parašyta JAVA programavimo kalba, įdėgiant sistemos Java Applet'ą į sistemos portalą, parašytą HTML ir CSS. Bendras architektūros modelis yra parodytas pav. 6.



Pav. 6: Bendroji sistemos architektūra

5.4. Šablono modelio parinkimas

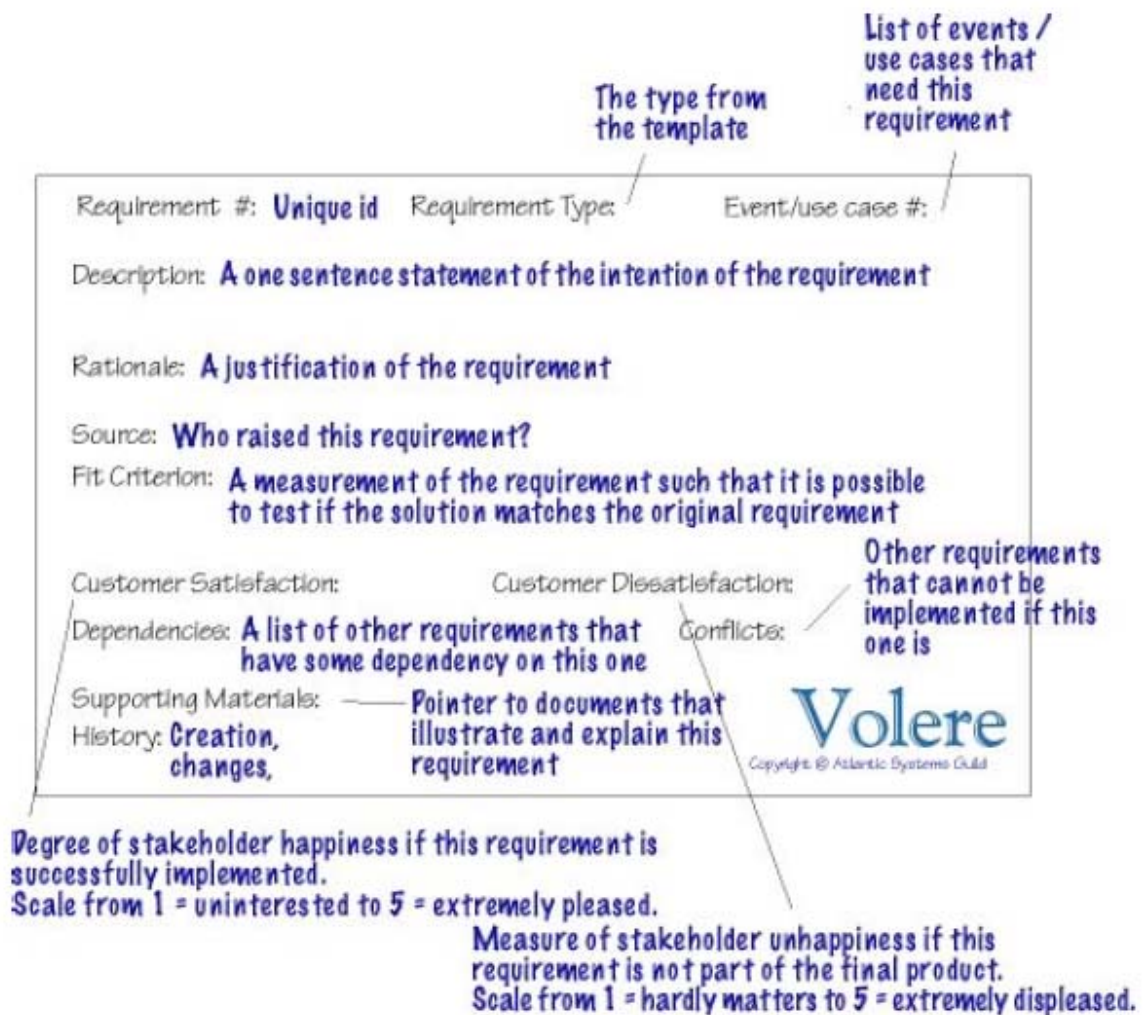
Egzistuoja du alternatyvūs šablonai: Volere ir IEEE STD 830-1998. Šio darbo 4.1. skyriuje yra aptartos šių šablonų pagrindinės savybės ir ypatumai. Prie aprašytų savybių galima pridėti ir tai, kad Volere šablonas yra ne tik pilnesnis, bet ir naujesnis IEEE atžvilgiu. Be to Volere šablonas pateikia išsamesnį reikalavimų aprašymą. Todėl galutinai nutarta vadovautis Volere šablonu kūriant reikalavimų specifikavimo šablono įrankio prototipą. Šį prototipą toliau sutrumpintai vadinsime SRSTemplate.

5.5. Duomenų struktūros

Volere šablonas turi daugybę skirtingų reikalavimų: sistemos interfeisai, vartotojo interfeisai, adaptavimo reikalavimai, reikalavimai duomenų bazei ir daugybė kitų. Kiekvinas toks atskiras reikalavimų tipas turi savo struktūrą arba formą. Tačiau Volere šablonas pateikia daugumos reikalavimų formų unifikavimo strategiją.

5.5.1. Volere siūloma reikalavimų formų unifikavimo strategija

Volere siūlo priimti vieningą reikalavimų formatą bent jau “atominiams” reikalavimams, t.y. funkciniais ir nefunkciniais reikalavimams. Šis formatas yra parodytas pav. 7 [5]. Pagal šią formą išskiriami pagrindiniai reikalavimų laukai.



Pav. 7: Volere reikalavimų formų unifikuojimo strategija

Requirement # - Unikalus reikalavimo atpažinimo numeris.

Requirement Type – Reikalavimo tipo numeris pagal Volere reikalavimų šabloną

Event / use case # - Panaudojimo atvejų sąrašas, kurie naudoja šį reikalavimą

Description – Trumpas reikalavimo aprašymas, išreiškiantis pagrindinę reikalavimo idėją.

Rationale – Išsamus reikalavimo aprašymas.

Source – Reikalavimo autorius.

Fit criterion - Laukas, nurodantis kiek realizuoto programinio produkto funkcionalumas atitinka šį reikalavimą.

Customer Satisfaction – Užsakovo pasitenkinimo kriterijus, jei šis reikalavimas bus išpildytas. Šis laukas parodo kiek svarbus yra šis reikalavimas užsakovui.

Customer Dissatisfaction – Šis laukas yra tiesiogiai susijęs su *Customer Satisfaction* lauku ir nurodo užsakovo nepasitenkinimo kriterijų, kai šis reikalavimas nebus išpildytas.

Dependencies – Čia yra nurodyti visi kiti reikalavimai, kurie tiesiogiai susiję, paveikia, ar priklauso nuo šito reikalavimo.

Conflicts – Prieštarų konkrečiam reikalavimui reikalavimų sąrašas.

Supporting Materials – Sąrašas šaltinių ar materialų, kurie padeda geriau suprasti ar išspręsti šį konkretų reikalavimą.

5.5.2. Volere šablono padalinimas pagal reikalavimų formatus.

Remiantis Volere nurodyta reikalavimų formų unifikavimo strategija, daugumą reikalavimų galima priskirti prie skyriuje 5.4.1. aprašytos formos. Tačiau visi šablono reikalavimai vis dėl to negali jos atitikti. Būtina padalinti Volere šablono reikalavimus ir surinkti juos į kelias bendras reikalavimų formas.

Paveiklė 3 yra norodytas bendras Volere šablono turinys. Pilnas Volere reikalavimų sąrašas yra pateiktas žemiau [5] [14]:

1 The Purpose of the Project (**Projekto paskirtis**)

1a. The user problem or background of the project effort. (**Vartotojo problema, arba projekto kūrimo pagrindimas**)

1b. Goals of the project. (**Projekto tikslas**)

2 Client, Customer and other Stakeholders (**Užsakovai, prikėjai ir kiti sistema suinteresuoti asmenys**)

2a. The client is the person/s paying for the development, and owner of the delivered product. (**Užsakovas**)

2b. The customer is the person/s who will buy the product. (**Pirkėjai**)

2c. Other stakeholders (**Kiti sprendimus priimančios asmenys**)

3 Users of the Product (**Vartotojai**)

3a. The hands-on users of the product (**Potencialūs vartotojai**)

3b. The priorities assigned to users (**Vartotojai pagal prioritetus**)

3c. User participation (**Vartotojai pagal dalyvavimą projekto kūrime**)

3d. Maintenance users (**Specifinius reikalavimus sistemai turintys potencialūs vartotojai**)

4 Mandated Constraints (**Apribojimai reikalavimams**)

4a. Solution design constraints (**Apribojimai sprendimui**)

4b. Implementation environment of the current system (**Diegimo aplinka**)

4c. Partner or collaborative applications (**Bendradarbiaujančios sistemos**)

4d. Off-the-shelf software (**Komerciniai specializuoti programų paketai**)

4e. Anticipated workplace environment (**Numatoma darbo vietos aplinka**)

4f. How long do the developers have for the project? (**Sistemos kūrimo terminai**)

4g. What is the financial budget for the project? (**Sistemos kūrimo biudžetas**)

5 Naming Conventions and Definitions (**Terminų žodynas**)

6 Relevant Facts and Assumptions (**Svarbūs faktai**)

6a. Factors that have an effect on the product, but are not mandated requirements constraints. (**Faktai, turintys įtaką produktui, bet neįtraukti į sistemos apribojimus**)

6b. Assumptions that the team is making about the project (**Projekto vykdytojų pasiūlymai ir prielaidos apie projektą**)

7 The Scope of the Work (**Veiklos sudėtis**)

7a. The current situation (**Egzituojančių veiklos procesų analizė**)

7b. The context of the work. (**Veiklos kontekstas / konteksto diagrama**)

7c. Work partitioning (List of business events) (**Veiklos padalinimas / įvykių sąrašas**)

8 The Scope of the Product (**Sistemos sudėtis**)

8a Product Boundary (**Sistemos ribos / panaudojimo atvejų diagrama**)

8b Product use case list (List of product use cases) (**Sistemos panaudojimo atvejų sąrašas**)

9 Functional and Data Requirements (**Funkciniai reikalavimai ir reikalavimia duomenims**)

9a. Functional Requirements. (**Funkciniai reikalavimai**)

9b. Data requirements. (**Reikalavimai duomenims**)

10 Look and Feel Requirements (**Reikalavimai sistemos išvaizdai**)

10a. The interface (**Reikalavimai vartotojo sąsajai**)

10b. The style of the product (**Produkto stilius**)

11 Usability and Humanity Requirements (**Reikalavimai panaudojamumui**)

11a. Ease of use. (**Naudojimumas**)

11b. Personalization and internationalization requirements Content (**Sistemos adaptyvumo laipsnis**)

11c. Ease of learning. (**Išmokstamumas**)

11d. Understandability and Politeness requirements. (**Suprantamumas**)

11e. Accessibility requirements. (**Pasiekiamumas**)

12 Performance Requirements (**Reikalavimai vykdymo charakteristikoms**)

12a. Speed and latency requirements (**Reikalavimai veikimo greičiui ir vėlinimui**)

12b. Safety critical requirements (**Kritiniai saugumo reikalavimai**)

12c. Precision or accuracy requirements (**Reikalavimai tikslumui**)

12d. Reliability and Availability requirements (**Patikimumo reikalavimai**)

12e. Robustness or fault tolerance requirements (**Reikalavimia klaidų toleravimui**)

12f. Capacity requirements (**Apimties reikalavimai**)

12g. Scalability or extensibility requirements (**Skalabilumo / išplečiamumo reikalavimai**)

12h. Longevity requirements (**Reikalavimai tęstinumui**)

13 Operational Requirements (**Reikalavimai veikimo sąlygoms**)

13a. Expected physical environment (**Fizinė aplinka**)

13b. Expected technological environment (**Techninė aplinka**)

13c. Partner applications (**Partnerinė / sąveikaujanti programinė įranga**)

13d. Productization requirements (**Reikalavimai sistemos pateikimui / pristatymui**)

14 Maintainability and Support Requirements (**Reikalavimai sistemos priežiūrai**)

14a. Maintenance requirements (**Reikalavimai sistemos priežiūrai**)

14b. Special conditions that apply to the maintenance of the product (**specialios priežiūros sąlygos**)

14c. Supportability requirements (**Palaikamumo reikalavimai**)

14d. Adaptability requirements (**Adaptyvumo reikalavimai**)

14e. Installation requirements (**Reikalavimai sistemos diegimui**)

15 Security Requirements (**Reikalavimai saugumui**)

15a. Access requirements (**Prieinamumo reikalavimai**)

15b. Integrity requirements (**Integralumo reikalavimai**)

15c. Privacy requirements (**Privatumo reikalavimai**)

15d. Audit requirements (**Audito reikalavimai**)

15e. Immunity requirements (**Atsparumo reikalavimai**)

16 Cultural and Political Requirements (**Kultūriniai-politiniai reikalavimai**)

16a. Cultural requirements (**Kultūriniai reikalavimai**)

16b. Political requirements (**Politiniai reikalavimai**)

17 Legal Requirements (**Teisiniai reikalavimai**)

17a. Compliance requirements (**Teisiniai reikalavimai**)

17b. Standards requirements (**Reikalavimai standartams**)

18 Open Issues (**Atviri klausimai /Neužbaigti reikalavimai**)

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

19a. Is there a ready-made product that could be bought? (**Ar yra jau pagamintų sistemų kurios gali būti nupirktos?**)

19b. Can ready-made components be used for this product? (**Ar yra jau pagamintų komponentų, kuriuos galima būti panaudoti?**)

19c. Is there something that we could copy? (**Ar galima ką nors nukopijuoti?**)

20 New Problems (Naujos problemos)

20a. What problems could the new product cause in the current environment? (**Kokias problemas nauja sistema sukels jos diegimo aplinkai?**)

20b. Will the new development affect any of the installed system? (**Ar nauja sistema paveiks jau įdiegtas sistemas?**)

20c. Will any of our existing users be adversely affected by the new development? (**Ar kas nors iš esamų vartotojų gali neigiamai nusiteikti prieš naują sistemą?**)

20d. What limitations exist in the anticipated (**Kokie apribojimai, galintys kliudyti naujai sistemai, egzistuoja numatomoje diegimo aplinkoje?**)

20e. Will the new product create other problems? (**Ar nauja sistema sukels kokių nors kitų problemų?**)

21 Tasks (Uždaviniai)

21a. What steps have to be taken to deliver the system? (**Kokie žingsniai reikalingi sistemai pateikti?**)

21b. Development phases (**Vystymo etapai**)

22 Cutover (Pritaikymas)

22a. What special requirements do we have to get the existing data, and procedures to work for the new system? (**Kokius specialius reikalavimus turime esamiems duomenims “paimti” bei procedūroms pritaikyti darbui su nauja sistema?**)

22b. What data has to be modified/translated for the new system? (**Kokie duomenys turės būti transformuoti, perkelti į naują sistemą?**)

23 Risks (Rizikos įvertinimas)

24 Costs (Kaina)

25 User Documentation and Training (Vartotojo dokumentacija ir apmokymas)

25a. The plan for building the user documentation. (**Vartotojo dokumentacijos kūrimo planas**)

26 Waiting Room (Perspektyviniai reikalavimai)

27 Ideas for Solutions (Idėjos ir sprendimai)

Remiantis pilnais Volere šablono reikalavimų turinių aprašymais [6] ir atlikus išsamią jų analizę pagal laukų atitikimą ir išskyrimą į apibendrintas formas, nutarta padalinti Volere reikalavimus į 3 pagrindinius ir 2 papildomus formatus. Šie formatai pateikti lentelėje pav. 8.

Formato tipas	Formato pavadinimas	Reikalavimai, atitinkantys šį formatą
1	Requirement Shell	1a, 1b, 6, 7a, 7b, 8a, 18, 19, 20, 21, 22, 23, 24, 25, 27, 4, 9, 10, 11, 12, 13, 14, 15, 16, 17, 26
2	Event / Use Case List	7c, 8b
3	User / Stakeholder Profile	2a, 2b, 2c, 3 (reikalavimai 3a-d apjungti į vieną)
4	Illustrations	Visi reikalavimų paveikslai
5	Glossary	5

Pav. 8: Volere reikalavimų išdėstymas pagal formatus

Volere šablonai, išdėstyti pagal formatus, turės unikalų ID numerį atitinkamoje formoje. Todėl siekiant užtikrinti kiekvieno reikalavimo unikalų numeraciją visoje dokumentacijoje, parinktas unikalus sudėtinis raktas `rq_id` (reikalavimo ID formoje) + `rq_type` (reikalavimo tipas pagal reikalavimų tipų lentelę, kuri bus aprašyta skyriuje 5.4.8.1.).

5.5.3. Duomenų struktūra 'Requirement Shell'

Field	Type
<code>rq_id</code>	int(10)
<code>rq_type</code>	int(11)
<code>use_case_event</code>	int(11)
<code>user_class</code>	int(11)
<code>description</code>	longtext
<code>rationale</code>	longtext
<code>source</code>	int(11)
<code>fit_criterion</code>	int(11)
<code>customer_satisfaction</code>	int(11)
<code>customer_dissatisfaction</code>	int(11)
<code>dependencies</code>	text
<code>conflicts</code>	text
<code>supporting_materials</code>	text
<code>version</code>	int(11)
<code>illustration</code>	text

Pav. 9: 'RequirementShell' duomenų struktūra

Requirement Shell forma realizuojama pagrindinėje sistemos MySQL duomenų bazės lentelėje 'RequirementShell'. Šioje lentelėje yra išsaugoma informacija apie pagrindinius (atominus) reikalavimų specifikavimo šablono reikalavimus. Lentelės struktūra parodyta pav. 9. Lentelės užpildymo pavidzys atvaizduotas pav. 10.

Laukas	Reikšmė
<code>rq_id</code>	1
<code>rq_type</code>	21
<code>use_case_event</code>	1
<code>user_class</code>	1
<code>description</code>	Close the elevator firewall if fire in turret.
<code>rationale</code>	In case of heavy turret damage and fire all elevator doors must be quickly closed to prevent turret arsenal explosion.
<code>source</code>	1
<code>fit_criterion</code>	5
<code>customer_satisfaction</code>	5
<code>customer_dissatisfaction</code>	5
<code>dependencies</code>	2, 3
<code>conflicts</code>	
<code>supporting_materials</code>	H.M.S. Lion Q Turret examination report
<code>version</code>	4
<code>illustration</code>	1

Pav. 10: 'RequirementShell' užpildymas

5.5.4. Duomenų struktūra ‘Event / Use Case List’

Event / Use Case List forma realizuojama lentelėje ‘EventUseCaseList’. Ši lentelė saugoja informaciją apie visus projektuojamos programinės įrangos panaudojimo atvejus. Lentelės struktūra parodyta pav. 11. Lentelės užpildymo pavyzdys atvaizduotas pav. 12.

Field	Type
<u>euc_id</u>	int(10)
rq_type	int(10)
euc_name	text
euc_description	longtext
illustration	int(10)
source	int(10)

Pav. 11:
‘EventUseCaseList’
duomenų struktūra

Laukas	Reikšmė
euc_id	1
rq_type	28
euc_name	Main turret operating system use case diagram
euc_description	This is the main use case diagram for main turret operating system
illustration	1
source	1

Pav. 12: ‘EventUseCaseList’ užpildymas

5.5.5. Duomenų struktūra ‘User / Stakeholder Profile’

User / Stakeholder Profile atvaizduota lentelėje ‘UserStakeholderProfile’. Ši lentelė aprašo visus projektuojamos sistemos vartotojus ir panaudojimo atvejų aktorius. Lentelės struktūra parodyta pav. 13. Lentelės užpildymo pavyzdys atvaizduotas pav. 14.

Field	Type
<u>person_id</u>	int(10)
rq_type	int(10)
person_name	text
person_surname	text
person_description	longtext
person_priority	int(10)
person_participation	int(10)
source	int(10)

Pav. 13:
‘UserStakeholderProfile’
duomenų struktūra

Laukas	Reikšmė
person_id	1
rq_type	5
person_name	Main turret Officer-In-Command
person_surname	
person_description	Main turret Officer-In-Command. Command officer of the main turret
person_priority	5
person_participation	5
source	1

Pav. 14: ‘UserStakeholderProfile’ užpildymas

5.5.6. Duomenų struktūra ‘Illustrations’

Ši speciali forma yra viena iš pagrindinių formų, tačiau jos funkcija yra daugiau pagalbinė. Ši forma išsaugo visas iliustracijas, kurios sutinkamos dokumentacijoje. Šią formą atspindi lentelė ‘Illustrations’. Lentelės struktūra parodyta pav. 15. Lentelės užpildymo pavidzys atvaizduotas pav. 16.

Field	Type
<u>pic_id</u>	int(10)
rq_type	int(10)
pic_description	text
picture	binary(1)
source	int(10)

Pav. 15: ‘Illustrations’ duomenų struktūra

Laukas	Reikšmė
pic_id	1
rq_type	5
pic_description	Main turret operating system use case diagram
picture	101010110010110101010011001101001010100100...
source	1

Pav. 16: ‘Illustrations’ užpildymas

5.5.7. Duomenų struktūra ‘Glossary’

Glossary yra speciali forma tik reikalavimų specifikavimo dokumento žodyno turinio aprašymui ir užpildymui. Šią formą reprezentuoja lentelė ‘Glossary’. Lentelės struktūra parodyta pav. 17. Lentelės užpildymo pavidzys atvaizduotas pav. 18.

Field	Type
<u>word_id</u>	int(10)
rq_type	int(10)
word_title	text
word_description	longtext
source	int(10)

Pav. 17: ‘Glossary’ duomenų struktūra

Laukas	Reikšmė
word_id	1
rq_type	13
word_title	firewall
word_description	Special fire-proof doors in elevators to prevent fire spreading inside the turret arsenal
source	1

Pav. 18: ‘Glossary’ užpildymas

5.5.8. Pagalbinės sistemos duomenų struktūros

5.5.8.1. Duomenų struktūra ‘Template Table’

Template Table yra viena pagrindinių pagalbinių sistemos duomenų struktūrų. Ji atsako už duomenų apie reikalavimus išsaugojimą. Atitinkama lentelė sistemos MySQL duomenų bazėje yra ‘TemplateTable’. Lentelės laukais ‘shell_type’ nurodo kurį šabloną naudoti šiam reikalavimui: 1 (RequirementShell), 2 (EventUseCaseList), 3 (UserStakeholderProfile), 4 (Illustrations), 5(Glossary). Lentelės struktūra parodyta pav. 19. Lentelės užpildymo pavyzdys atvaizduotas pav. 20.

Field	Type
req_type	int(10)
req_number	text
req_title	text
req_info	longtext
shell_type	int(10)

Pav. 19:
‘TemplateTable’
duomenų struktūra

Laukas	Reikšmė
req_type	21
req_number	9a
req_title	Functional and Data Requirements. Functional Requirements
req_info	A specification for each individual functional requirement. ...
shell_type	1

Pav. 20: ‘TemplateTable’ užpildymas

5.5.8.2. Duomenų struktūra ‘System Users’

Field	Type
user_id	int(11)
user_status	int(11)
user_name	text
user_surname	text
email	text
f_phone	text
m_phone	text
icq	text
login	text
password	text

Pav. 21: ‘SystemUsers’
duomenų struktūra

System Users atsako už informaciją apie visus reikalavimų specifikavimo automatizavimo sistemos vartotojus. Šie vartotojai yra įvairių profilių specialistai, kurie rašo reikalavimų specifikavimo dokumentą. Šią formą atvaizduoja lentelė ‘SystemUsers’. Lentelės struktūra parodyta pav. 21. Lentelės užpildymo pavyzdys atvaizduotas pav. 22.

Laukas	Reikšmė
user_id	1
user_status	0 (1 – “logged in”, 0 – “logged out”)
user_name	Michail
user_surname	Mazo
email	mazo@one.lt
f_phone	555 555 555
m_phone	888 888 888
icq	333-664-585
login	mozas
password	mazo

Pav. 22: ‘SystemUsers’ užpildymas

5.5.8.3. Duomenų struktūra ,Change Log’

Change Log atsakingas už visų pakeitimų, kurie vyksta reikalavimų specifikavimo dokumente, fiksavimą. Tai gali būti reikalavimo sukūrimas, redagavimas ar panaikinimas. Change Log atitinka lentelė ‘ChangeLog’. Lentelės struktūra parodyta pav. 23. Lentelės užpildymo pavyzdys atvaizduotas pav. 24.

Field	Type
<u>change_id</u>	int(10)
date	text
time	text
req_id	int(10)
req_type	int(10)
req_title	text
info	text
user_login	text

Pav. 23: ‘ChangeLog’ duomenų struktūra

Laukas	Reikšmė
change_id	1
date	05.12.2005
time	14:37
req_id	1
req_type	21
req_title	Close the elevator firewall if fire in turret.
info	modified
user_login	mozas

Pav. 24: ‘ChangeLog’ užpildymas

5.5.8.4. Duomenų struktūra ,Modify List’

Modify List yra svarbi pagalbinė lentelė ‘ModifyList’, kurios funkcija yra užtikrinti reikalavimo prieinamumo pakeitimams tvarkaraštį. ‘ModifyList’ laikinai išsaugo informaciją apie kuris reikalavimas yra šiuo metu modifikuojamas. Lentelė taip pat išsaugo informacija apie vartotoją, kuris šią modifikaciją atlieka. Kitam vartotojui norint modifikuoti tą patį reikalavimą, sistema paduoda užklausa į šią lentelę ir jeigu pageidaujamo reikalavimo req_id + req_type yra

lentelėje, šio reikalavimo modifikuoti neleidžiama. Taip realizuojama paprasčiausia, bet patikimiausia prieinamumo kontrolė “užrakto” metodu [9] (plačiau 5.5.10). Lentelės struktūra parodyta pav. 25. Lentelės užpildymo pavyzdys atvaizduotas pav. 26.

Field	Type
mod_id	int(10)
req_id	int(10)
req_type	int(10)
system_user	text

Pav. 25: ‘ModifyList’ duomenų struktūra

Laukas	Reikšmė
mod_id	4
req_id	1
req_type	21
system_user	mozas

Pav. 26: ‘ModifyList’ užpildymas

5.5.8.5. Duomenų struktūra ,System Settings’

System Settings nustato reikalavimų specifikavimo sistemos darbo parametrus. System Settings atitinka lentelė “SystemSettings”. Lentelės vienintelis duomenų laukas ‘re_versions’ atsako už reikalavimo versijų skaičiaus gylio nustatymą. Lentelės struktūra parodyta pav. 27. Lentelės užpildymo pavyzdys atvaizduotas pav. 28.

Field	Type
id	int(10)
re_versions	int(11)

Pav. 27: “SystemSettings” duomenų struktūra

Laukas	Reikšmė
id	1
re_versions	5

Pav. 28: “SystemSettings” užpildymas

5.5.8.6. Duomenų struktūra ,Requirement Shell Versions’

Requirement Shell Versions turi specialiąją paskirtį. Atitinkamoje lentelėje ‘RequirementShellVersions’ yra išsaugomos visos turimos atominų reikalavimų versijos. Kiekvieno reikalavimo versijų skaičius priklauso nuo versijų skaičiaus gylio (5.4.8.5.). Lentelės struktūra parodyta pav. 29. Lentelės užpildymo pavyzdys atvaizduotas pav. 30.

Field	Type
version_id	int(10)
rq_id	int(10)
rq_type	int(11)
use_case_event	int(11)
user_class	int(11)
description	longtext
rationale	longtext
source	int(11)
fit_criterion	int(11)
customer_satisfaction	int(11)
customer_dissatisfaction	int(11)
dependencies	text
conflicts	text
supporting_materials	text
version	int(11)
illustration	text

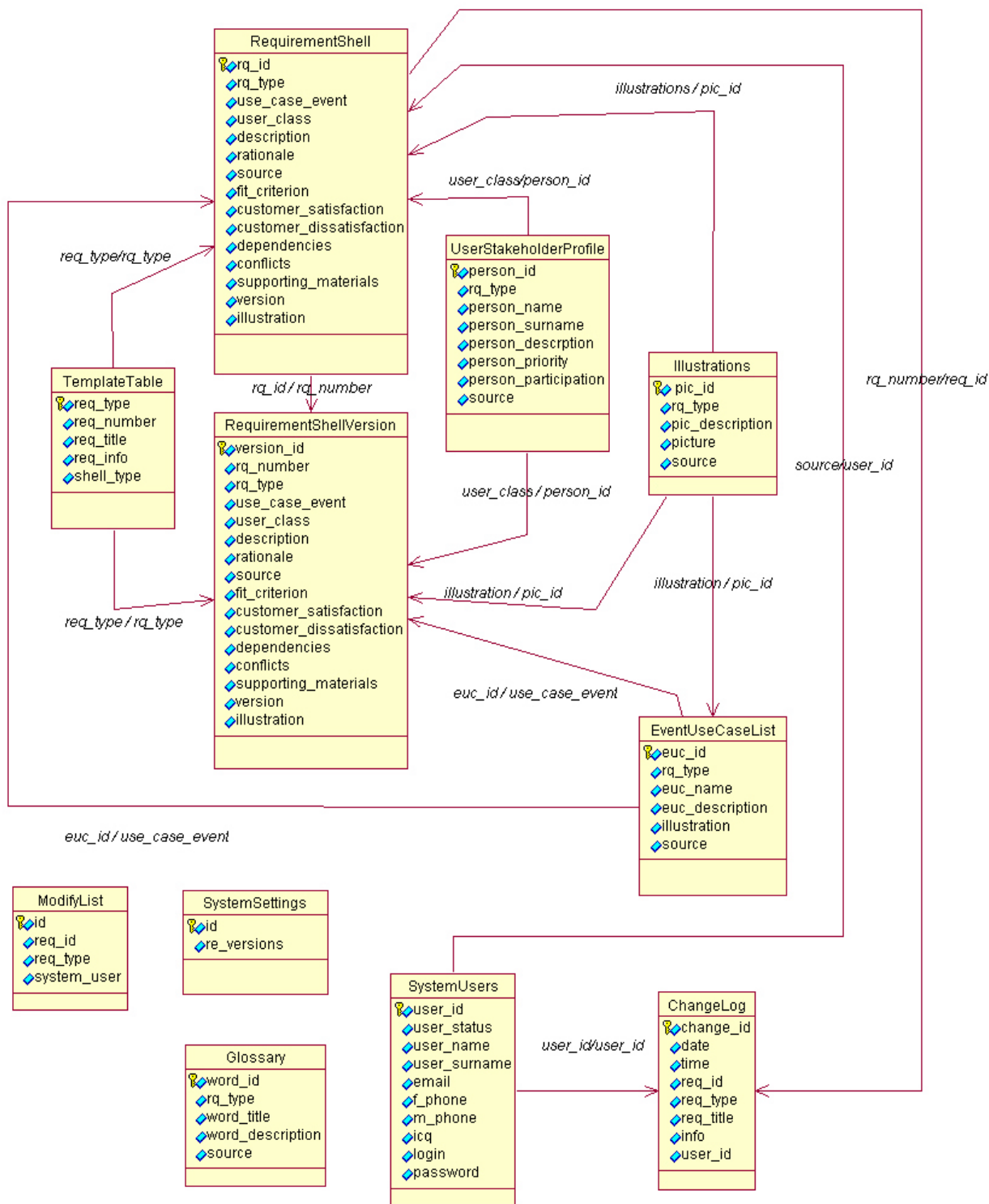
Pav. 29:
‘ReuirementShellVersion’
duomenų struktūra

Laukas	Reikšmė
version_id	3
rq_id	1
rq_type	21
use_case_event	1
user_class	1
description	Close the elevator firewall if fire in turret.
rationale	In case of heavy turret damage elevator doors must be quickly closed.
source	1
fit_criterion	5
customer_satisfaction	5
customer_dissatisfaction	5
dependencies	2, 3
conflicts	
supporting_materials	
version	3
illustration	1

Pav. 30: ‘RequirementShellVersion’ užpildymas

5.5.9. Bendra sistemos duomenų saugyklos loginė schema

Aukščiau aprašyta duomenų struktūra turi būti apjungta į MySQL duomenų bazę, kuri dirba vartotojo serveryje. Duomenų bazės pavadinimas sąlyginai yra ‘srs’. Duomenų bazės ‘srs’, kaip bendras sistemos duomenų struktūros vaizdas, yra parodytas pav. 31 sekančiame puslapyje.



Pav. 31: Bendras sistemos duomenų struktūros vaizdas

5.6. Duomenų apdorojimo procesai

5.6.1. Duomenų apdorojimo konceptas

Duomenų bazės 'srs' lentelių duomenys, kurie buvo aprašyti aukščiau, yra apdorjami ir pasiekiami MySQL užklausomis, formuojamomis sistemos (SRSTemplate) darbo metu. Žemiau yra parodytos pagrindinės sistemos užklausos, užtikrinančios svarbiausias sistemos funkcijas.

5.6.2. Sistemos parametrų nustatymas

Sistemos parametrai yra tik vienas lentelės 'SystemSettings' laukas 're_versions'. Pirmiausia reikia šį lauką atrinkti. Tokia užklausa atrodo taip:

```
"SELECT re_versions FROM SystemSettings WHERE id = '1';"
```

Norint pakeisti reikalavimų versijų skaičiaus gylį, reikia pateikti tokią užklausą:

```
"UPDATE SystemSettings SET re_versions = [nauja reikšmė] WHERE id = '1';"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 58).

5.6.3. Vartotojo sukūrimas / panaikinimas

Reikalavimų specifikavimo proceso automatizavimo sistemos administratorius gali sukūrti ar panaikinti atitinkamą vartotoją. Norint sukūrti naują vartotoją sudaroma tokia užklausa:

```
"INSERT INTO `systemusers` (`user_id` , " +  
    "`user_status` , `user_name` , `user_surname` , " +  
    "`email` , `f_phone` , `m_phone` , `icq` , " +  
    "`login` , `password` )" +  
    "VALUES (" , '0' , " , " , " , " , " , " , " +  
    [vartotojo prisijungimo vardas] +  
    " , " +  
    [vartotojo prisijungimo slaptažodis] +  
    " );"
```

Užklausos metu yra sukūriamas naujas vartotojas, kurio visi laukai yra tušti ir tik suvedamas pradinis vartotojo prisijungimo vardas ir vartotojo prisijungimo slaptažodis.

Siekiant panaikinti vartotoją padaroma tokia užklausa:

```
"DELETE FROM SystemUsers WHERE login = '[vartotojo prisijungimo vardas]';"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 58).

5.6.4. Vartotojo prisijungimas / atsijungimas

Vartotojui bandant prisijungti prie sistemos yra patikrinamas jo vartotojo prisijungimo vardas ir slaptažodis. Tai atrodo taip:

```
"SELECT user_id FROM SystemUsers WHERE login = " +  
    [vartotojo prisijungimo vardas] +  
    " AND password = " +  
    [vartotojo prisijungimo slaptažodis] +  
    " , ,  
    , ,
```

Vartotojas prisijungia prie sistemos ir jo 'status' lauko reikšmė pasikeičia į '1'. Užklausa atrodo taip:

```
"UPDATE SystemUsers SET user_status = 1 WHERE login = " +  
    [vartotojo prisijungimo vardas] +  
    " AND password = " +  
    [vartotojo prisijungimo slaptažodis] +  
    " , ,  
    , ,
```

Priešingai, vartotojui norint atsijungti nuo sistemos, jo 'status' reikšmė pasikeičia į '0'. Užklausa struktūra tokia pat, kaip ir vartotojui prisijungiant prie sistemos.

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 61).

5.6.5. Vartotojo duomenų redagavimas

Vartotojas gali pakeisti savo duomenys užduodamas tokią užklausą:

```
"UPDATE SystemUsers SET " +  
    " user_name = " +  
    [vartotojo vardas] +  
    " , user_surname = " +  
    [vartotojo pavardė] +  
    " , email = " +  
    [vartotojo elektroninio pašto adresas] +  
    " , f_phone = " +  
    [vartotojo telefono numeris] +  
    " , m_phone = " +  
    [vartotojo mobiliojo telefono numeris] +  
    " , icq = " +  
    [vartotojo icq numeris] +  
    " , login = " +  
    [vartotojo prisijungimo vardas] +  
    " , password = " +  
    [vartotojo prisijungimo slaptažodis] +  
    " WHERE login = " +  
    [vartotojo prisijungimo vardas] +  
    " , ,  
    , ,
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 60).

5.6.6. Sistemos vartotojų peržiūra

Informaciją apie sistemos vartotojus galima pasiekti tokia užklausa:

```
"SELECT user_id, user_name, user_surname, email, f_phone, m_phone, icq FROM SystemUsers;"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 60).

5.6.7. Informacija apie seansą

Informacija apie seansą susideda iš pačio vartotojo prisijungimo vardo ir sąrašo vartotojų, kurie šiuo metu irgi dirba prie sistemos. Toks sąrašas gaunamas taip:

```
"SELECT user_name FROM SystemUsers WHERE user_status = 1;"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 61).

5.6.8. Reikalavimo sukūrimas / redagavimas / panaikinimas

Kiekvienas vartotojas gali sukūrti naują reikalavimą. Reikalavimui sukūrti turi būti panaudota tokia užklausa:

```
"INSERT INTO RequirementShell " +  
    "(" `rq_id` , `rq_type` , `use_case_event` , " +  
    "`user_class` , `description` , `rationale` , " +  
    "`source` , `fit_criterion` , `customer_satisfaction` , " +  
    "`customer_dissatisfaction` , `dependencies` , " +  
    "`conflicts` , `supporting_materials` , `version` , " +  
    "`illustration` ) VALUES (" + " +  
    "" , "" + [reikalavimo tipas] +  
    "" , "" + [reikalavimo panaudojimo atvejis]+  
    "" , "" + [reikalavimo vartotojas/aktorius] +  
    "" , "" + [reikalavimo trumpas aprašymas] +  
    "" , "" + [reikalavimo aprašymas] +  
    "" , "" + [reikalavimo autorius] +  
    "" , "" + [tinkamumo kriterijus]+  
    "" , "" + [vartotojo pasitekinimas]+  
    "" , "" + [vartotojo nepasitenkinimas] +  
    "" , "" + [reikalavimo priklausomumai] +  
    "" , "" + [reikalavimo konfliktai] +  
    "" , "" + [reikalavimo šaltiniai] +  
    "" , "" + [reikalavimo versija] +  
    "" , "" + [reikalavimo paveikslai] + "" );";
```

Čia yra petikektas pavyzdys kaip reikalavimas įterpiamas į 'RequirementShell' lentelę. Ta pati strategija naudojama ir kitoms lentelėms, tačiau visų pirmą reikia atpažinti į kokią lentelę reikia įterpti reikalavimą.

Pirmiausia atrenkami visi reikalavimų tipai ir jų aprašymai :

```
"SELECT * FROM TemplateTable ;"
```

Toliau vartotojas pasirenka vieną jam tinkamą reikalavimą. Pagal reikalavimo numeri nustatomas jo formatas:

```
"SELECT shell_type FROM TemplateTable WHERE req_number =[parinkto reikalavimo numeris];"
```

Vėliausiai analogiškai yra parenkamas reikalavimo tipas:

```
"SELECT req_type FROM TemplateTable WHERE req_number =[parinkto reikalavimo numeris];"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 59).

Reikalavimo redagavimas yra sudėtingas procesas dėl versijų išsaugojimo (kontrolės) ir reikalavimo užrakinimo / atrakinimo. Išsaugant redaguotą reikalavimą, jo senoji versija yra išsaugojama 'RequirementShellVersions' lentelėje (šiuo atveju):

```
"INSERT INTO RequirementShellVersion " +  
    "(" `version_id` , `rq_id` , `rq_type` , `use_case_event` , " +  
    "`user_class` , `description` , `rationale` , " +  
    "`source` , `fit_criterion` , `customer_satisfaction` , " +  
    "`customer_dissatisfaction` , `dependencies` , " +  
    "`conflicts` , `supporting_materials` , `version` , " +  
    "`illustration` ) VALUES (" , " +  
    ....
```

Naujoji versija išsaugojama 'RequirementShell' lentelėje:

```
"UPDATE RequirementShell SET " +  
    " rq_type = " +  
    [reikalavimo tipas] +  
    " , use_case_event = " +  
    ....
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 62).

5.6.9. Reikalavimo paieška

Reikalavimo paieška tarp visų reikalavimų specifikavimo dokumento šablono reikalavimų, tačiau svarbiausias paieškos aspektas yra paieška tarp atominių reikalavimų. Taip yra dėl šių reikalavimų gausybės ir svarbumo vietos dokumente. Todėl nuspręsta kaip reikalavimų paieškos atskaitos tašką laikyti reikalavimų paiešką Requirements Shell formą atitinkančioje 'RequirementsShell' lentelėje.

'RequirementsShell' paieškos raktai turi užtikrinti greitą ir išsamią paiešką. Būtina užtikrinti paiešką įvairiais pjūviais.

Svarbiausi p̄jūviai yra :

- pagal reikalavimo tipą (*funkcinis, nefunkcinis, ...*)
- pagal panaudojimo atvejį (*greitai atskirti visus reikalavimus priklausančius konkrečiam atvejui*)
- pagal vartotoją / aktorį (*greitai atskirti visus reikalavimus priklausančius konkrečiam aktoriui*)
- pagal reikalavimo autorių
- pagal raktąžodį reikalavimo trumpajame aprašyme

Prie pajieškos raktų nuspręsta priskirti laukus:

- *rq_id* (reikalavimo ID numeris),
- *rq_type* (reikalavimo tipas),
- *description* (reikalavimo trumpas aprašymas),
- *source* (reikalavimo autorius),
- *use_case_event* (panaudojimo atvejis, kuriame šis reikalavimas panaudotas),
- *user_class* (vartotojas / aktorius, kuris naudos šį reikalavimą).

Gali būti užpildytas kaip keletas, vienas, nevienas ar visi pajieškos laukai. Paieška vykdoma tik pagal užpildytus laukus panaudojant loginį junginį ‘AND’. Jei laikai yra neužpildyti, parodomi visi reikalavimai.

Pajieškos užklausa ‘RequirementShell’ atrodo taip:

```
"SELECT * FROM RequirementShell [WHERE]
    [rq_id = reikalavimo_ID_numeris ]
    [AND]
    [rq_type = reikalavimo_tipas]
    [AND]
    [source = reikalavimo_autorius]
    [AND]
    [use_case_event = panaudojimo_atvejis]
    [AND]
    [user_class = vartotojas / aktorius]
    [AND]
    [description LIKE '%trumpo_aprašymo_raktąžodis%'];"
```

Tuo pačiu metu, panaudojus atitinkamus iš ‘RequirementShell’ įvestų pajieškos raktų, vykdoma paieška kituose reikalavimų specifikavimo dokumento formuose. Tai užtikrina atrinkti tuos reikalavimus, kurie yra susiję su jiejkomu ‘RequirementShell’ formato reikalavimu.

Panaudojimo atvejų paieška atrodo taip:

```
"SELECT * FROM EventUseCaseList [WHERE]
    [source = reikalavimo_autorius]
    [AND]
    [euc_id = panaudojimo_atvejis]
    [AND]
    [euc_description LIKE '%trumpo_aprašymo_raktažodis%'];"
```

Vartotojų / aktorių paieška atitinkamai atrodo kaip parodyta žemiau:

```
"SELECT * FROM UserStakeholderProfile [WHERE]
    [source = reikalavimo_autorius]
    [AND]
    [person_id = vartotojas / aktorius]
    [AND]
    [person_description LIKE '%trumpo_aprašymo_raktažodis%'];"
```

Tokia pat strategija naudojama ir paieškai tarp paveikslų sąrašo ('Illustrations') ir žodyno sąrašo ('Glossary'):

```
"SELECT * FROM Illustrations [WHERE]
    [source = reikalavimo_autorius]
    [AND]
    [pic_description LIKE '%trumpo_aprašymo_raktažodis%'];"
```

```
"SELECT * FROM Glossary [WHERE]
    [source = reikalavimo_autorius]
    [AND]
    [word_description LIKE '%trumpo_aprašymo_raktažodis%'];"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 58).

5.6.10. Reikalavimo užrakinimas / atrakinimas

Jeigu sistemos vartotojas pradeda reikalavimo redagavimą, sistema automatiškai uždeda "užraktą" ant to reikalavimo ir neleidžia kitiems vartotojams redaguoti šį reikalavimą, kol jis nebus pabaigtas. Užbaigus redagavimą, "užraktas" yra nuimamas. Reikalavimas yra užrakinamas panaudojus tokią užklausą:

```
"INSERT INTO ModifyList (mod_id, req_id, req_type, system_user)" +
" VALUES (" +
    [reikalavimo ID numeris] +
    ", " +
    [reikalavimo tipas] +
    ", " +
    [vartotojo prisijungimo vardas] +
    ");";
```

Reikalavimas yra atrakinamas panaikinus anksčiau įterptą įrašą iš 'ModifyList' lentelės. Yra naudojama tokia užklausa:

```
"DELETE FROM ModifyList WHERE req_id = '[reikalavimo ID numeris]' AND req_type = '[reikalavimo tipas]';"
```

Proceso funkcionalumas grafiškai pavaizduotas sekų diagrama (Priedas. Skyrius 10.2. psl. 62).

5.6.11. Pakeitimų žurnalo papildymas

Pakeitimų žurnalas parodo kokie pakeitimai buvo įvykdyti reikalavimų specifikavimo proceso metu. Pakeitimai yra automatiškai įvedami į lentelę 'ChangeLog', įvykus reikalavimo tvarkymui. Pakeitimų žurnale nurodomas pakeisto / tvarkyto reikalavimo unikalus numeris, tipas, pavadinimas, pakeitimo prigimtis (sukūrtas, koreguotas, panaikintas), vartotojo prisijungimo vardas bei pakeitimo data ir laikas. Naudojama tokia užklausa:

```
"INSERT INTO ChangeLog " + "(change_id, date, time, req_id, req_type, req_title, info, user_login)
VALUES (" + , .....
```

5.7. Apibendrintas sistemos klasių vaizdas

Remiantis panaudojimo atvejų diagrama (5.2.3.) ir panaudojimo atvejų aprašymų (5.2.4.) bei sekų diagramomis (Priedas. Skyrius 11.2.), galima sukūrti projektuojamos sistemos klasių diagramą. SRSTemplate architektūra yra griežtai objektiškai orientuota ir sudaryta remiantis JAVA 'Applet' struktūra. Reikia pabrėžti, kad SRSTemplate naudoja standartinę prisijungimo prie MySQL duomenų bazės paketą 'mysql-connector-java-3.0.11-stable', kurio klasės yra sistemos bibliotekoje 'com'. Taip pat naudojamos standartinės SUN klasės JDBCAdapter.java, TableMap.java, TableSorter.java. Šios klasės yra saugojamos sistemos bibliotekoje 'database'. SRSTemplate apibendrinta klasių diagrama parodyta (Priedas. Skyrius 11.1.).

6. SRSTemplate eksperimentinis tyrimas

Sistemos prototipo SRSTemplate eksperimentinis įvertinimas yra pdalintas į dvi dalis. Pirmoji yra sistemos panaudojamumo įvertinimas, kur yra testojamas sistemos panaudojamumas, funkcionalumas ir vartotojo sąsaja. Tyrimas vykdomas panaudojant “constructive interaction” metodą ir specializuotą “Morae” testavimo įrangą. Antroji testavimo dalis yra sistemos palyginimas su analogiška egzistuojančia sistema – AnalystPRO ir RequisitePRO.

6.1. Sistemos panaudojamumo įvertinimas

6.1.1. “Constructive Interaction” metodas

Sistemos panaudojamumui įvertinti buvo panaudotas “constructive interaction” metodas. Šis metodas yra plačiai paplitęs testuojant įvairių programinę įrangą. Jakob Nielsen [10] aprašo šį metodą kaip kito populiaraus “Thinking aloud” metodo išplėtimą. “Thinking aloud” metodo esmė yra paprasta. Testuotojas (vartotojas) bando dirbti su testuojama programine įranga, vykdydamas pateiktą užduotį. Testavimo metu testuotojas turi garsiai išreikšti savo mintis, ketinimus ir darbo su programa eigą. Apžvalgininkas turi konspektuoti visus testuotojo komentarus ir esminius vieksmus. Šis metodas yra labai populiarus, nes nereikalauja daug papildomų išlaidų sistemos testavimui. Dažnai pakanka ne vieno testuotojo, o testavimo rezultatų efekyvumas yra aukštas. Tačiau egzistuoja viena problema. Žmogus nėra pripratęs garsiai išstarti savo mintys, todėl vartotojai dažnai pradeda tylėti eksperimento metu. Tokiam trūkumui eliminuoti buvo išrastas “Constructive Interaction” metodas. Jis skyriasi nuo “Thinking Aloud” metodo tik tuo, kad du testuotojai (vartotojai) bando bendromis pastangomis išspręsti duotą užduotį. Testavimo metu vyksta darbinis dialogas tarp sistemos testutoju ir “užtylėjimo” galimybė labai maža.

Labai svarbu, kad eksperimento metu apžvalgininko įtaka būtų minimali. Geriausia yra visiškai izaliuoti apžvalgininką ir testuotojus. Tačiau apžvalgininkas tuo pačiu metu turi turėti galimybę matyti visus testuotojų jūdesius, komantarus ir konspektuoti kiekvieną jų žingsnį. Kad užtikrinti tokias testavimo sąlygas, buvo panaudotas specializuotas testavimo įrankis – MORAE testavimo paketas.

6.1.2. MORAE testavimo paketas

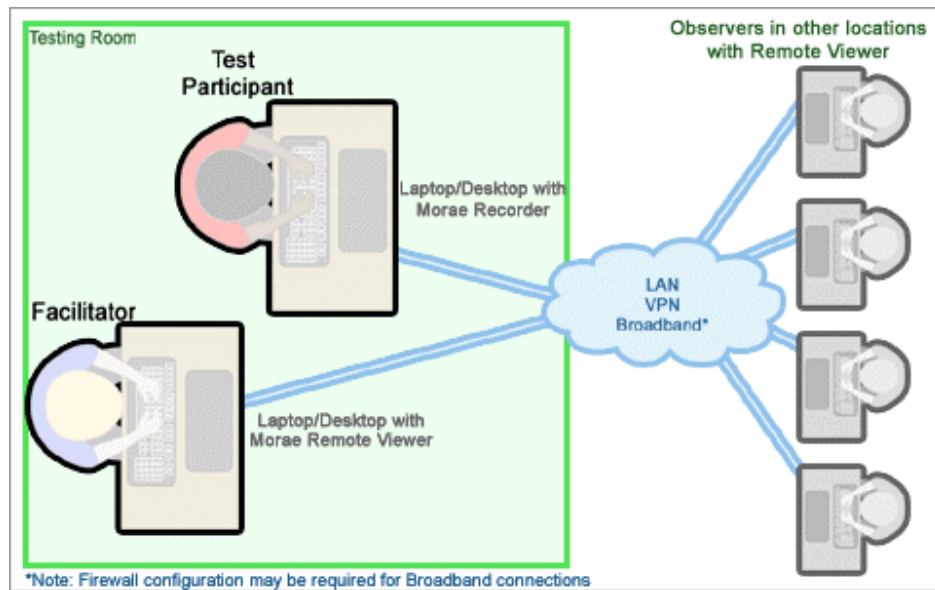
MORAE testavimo paketas yra profesionalus įrankis programinės įrangos panaudojamumui testuoti. Šį įrankį siūlo kompanija 'TechSmith' (www.techsmith.com). *“Morae yra pirmasis vieningas, visiškai automatizuotas produktas, įvairiai programiniai įrangai testuoti. Tai gali būti programos, programų sistemos, interneto sveainės ir portalai ir t.t. Morae Viewer, Recorder ir Manager kartu siūlo lengvą vartojimui ir upratimui testavimo įrašymo sistemą, pagrįstą TechSmiths Rich Recording Technology™. Ši technologija leidžia žvelgti gyčiau į panaudojamumo procesą naudojant testavimo vaizdo įrašymą ir sinchronizaciją su testavimo proceso eiga ir komentarais apie testuotojo patirtį naudojant testuojamą produktą”*[11].

Morae paketas susideda iš 3 komponentų: Morae Recorder, Morae Viewer, Morae Manager. Morae Recorder naudoja Web Kamera ir išrašo netik testavimo darbo aplinką, bet ir testuotojų veido išraiškas, komentarus ir diskusijas bei judesius testavimo metu. Testavimo metu, apžvalginikas stebi testuotojus naudojant programą Morae Viewer, kuri prisijungia prie programos Morae Recorder. Morae Viewer deka apžvalginikas mato testavimo darbo aplinką, testuotojus, girdi jų diskusijas ir gali įterpti pastabas, nurodydamas ypatingas ir svarbias testavimo eigos vietas. Šios pastabos yra siunčiamos atgal į Morae Recorder aplinką ir yra išsaugomos kartu su baziniu testavimo failu. Morae Manager yra svarbiausias Morae paketo komponentas. Naudojant Morae Manager, bazinio testavimo failo pagrindu yra sukūriamas testavimo projektinis failas. Testavimo projektas leidžia visapusiškai ir bet kuriuo metu peržiūrėti visą testavimo procesą, padarytus komentarus, laisvai pasiekti pažymėtas testavimo vietas ir atlikti jų analizę.

Darbo autorius susipažino su MORAE produktu Bremeno Universitete darbo programinės įrangos panaudojamumo testavimo laboratorijoje metu. Ten pat ir buvo atliktas SRSTemplate prototipo testavimas. Testavimo procesas aprašytas skenačiame skyriuje.

6.1.3. Testavimo procesas

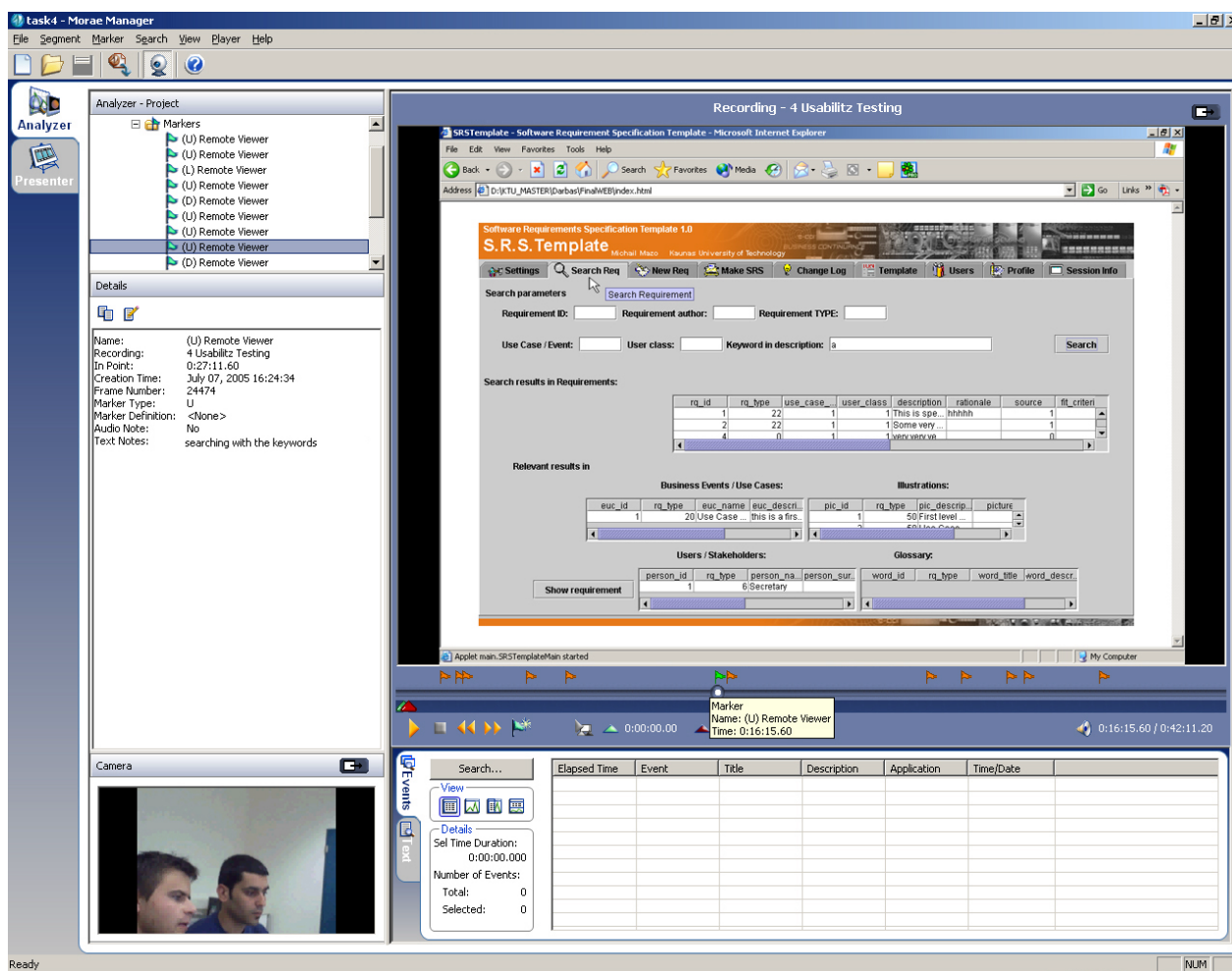
SRSTemplate testavimo aplinka buvo Bremeno Universiteto programinės įrangos panaudojamumo testavimo laboratorijos aplinka. Morae siūlomas testavimo planas parodytas pav. 34. Laboratorija turi tik testavimo kambarį, todėl kitų apžvalginikų testavimo metu nebuvo. SRSTemplate buvo lokaliai įdiegta testuotojų kompiuteryje. Sistemos testuotojais buvo darbo autorius ir vienas iš Bremeno universiteto magistrantų. Kitas Bremeno universiteto magistrantas atliko apžvalginiko (stebėtojo) vaidmenį.



Pav. 34: Morae testavimo aplinka

Testavimo metu buvo pateikta užduotis:

- Sukūrti naują vartotoją (darbo autorius atliko sistemos administratoriaus vaidmenį).
- Atsijungti nuo sistemos ir prisijungti kaip naujas vartotojas.
- Sutvarkyti naujo vartotojo asmeninius duomenis.
- Sukūrti naują reikalavimą.
- Surasti reikalavimą pagal aprašymo raktažodį.
- Redaguoti surastą reikalavimą.
- Išsaugoti naują reikalavimo versiją.
- Peržiūrėti informaciją apie vartotojus.
- Atsijungti nuo sistemos.
- Prisijungti prie sistemos kaip sistemos administratorius ir panaikinti sukurtą vartotoją.



Pav. 35: Morae Manager darbo aplinka

Testavimo metu buvo sudarytas naujas Morae testavimo projektas. Morae Manager pagalba testavimo procesas buvo analizuojamas kelis kartus (pav. 35). Analizės rezultatai pateikti skyriuje 6.1.4.

6.1.4. Testavimo rezultatai pagal ISO 9241

ISO 9241 yra plačiai paplitęs tarptautiniės standartizavimo organizacijos standartas sistemos panaudojamumo trūkumams tirti. [12] Šis aprašymo standartas susideda iš 17 dalių. ISO 9241 dokumentas suteikia standartus visų programinės įrangos panaudojamumo parametrų tvistavimui, nuo spalvų gamos parinkimo iki integralumo į darbo aplinką.

Interaktyvių sistemų panaudojamumo ir ergonomikos testavimui svarbiausios aprašymo dalys yra 10 ir 11. 10 dalis (Dialogue Principles), aprašo dialogo tarp žmogaus ir kompiuterio principą ir ergonomiką. 11 dalis (Guidance on Usability) fokusuojasi į sistemos panaudojamumo

kriterijus. Šie skyriai ir jų poskyrių vertimai (vertimai yra netiesioginiai, bet atspindintys paskyrio turinį) pateikti lentelėje pav. 36.

Dialogue Principles (part 10)	Dialogo principai (10 dalis)
Task appropriateness	Užduoties atitinkamumas
Self describing capability	Sistemos išvaizdumas
Controllability	Sistemos valdymas
Agreeing on expectation	Navigacijos išdėstymas
Error tolerance	Klaidų toleravimas
Customisable	Adaptyvumas
Subserves learning	Išmokstamumas
Requirements to the Usability (part 11)	Panaudojamumo reikalavimai (11 dalis)
Effectivity to solve the task	Užduoties sprendimo efektyvumas
Efficiency in using the system	Sistemos panaudojimo produktyvumas
User satisfaction	Vartotojo pasitenkinimas

Pav. 36: ISO 9241 standarto 10 ir 11 skyriai

Žemiau pateikti testavimo rezultatai ir jų suvestinė lentelė (pav. 37):

Dialogo principai

Užduoties atitinkamumas – pagrindinė SRSTemplate užduotis yra užtikrinti reikalavimų specifikavimo dokumento kooperatyvų kurimą keliems sistemos analitikams / projektuotojams. Sistema pilnai atitinka savo projektinę paskirtį.

Sistemos išvaizdumas – sistema turi labai paprastą navigaciją, tačiau yra keletas trūkumų. Svarbiasias, kad informacija apie darbo sesiją kartu su atsijungimo mygtuku yra atskiroje sistemos kortelėje. Vartotojas dažnai tiesiog pamiršta atsijungti.

Sistemos valdymas – sistemos valdymas yra nesudėtingas, tačiau, kaip buvo minėta aukščiau, informacija apie darbo sesiją yra paslėpta.

Navigacijos išdėstymas - nesudėtinga ir ergonomiška navigacija nevargina vartotojo ir yra optimali ir aiški.

Klaidų toleravimas – sistemos atsparumas klaidoms yra aukštas. Praktiškai visos sistemos klaidingos situacijos yra dubliuotos klaidų pranešimų formomis.

Adaptyvumas – SRSTemplate adaptyvumas pagal vartotojo pageidavimus yra pakankamas. Applet technologija keičia vartotojo sąsają priklausomai nuo operacinės sistemos aplinkos. Sistema atskyria sistemos administratoriu ir sistemos vartotojus, užtikrindama minimalią kontesktinę navigaciją.

Išmokstamumas – sistema nereikalauja papildomų žinių ir yra lengvai išmokstama tik pradėjus darbą su ja.

Panaudojamumo reikalavimai

Užduoties sprendimo efektyvumas – naudojant SRSTemplate, galima iškart pereiti prie reikalavimų specifikavimo dokumento rašymo. Sistema palaiko visas pagrindines reikiamas kooperatyvaus naudojimosi funkcijas: sistemos įvykių vizualizacija, versijos kontrolė. Sprendimo efektyvumas yra aukštas.

Sistemos panaudojamumo produktyvumas – kaip ir sprendimo efektyvumas, sistemos panaudojamumo efektyvumas yra aukštas.

Vartotojo pasitenkinimas – sistemos testuotojas buvo patenkintas sistemos darbu, tačiau kylo klausimų dėl vartotojo sąsajos neužbaigtumo. Tačiau, atsižvelgiant į tai, kad SRSTemplate yra tik busimos sistemos prototipas ir dar neturi visų reikiamų sistemos versijų, pasitenkinimo įvertinimas yra aukštas.

Įvertinimas Kriterijus	I. aukštas	aukštas	vid.	žemas	I. žemas
Užduoties atitinkamumas	X				
Sistemos išvaizdumas		X			
Sistemos valdymas			X		
Navigacijos išdėstymas		X			
Klaidų toleravimas		X			
Adaptyvumas				X	
Išmokstamumas	X				
Užduoties sprendimo efektyvumas		X			
Sistemos panaudojimo produktyvumas		X			
Vartotojo pasitenkinimas		X			

Pav. 37: SRSTemplate prototipo testavimo rezultatų suvestinė lentelė

6.2. Sistemos palyginimas su AnalystPRO ir RequisitePRO

Artimiausi ir kone vieninteli analogai SRSTemplate yra Goda Software siūlomas programinis paketas AnalystPRO [8] ir IBM siūlomas RequisitePRO [13]. AnalystPRO bei RequisitePRO, kaip ir SRSTemplate, tikslas yra palengvinti ir maksimaliai automatizuoti reikalavimų specifikavimo procesą. AnalystPRO ir RequisitePRO yra profesionali programinė įranga, jau kelis metus rinkoje. SRSTemplate prototipas yra tik idėjos realizavimo primasis žingsnis, todėl kai kurie bruožai pateikti palyginimo lentelėje yra teoriniai ir dar nerealizuoti (kaip pvz. Reikalavimų specifikavimo dokumento generavimas). Pagrindinių sistemų parametrų palyginimas yra pateiktas lentelėje pav. 38.

Sistemos bruožas	SRSTemplate	AnalystPRO	RequisitePRO
Nepriklausomumas nuo operacinės sistemos	+	-	+
Reikalavimų specifikavimo šablonas	+	-	-
Reikalavimų versijų kontrolė	+	+	+
Konkurencijos kontrolė	+	+	+
Reikalavimų specifikavimo dokumento generavimas	+	+	+

Nepriklausomumas nuo darbo vietos	+	-	+
Integruotas diagramų kūrimo įrankis	-	+	-
Pakeitimų fiksavimas	+	+	+
Reikalavimų ir pakeitimų paieška	+	+	+
Informacija apie sistemos vartotojus, ir jų darbą	+	-	-

Pav. 38: SRSTemplate ir AnalystPRO pagrindinių bruožų palyginimas

SRSTemplate svarbiausi skyriamieji bruožai yra sistemos įdiegimo lengvumas ir nereiklumas. SRSTemplate kliento dalis yra tik JAVA Applet, kuris yra laikinai išsaugotas vartotojo kompiuteryje. Jis nereikalauja papildomo įdiegimo ir yra visiškai nepriklausomas nuo operacinės sistemos. Taip pat ir tarnybinės stoties (serverio) dalis nereikalauja papildomų išlaidų ir reikalavimų. SRSTemplate naudoja Apache ir MySQL serverius ir bet kokią tarnybinės stoties operacinę sistemą. Šie bruožai teigiamai skiria SRSTemplate nuo AnalystPro [8] ir RequisitePro [13] paketų (pav. 39).

	SRSTemplate	AnalystPRO	RequisitePRO
Kliento pusė	Bet kokia o.s.	Windows NT, 2000, XP	Windows NT, 2000, XP
Tarnybinės stoties pusė	Bet kokia o.s. + Apache + MySQL	Win NT Server, Win 2000 Server, Win 2003 Server	Win NT Server, Win 2000 Server, Win 2003 Server
Papildoma Programinė įranga	nėra	MS Word (Office)	MS Word 2000/02/03, MS Access 2000/02/03, IBM UDB DB2, MS SQL Server, Oracle, Rational paketas

Pav. 39: SRSTemplate ir AnalystPRO esminių skirtumų palyginimas

7. Išvados

Remiantis į atliktą darbą, galima padaryti šias išvadas:

1. Atliktas pagrindinių reikalavimų specifikavimo šablonų VOLERE ir IEEE STD 830-1998 tyrimas. Analizuotos ir palygintos pagrindinės šių šablonų savybės ir ypatumai. Remiantis šiu populiariausių šablonų palyginimu, buvo nuspręsta, kuri šabloną automatizuoti ir realizuoti kūriamame reikalavimų specifikavimo automatizavimo sistemos prototipe SRSTemplate.
2. Ištirti reikalavimų specifikavimo dokumento kūrimo procesai ir reikalavimų specifikavimo ciklas. Atlikta reikalavimų specifikavimo dokumento savybių ir esminių bruožų analizė. Ši analizė padėjo suprasti ir iskirti specifines reikalavimų specifikavimo dokumento puses bei suprasti kūrimos sistemos vietą reikalavimų specifikavimo procese. Tai buvo ypač svarbu sudarant sistemos veiklos konteksto diagramą ir panaudojimo atvejų diagramas.
3. Atlikta jau egzistuojančių panašių reikalavimų specifikavimo sistemų analizė. Ištirtos sistemos AnalystPRO ir RequisitePRO. Analizuojant šias sistemas buvo išskirtos jų pagrindinės teigiamos savybės ir trūkumai. Tokia analizė tapo kūrimos sistemos prototipo SRSTemplate funkcinių ir nefunkcinių reikalavimų pagrindu.
4. Specifikuoti ir užfiksuoti funkciniai ir nefunkciniai reikalavimai kūrimai reikalavimų specifikavimo automaizavimo sistemai SRSTemplate. Remiantis šiais reikalavimais buvo specifikuotas sistemos architektūros modelis, duomenų struktūros ir duomenų apdorojimo procesai, sistemos klasių struktūra.
5. Realiztuos SRSTemplate reikalavimų specifikavimo automatizavimo sistemos prototipas. SRSTemplate siūloma architektūra užtikrina visišką nepriklausomumą nuo operacinės sistemos, reikalauja minimalių pastangų įdiegiant šį programinį produktą. Išvaizdi ir paprasta vartotojo sąsajos struktūra ir sistemos navigacija nereikalauja papildomo laiko darbo su sistema apmokymui. SRSTemplate

užtikrina galimybę keliems programinės įrangos analitikams vienu metu dirbti prie sistemos. Naudojamas VOLERE reikalavimų specifikavimo šablonas. SRSTemplate pasižymi visomis reikalavimų specifikavimo įrankio savybėmis: reikalavimų tvarkymas, reikalavimų paieška, reikalavimų specifikavimo dokumento generavimas. SRSTemplate yra interaktyvus, daugelio vartotojų darbą palaikantis, tinklapio pagrindo įrankis. SRSTemplate taip pat pasižymi esminėmis grupinio darbo įrankio savybėmis: versijų kontrole, konkurencijos kontrole, pakeitimu atsekamumu, informacija apie sistemos vartotojus, ir jų darbą.

6. Atliktas SRSTemplate prototipo panaudojamumo įvertinimas. Įvertinimo kokybei užtikrinti buvo naudojamas “constructive interaction” metodas ir specializuota panaudojamumo įvertinimo programinė įranga “Morae”. Panaudojamumo įvertinimo rezultatai užfiskuoti panaudojamumo įvertinimo protokole pagal ISO 6241 10 ir 11 skyrių. Toks įvertinimas padėjo maksimaliai įvertinti sistemos panaudojamumą ir atskleisti sistemos teigiamas savybes ir trūkumus. Remiantis šiuo įvertinimu galima specifiuoti papildomus reikalavimus sistemos modernizavimui.
7. Galutinis darbo įvertinimas – palyginimas su jo analogais. SRSTemplate prototipo pagrindiniai bruožai ir savybės buvo palygintos su AnalystPRO ir RequisitePRO. Reikia pabrėžti, kad nors SRSTemplate ir tik sistemos prototipas, jo savybės teigiamai pasižymėjo palyginimo metu. Pagrindinis SRSTemplate laimėjimas – sistemos visiškai nepriklausomumas nuo vartotojo operacinės sistemos ir darbo aplinkos, bei paprastumas lygiagrečiai su visomis pagrindinėmis kolektyvinio reikalavimų specifikavimo dokumento kūrimo proceso palaikymo funkcijomis. Remiantis šiuo darbu ir realizotu SRSTemplate prototipu galima sukūrti profesionalų reikalavimų specifikavimo automatizavimo įrankį, unikalų pagal savo technines charakteristikas.

Michail Mazo, 2005 m. gruodis 18 d.

8. Literatūra

- [1] “Mastering the requirements process” kurso apžvalga.
Url: <http://www.systemsguild.com/GuildSite/Robb/MRP.html> [2005.02.25]
- [2] **Japenga, Robert**: How to write a software requirements specification
Url: <http://www.microtoolsinc.com/Howsrs.php> [2005.02.25]
- [3] IEEE puslapis
Url: <http://www.ieee.org/portal/site> [2005.02.25]
- [4] IEEE STD 830-1998 dokumentacija
Url: <http://www.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf> [2005.02.25]
- [5] Volere puslapis ir šablonas
Url: <http://www.volere.co.uk/> [2005.02.25]
- [6] **Robertson, Suzanne. Robertson, James**: Mastering the Requirements process
Addison Wesley, London, 2005; ISBN 0-201-36046-2
- [7] AnalystPRO produkto puslapis
Url: <http://www.analysttool.com> [2005.02.25]
- [8] AnalystPRO savybių aprašymas
Url: http://www.analysttool.com/AnalystPro_Features.pdf [2005.02.25]
- [9] **Koch, Michael**: Telecooperation Systems (Computer-Supported Cooperative Work)
Universität Bremen, Digital Media master program, 6 ECTS credits. Paskaitų konspektas
- [10] **Jakob Nielsen**: Usability Engineering. Boston u.a.: AP Professional, 1993.
- [11] Morae's Getting Started Guide v.1.2
Url: http://download.techsmith.com/morae/docs/gettingstarted/Morae_getting_started.pdf
[2005.07.09]
- [12] Ergoweb – Ergonomics Standard and Guidelines – ISO 9241
Url: <http://www.ergoweb.com/resources/reference/guidelines/iso9241.cfm> [2005.07.14]
- [13] Success starts with requirements management: IBM Rational RequisitePRO
Url: <ftp://ftp.software.ibm.com/software/rational/web/datasheets/version6/reqpro.pdf> [2005.12.25]
- [14] **Butleris, Rimantas**: Reikalavimų specifikavimas naudojant šablonus. Volere šablonas
reikalavimams specifikuoti. KTU, Paskaitų medžiaga.

9. Terminų ir sutrumpinimų žodynas

AnalystPRO – specializuotas reikalavimų specifikavimo dokumento rašymo ir sistemos projektavimo produktas. Siūlomas Goda Software.

RequisitePRO – specializuotas reikalavimų specifikavimo dokumento rašymo ir sistemos projektavimo produktas. Siūlomas IBM.

Constructive Interaction – programinės įrangos panaudojamumo tyrimo metodas. Metodo esmė : du testuotojai bando bendromis pastangomis išspręsti užduotą užduotį, panaudojant testuojamą programinę įrangą. Testuotojai garsiai išreiškia savo mintis. Apžvalgininkas fiksuoja testuotojų darbą ir aprašo testavimo eigą.

IEEE STD 830-1998 – IEEE siūlomas reikalavimų programinei įrangai specifikavimo dokumento šablonas.

ISO 9241 – ISO siūlomas programinės įrangos panaudojamumo testavimo dokumento standartas.

MORAE – specializuotas programinės įrangos paketas, programinės įrangos panaudojamumo testavimui. Siūlomas “TechSmith”.

RS – (Requirements Specification) reikalavimų specifikacija.

SRS – (Software Requirements Specification) reikalavimų programinei įrangai specifikacija.

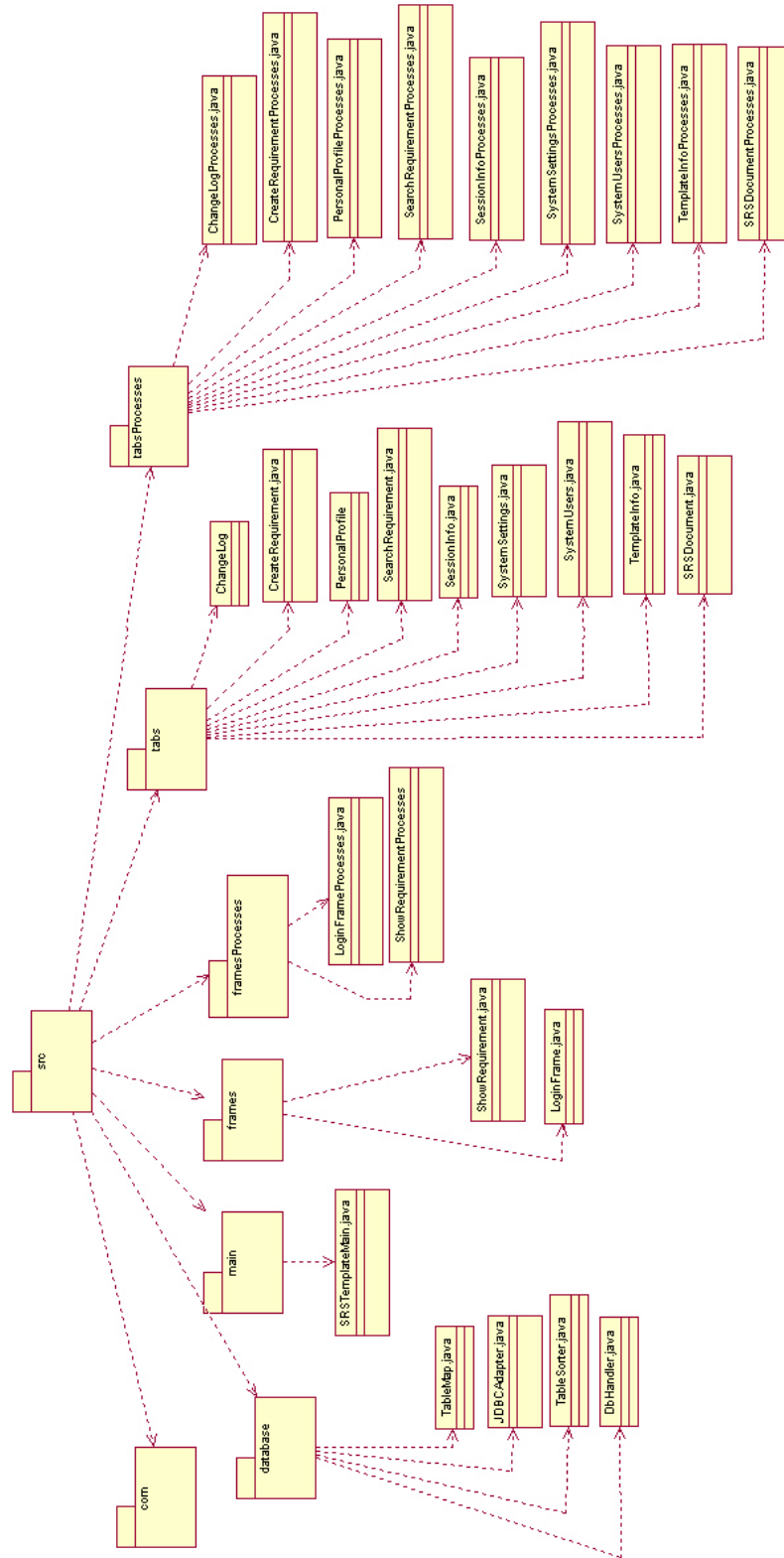
SRSTemplate – reikalavimų specifikavimo dokumento rašymo automatizavimo programinės įrangos prototipas.

Užrakto metodas – vienas iš konkurencijos kontrolės metodų. Metodo esmė – reikalavimas yra “užrakinamas” kai vienas iš vartotojų pradeda jo tvarkymą. Kiti vartotojai gali tvarkyti reikalavimą tik kai reikalavimo tvarkymas buvo užbaigtas ir reikalavimas buvo “atrankintas”.

VOLERE – Plačiai paplitęs reikalavimų programinei įrangai specifikavimo dokumento šablonas.

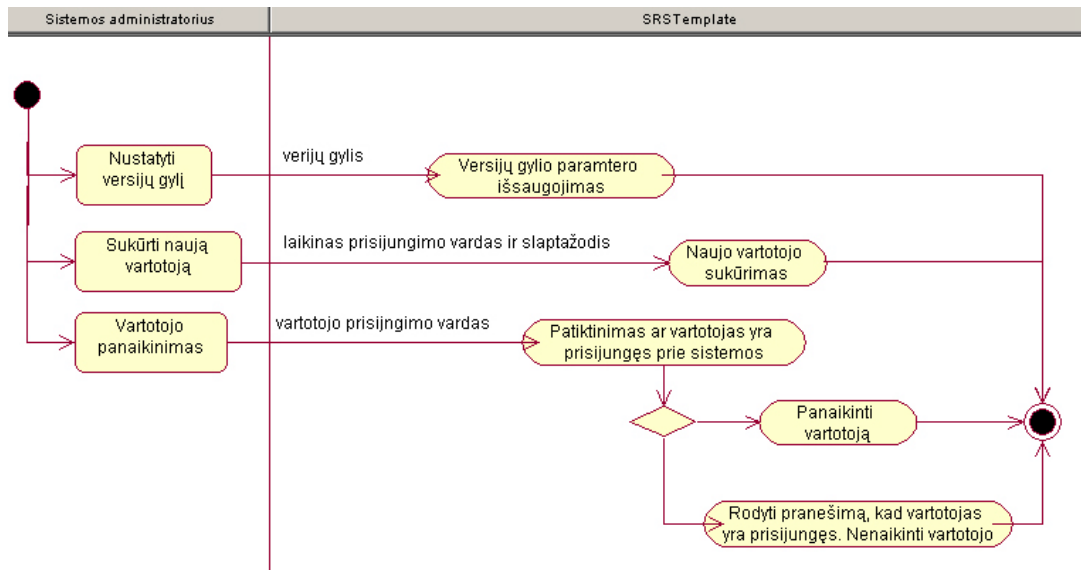
10. Priedai

10.1. Apibendrinta klasių diagrama

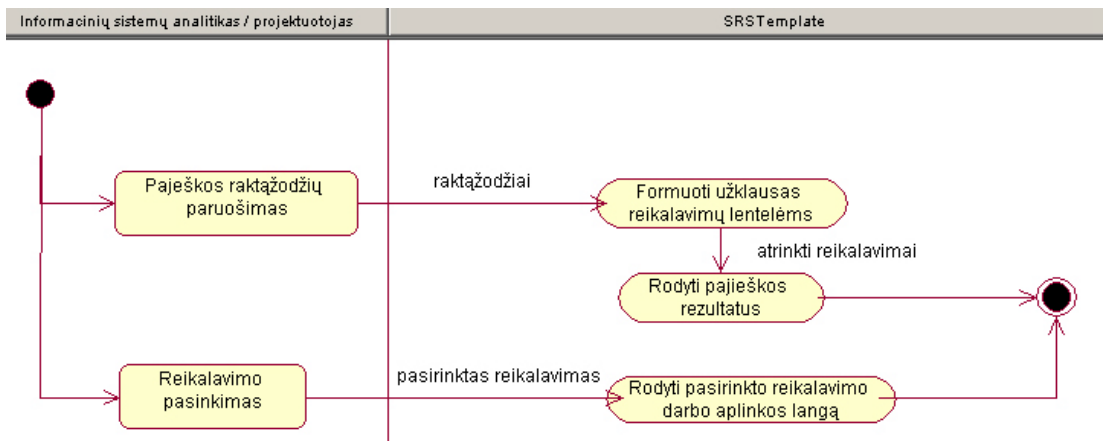


10.2. Panaudojimo atvejų sekų diagramos

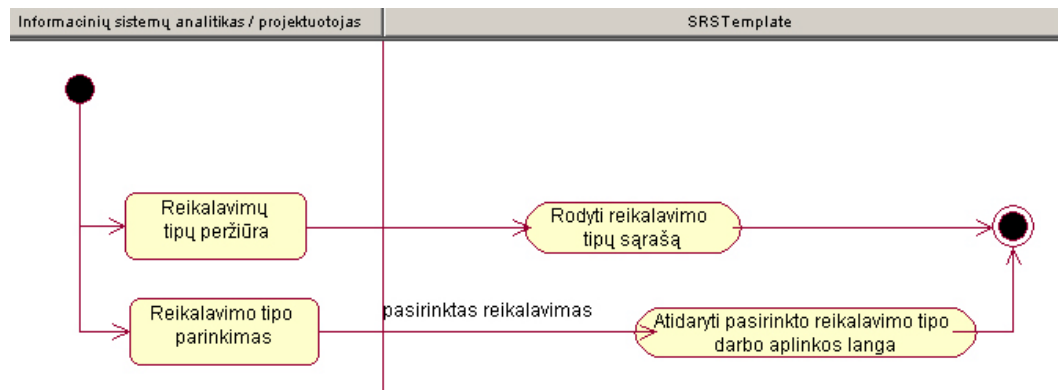
1. Panaudojimo atvejis: Sistemos nustatymai



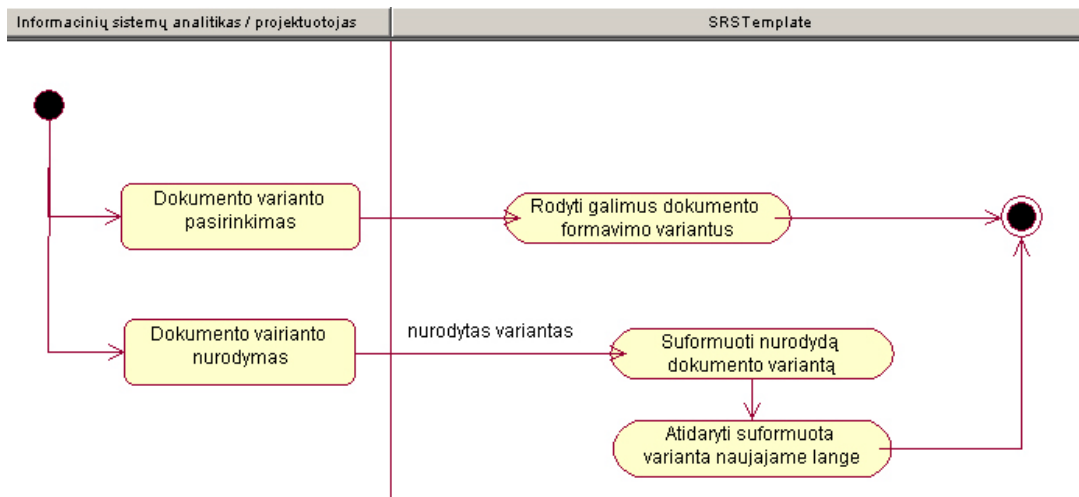
2. Panaudojimo atvejis: Reikalavimo paieška



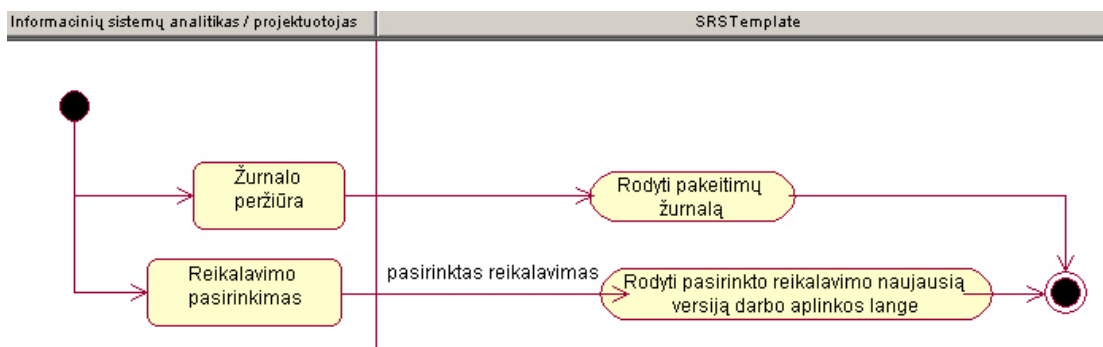
3. Panaudojimo atvejis: Reikalavimo sukūrimas



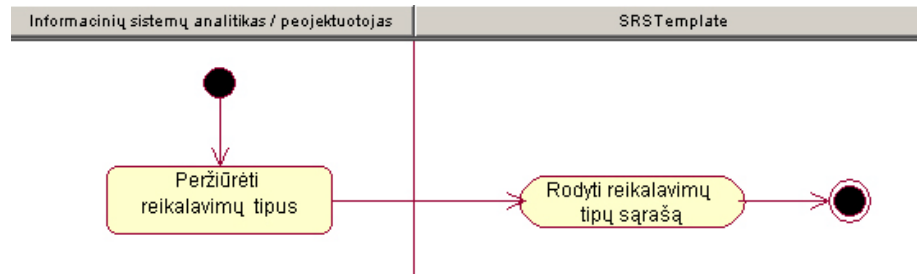
4. Panaudojimo atvejis: Reikalavimų dokumento formavimas



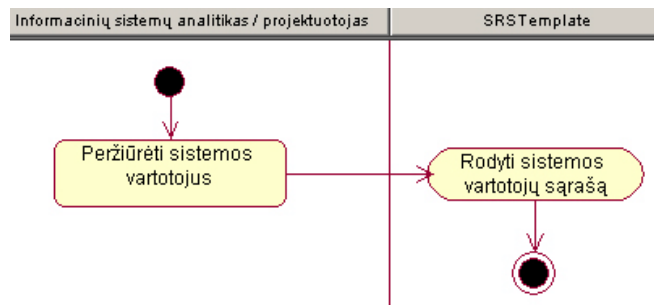
5. Panaudojimo atvejis: Pakeitimų žurnalas



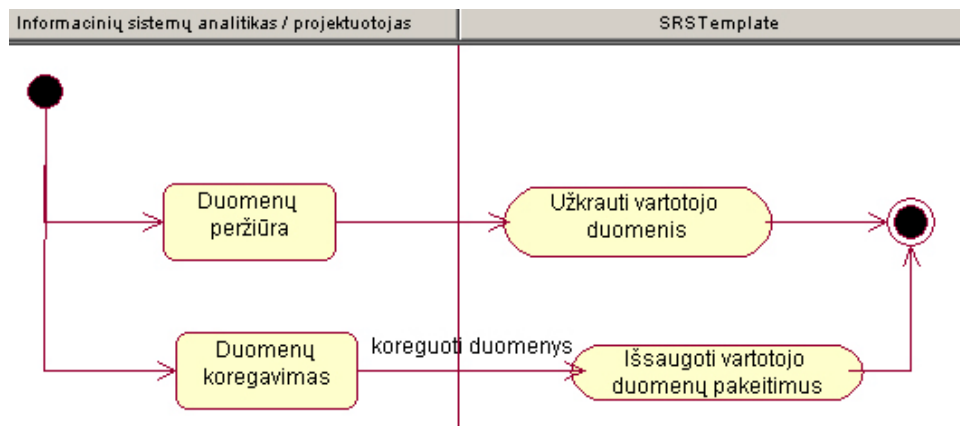
6. Panaudojimo atvejis: Informacija apie reikalavimus



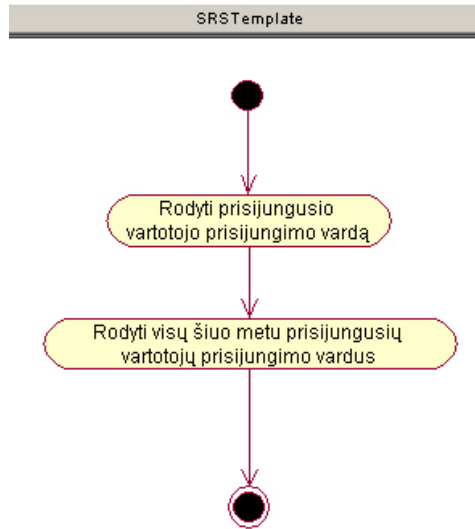
7. Panaudojimo atvejis: Sistemos vartotojai



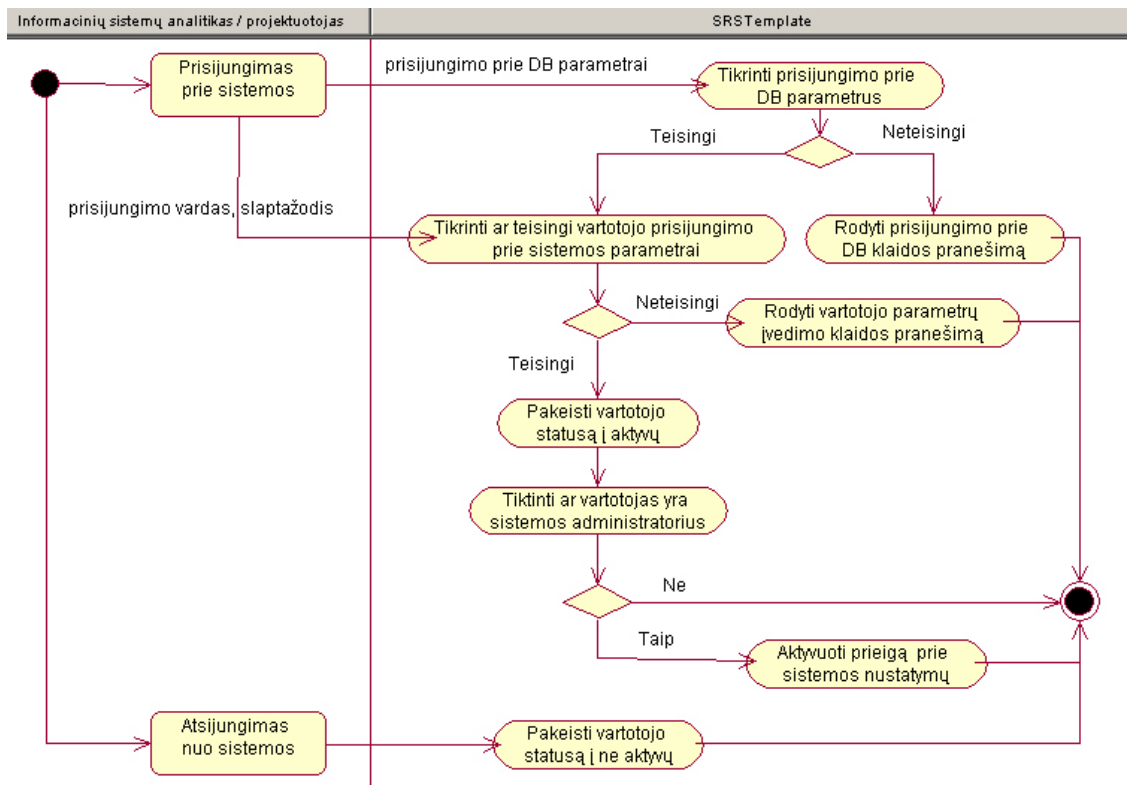
8. Panaudojimo atvejis: Asmeniniai duomenys



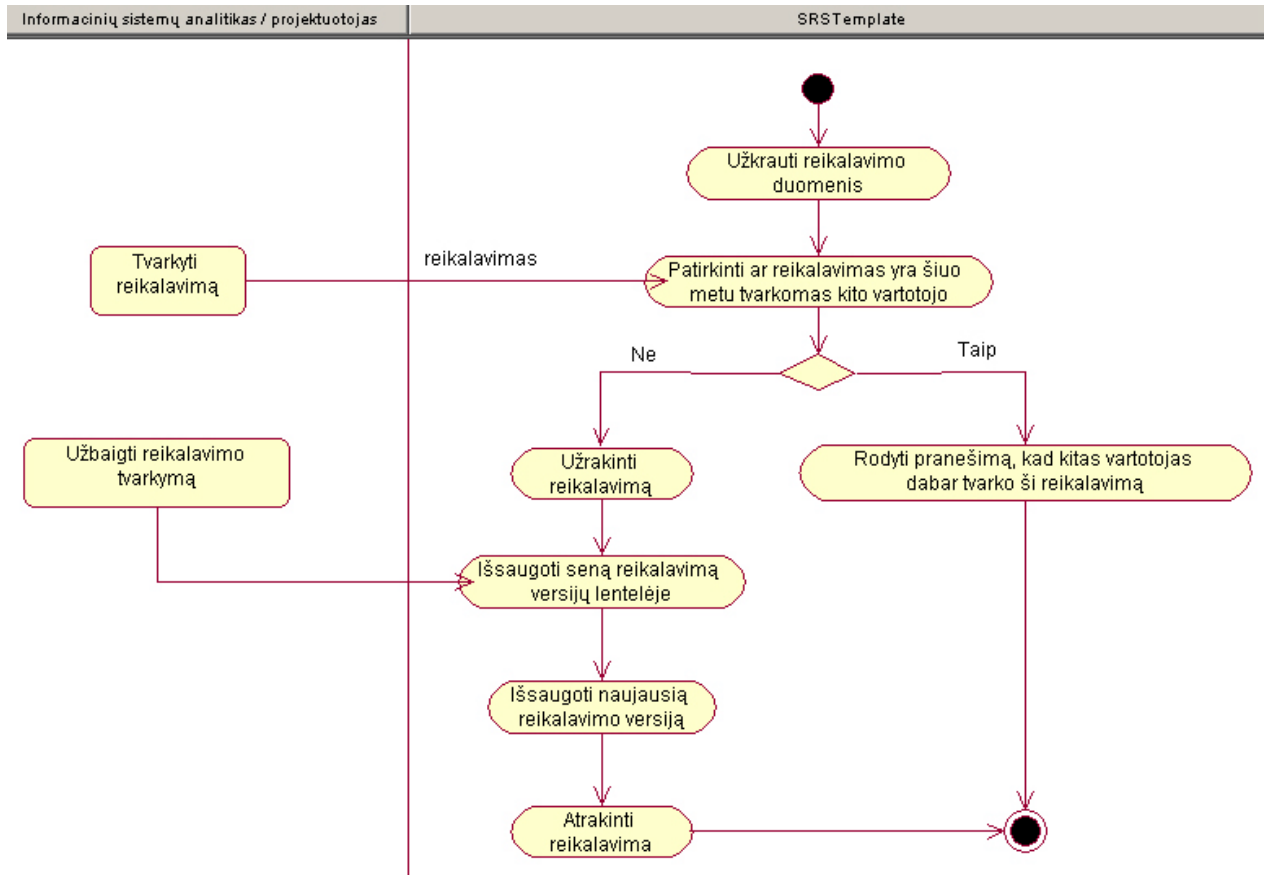
9. Panaudojimo atvejis: Informacija apie seansą



10. Panaudojimo atvejis: Prisijungimas / atsijungimas

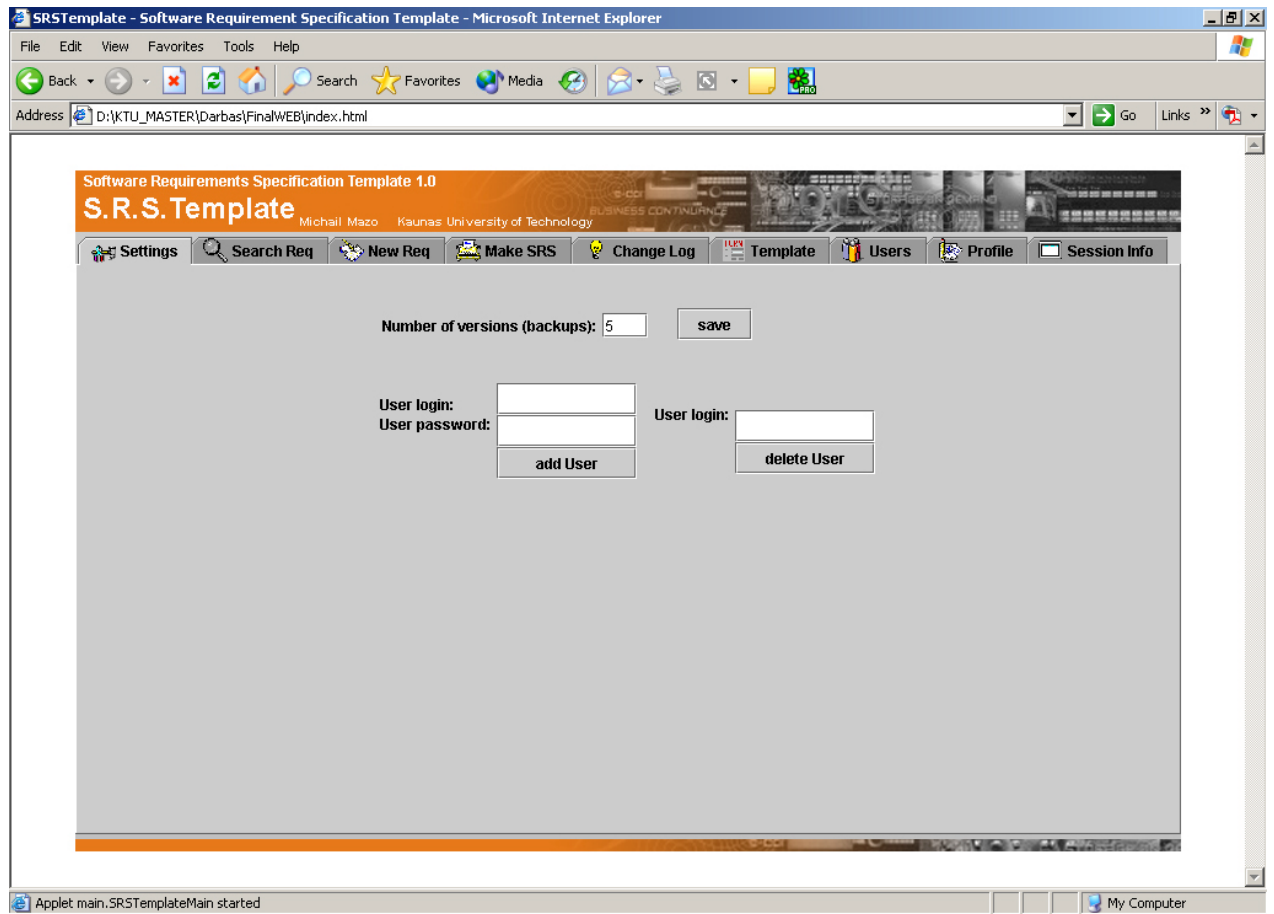


11. Panaudojimo atvejis: Reikalavimo darbo aplinka



10.3. Sistemos vaizdai

Sistemos nustatymai



Reikalavimo paieška

SRSTemplate - Software Requirement Specification Template - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail

Address D:\KTU_MASTER\Darbas\FinalWEB\index.html Go Links

Software Requirements Specification Template 1.0
S.R.S.Template
 Michail Mazo Kaunas University of Technology

Settings Search Req New Req Make SRS Change Log Template Users Profile Session Info

Search parameters

Requirement ID: Requirement author: Requirement TYPE:

Use Case / Event: User class: Keyword in description:

Search results in Requirements:

rq_id	rq_type	use_case...	user_class	description	rationale	source	fit_criteri
1	22	1	1	This is spe...	hhhhh	1	
2	22	1	1	Some very ...		1	
4	0	1	1	very very ve		0	

Relevant results in

Business Events / Use Cases:

euc_id	rq_type	euc_name	euc_descri...
1	20	Use Case ...	this is a firs..

Illustrations:

pic_id	rq_type	pic_descrip...	picture
1	50	First level ...	
2	50	Use Case ...	

Users / Stakeholders:

person_id	rq_type	person_na...	person_sur..
1	6	Secretary	

Glossary:

word_id	rq_type	word_title	word_descr..

Applet main.SRSTemplateMain started My Computer

Pakeitimų žurnalas

The screenshot shows a web browser window titled "SRSTemplate - Software Requirement Specification Template - Microsoft Internet Explorer". The address bar shows the URL "D:\KTU_MASTER\Darbas\FinalWEB\index.html". The page content includes a navigation menu with items like "Settings", "Search Req", "New Req", "Make SRS", "Change Log", "Template", "Users", "Profile", and "Session Info". The "Change Log" section displays a table with the following data:

date	time	req_id	req_type	req_title	info	user_login
2005.10.30	14:27	1	1	This is spe...	requireme...	serioga
2005.10.30	16:44	1	1	This is spe...	requireme...	mozas
2005.11.2	0:20	1	1	This is spe...	requireme...	mozas
2005.11.2	0:21	1	1	This is spe...	requireme...	mozas
2005.11.2	0:25	1	1	This is spe...	requireme...	mozas
2005.11.2	0:43	1	1	This is spe...	requireme...	tadas
2005.11.2	0:44	1	1	This is spe...	requireme...	mozas
2005.11.2	0:59	1	1	This is spe...	requireme...	tadas
2005.11.2	1:1	1	1	This is spe...	requireme...	mozas
2005.11.2	1:2	1	1	This is spe...	requireme...	tadas
2005.11.2	13:20	1	1	This is spe...	requireme...	mozas

Below the table is a button labeled "Show Selected Requirement".

Sistemos vartotojai

user_id	user_name	user_surname	email	f_phone	m_phone	icq
1	Michailas	Mazo	mazo@one.lt	555 777	2323 34 56	333-567-890
3	Sperauskas	Tadas	tadas@sperauskas.lt	444444444	777777777	989898989
4	Mihail	Dimitrov	mihail@dimitrov.bg	343434343	343434434	343434333
6						

Applet main.SRSTemplateMain started

My Computer

Asmeniniai duomenys

SRSTemplate - Software Requirement Specification Template - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Refresh Print Mail Stop

Address D:\KTU_MASTER\Darbas\FinalWEB\index.html Go Links

Software Requirements Specification Template 1.0
S.R.S.Template
Michail Mazo Kaunas University of Technology

Settings Search Req New Req Make SRS Change Log Template Users Profile Session Info

User name:	<input type="text" value="Michailas"/>
User surname:	<input type="text" value="Mazo"/>
User email:	<input type="text" value="mazo@one.lt"/>
User fixed phone:	<input type="text" value="555 777"/>
User mobile phone:	<input type="text" value="2323 34 56"/>
User ICQ/Skype/MSN, etc.:	<input type="text" value="333-567-890"/>
User login:	<input type="text" value="mozas"/>
User password:	<input type="password" value="****"/>

Save

Applet main.SRSTemplateMain started My Computer

Informacija apie seansą

Software Requirements Specification Template 1.0
S.R.S.Template
Michail Mazo Kaunas University of Technology

Settings Search Req New Req Make SRS Change Log Template Users Profile Session Info

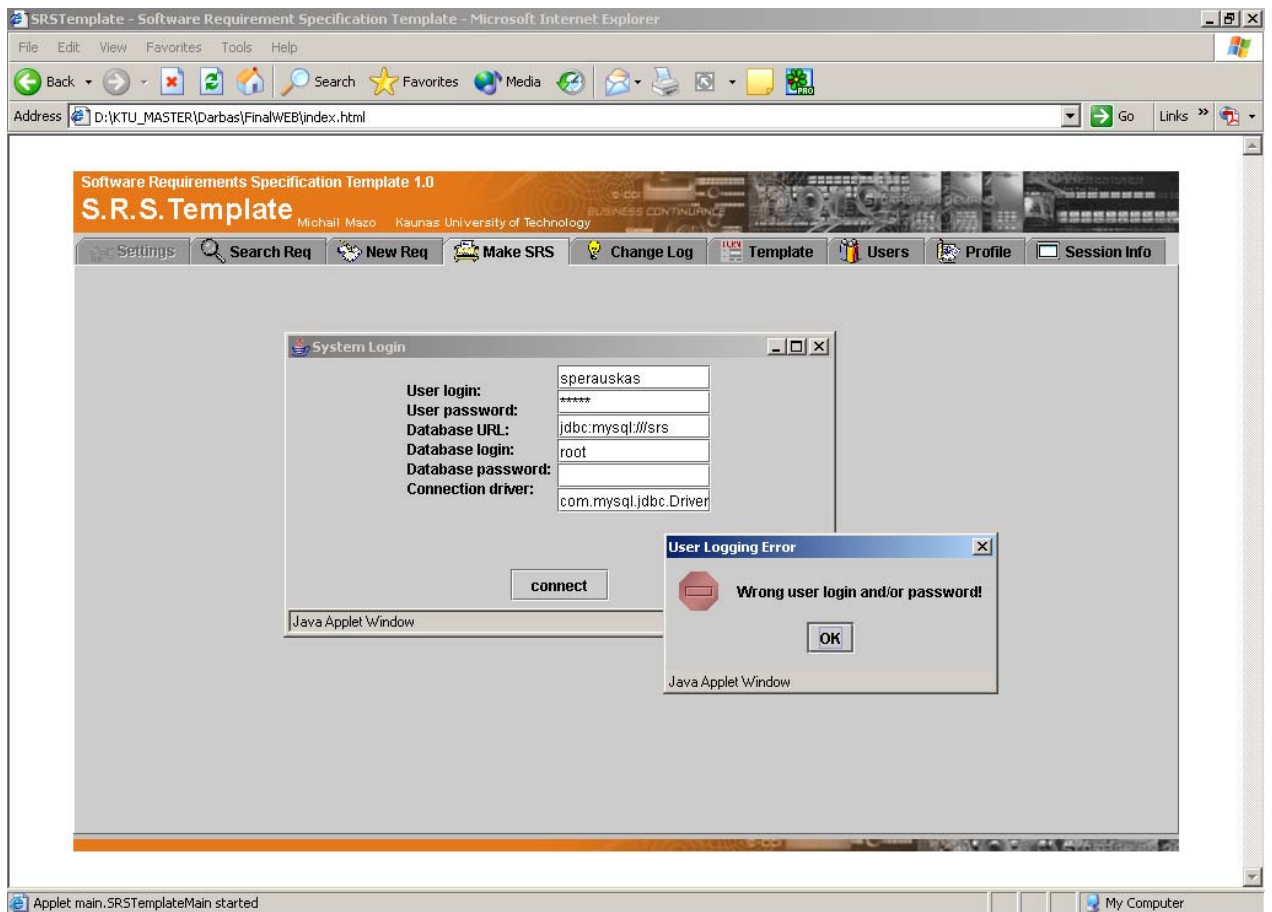
User: tadas

user_name
Michallas
Sperauskas

Logout

Applet main.SRSTemplateMain started My Computer

Prisijungimas / atsijungimas



Reikalavimo darbo aplinka

SRSTemplate - Software Requirement Specification Template - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail

Address D:\KTU_MASTER\Darbas\FinalWEB\index.html Go Links

Software Requirements Specification Template 1.0
S.R.S.Template
 Michail Mazo Kaunas University of Technology

Requirement frame

Requirement ID: 1 Author: 1
 Requirement type: 22 Version: 11
 Use Case / Event ID: 1
 Use Case / Event ID: 1

This is special data requirement for data input. MAMA

Description:

hhhhh

Rationale:

Customer satisfaction: 3 Customer dissatisfaction: 3
 Dependencies: Conflicts:
 Support materials: Fit criterion: 0

Illustrations

pic_id	rq_type	pic_descip...	picture	s
1	50	First level ...		
2	50	Use Case		

Show All Stick illustration Unstick illustration Show Illustration

version_id	rq_id	rq_type	use_case...	user_class	description	rationale
19	1	22	1	1	This is spe...	
20	1	22	1	1	This is spe...	

Ok Save Delete Print Show version

Java Applet Window

Applet main.SRSTemplateMain started My Computer

Pranešimas apie užrakintą reikalavimą

SRSTemplate - Software Requirement Specification Template - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail

Address D:\KTU_MASTER\Darbas\FinalWEB\index.html Go Links

Software Requirements Specification Template 1.0
S.R.S.Template
 Michail Mazo Kaunas University of Technology

Settings Search Req New Req Make SRS Change Log Template Users Profile Session Info

Search parameters


Requirement ID: Requirement author: Requirement TYPE:

Use Case / Event: User class: Keyword in description:

Search results in Requirements:

rq_id	rq_type	use_ca
1	22	

Requirement is Locked

 **User mozas is currently working on this requirement!**

Java Applet Window

Relevant results in

Business Events / Use Cases:

euc_id	rq_type	euc_name	euc_descri...
1	20	Use Case ...	this is a firs...

Illustrations:

pic_id	rq_type	pic_descrip...	picture
1	50	First level ...	
2	50	Use Case ...	

Users / Stakeholders:

person_id	rq_type	person_na...	person_sur..
1		6	Secretary

Glossary:

word_id	rq_type	word_title	word_descr..

Applet main.SRSTemplateMain started

My Computer

10.4. Darbo CD

10.5. Recenzija

10.6. Mokslinis straipsnis

Prie šio darbo yra pridedamas autoriaus straipsnis, paskelbamas X tarpuniversitetinėje magistrantų ir doktorantų konferencijoje INFORMACINĖS TECHNOLOGIJOS ir atspausdintas šios konferencijos pranešimų medžiagoje (leidinio ISBN 9955-09-852-X). Straipsnio kopija pridedama sekančiuose puslapiuose.