

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

EDITA ŠARAPNICKYTĖ

SAITYNO INFORMACINIŲ SISTEMŲ VARTOTOJO
SĄSAJOS TESTAVIMO METODIKA

Magistro darbas

Darbo vadovas
doc. dr. K. Kapočius

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

EDITA ŠARAPNICKYTĖ

SAITYNO INFORMACINIŲ SISTEMŲ VARTOTOJO
SĄSAJOS TESTAVIMO METODIKA

Magistro darbas

Darbo vadovas:
doc. dr. K. Kapočius
2013-05-

Recenzentas:
dr. A. Armonas
2013-05-

Atliko:
IFM-1/4 gr. studentas
Edita Šarapnickytė
2013-05-

KAUNAS, 2013

AUTORIŲ GARANTINIS RAŠTAS
DĖL PATEIKIAMO KŪRINIO

20.. - - d.
Kaunas

Autoriai, _____
(vardas, pavardė)

_____ ,
patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas
(toliau vadinama – Kūrinys) _____
(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

_____	_____	_____
(vardas, pavardė)	(parašas)	
_____	_____	_____
(vardas, pavardė)	(parašas)	
_____	_____	_____
(vardas, pavardė)	(parašas)	
_____	_____	_____
(vardas, pavardė)	(parašas)	

SANTRAUKA

Svarbus veiksnys, lemiantis vartotojų požiūrį į sistemą ir jos efektyvumą – tai aukštas sistemos panaudojamumas, kurį ženkliai veikia vartotojo sąsajos charakteristikos. Dėl milžiniškos konkurencijos pranašumas panaudojamumo srityje ypač aktualus saityne, todėl akivaizdu, jog saityno informacinių sistemų sąsają tikslinga nuodugniai testuoti. Vis dėlto atlikus žinomų testavimo būdų analizę nustatyta, jog jų pagalba saityno sistemų sąsajai testuoti keliami reikalavimai gali būti tenkinami tik iš dalies. Sprendžiant šią problemą buvo sukurtas specializuotas, būtent saityno IS vartotojo sąsajai testuoti pritaikytas sprendimas, apjungiantis ir pagal poreikį praplečiantis egzistuojančius metodus. Atlikta preliminari sukurtosios metodikos taikymo galimybių analizė leidžia daryti prielaidą, jog metodika turėtų leisti ne tik aptikti daugiau ir esminių vartotojo sąsajos klaidų, bet ir operatyviai priimti sprendimus dėl aptiktų defektų šalinimo.

SUMMARY

One of the key factors influencing the users' perception of a system is high usability, which, in turn, often comes down to the quality of the user interface of the system. Because of enormous competition, having an edge in usability is especially important on the Web. No wonder that Web information systems (WIS) user interface testing is of extreme importance. However, analysis of known testing methods has shown that none of them meets all requirements set for the WIS user interface testing process. To address this problem, a specialized WIS user interface testing methodology that combines and extends features of some known methods has been developed. Preliminary analysis of the methodology makes it possible to presume that it should allow for detection of a bigger number and of more significant defects, at the same time helping developers come up with appropriate corrections.

TURINYS

PAVEIKSLŲ SĄRAŠAS	8
LENTELIŲ SĄRAŠAS	9
ĮVADAS	10
1 ŽINOMŲ VARTOTOJO SĄSAJOS TESTAVIMO SPRENDIMŲ ĮVERTINIMAS	13
1.1. ANALIZĖS TIKSLAS IR METODAI.....	13
1.2. TYRIMO SRITIS IR OBJEKTAS	14
1.3. SAITYNO SAVITARNOS SVETAINĖ	14
1.4. SISTEMOS PANAUDOJAMUMAS IR JO RYŠYS SU VARTOTOJO SĄSAJA	15
1.5. PROGRAMINĖS ĮRANGOS TESTAVIMAS	16
1.5.1. <i>Testavimo procese dalyvaujantys veikėjai ir ryšiai tarp jų</i>	19
1.6. DAŽNIAUSIAI PASITAIKANČIOS VS IR PANAUDOJAMUMO KLAIDOS	20
1.7. VARTOTOJŲ AIBĖ, TIPAI IR SAVYBĖS.....	21
1.7.1. <i>Vartotojų tikslai ir problemos</i>	22
1.8. EGZISTUOJANČIOS PRIEMONĖS VS TESTUOTI	23
1.9. EGZISTUOJANTYS VS TESTAVIMO BŪDAI IR METODAI	24
1.9.1. <i>Rankinis VS testavimo būdas</i>	24
1.9.2. <i>Tradicinių PĮ funkcionalumo testavimo metodų taikymas VS testuoti</i>	25
1.9.3. <i>Atsitiktinės įvesties VS testavimo metodas</i>	25
1.9.4. <i>„Įrašymo atkūrimo“ VS testavimo metodas</i>	25
1.9.5. <i>Išbaigtų sąveikų sekų VS testavimo metodas</i>	26
1.9.6. <i>Pažintinės peržvalgos VS vertinimo metodas</i>	26
1.9.7. <i>Atitikties rekomendacijoms ir standartinio vertinimo metodas</i>	27
1.9.8. <i>Euristinis vertinimo metodas</i>	27
1.10. INFORMACINIŲ SISTEMŲ TESTAVIMO BŪDŲ PALYGINIMAS	28
1.11. VARTOTOJO SĄSAJAI TESTUOTI IR VERTINTI TAIKOMŲ METODŲ PALYGINIMAS	30
1.12. INFORMACINIŲ SISTEMŲ TESTAVIME NAUDOJAMŲ ĮRANKIŲ PALYGINIMAS	31
1.13. SIEKIAMAS SPRENDIMAS	33
1.14. ANALIZĖS IŠVADOS	34
2 SIŪLOMOS SAITYNO IS VS TESTAVIMO METODIKOS SUDARYMAS	35
2.1 METODIKOS KONCEPTAI.....	35
2.2 DALYKINĖS SRITIES KLASIŲ MODELIS.....	36
2.3 METODIKOS TAIKymo PANAUDOJIMO ATVEJŲ MODELIS.....	37
2.4 VS TESTAVIMO METODIKOS ETAPAI	38
2.4.1 <i>Testavimo planavimas</i>	41

2.4.2	<i>Vartotojo sąsajos specifikavimo etapas</i>	43
2.4.3	<i>Testo kūrimo etapas</i>	44
2.4.4	<i>Sprendimų pateikimo etapas</i>	46
2.4.5	<i>Validavimo ir verifikavimo etapas</i>	46
2.4.6	<i>Sprendimų paieškos etapas</i>	48
2.4.7	<i>Defektų verifikavimo etapas</i>	49
2.5	SIŪLOMO SPRENDIMO IŠVADOS	50
3	VARTOTOJO SĄSAJOS TESTAVIMO METODIKOS REALIZACIJA	51
3.1	VS KOMPONENTŲ AIBĖ	51
3.2	VS TESTAVIMO ATVEJŲ RINKINYS	52
3.3	DUOMENŲ BAZĖS MODELIS.....	54
3.4	REALIZACIJOS IŠVADOS	55
4	VS TESTAVIMO METODIKOS EKSPERIMENTINIS TYRIMAS	56
4.1	EKSPERIMENTO PLANAS	56
4.2	EKSPERIMENTO EIGA IR REZULTATAI.....	57
4.3	EKSPERIMENTO REZULTATŲ APIBENDRINIMAS	68
5	REZULTATŲ APIBENDRINIMAS IR IŠVADOS	70
6	LITERATŪRA	71
7	PRIEDAI	74
7.1	PRIEDAS. STRAIPSNIS	74
7.2	PRIEDAS. SUKURTO SPRENDIMO PROTOTIPO VARTOTOJO VADOVAS.....	80

PAVEIKSLŲ SĄRAŠAS

1 pav. Testavimo proceso pagrindas	17
2 pav. „V modelis“. Testavimo objekto kitimas testuojant sistemą.....	18
3 pav. Testavimo procese dalyvaujantys veikėjai	19
4 pav. Rankinio testavimo etapai	24
5 pav. Konceptualus testavimo metodikos modelis	35
6 pav. Dalykinės srities klasių modelis	36
7 pav. VS testavimo metodikos taikymo veiklos	37
8 pav. Metodikos taikymas testavimo lygiuose	38
9 pav. Vartotojo sąsajos testavimo metodikos etapai.....	39
10 pav. Siūlomos metodikos ir euristinio vertinimo metodo procesų sulyginimas	40
11 pav. Bendras VS testavimo metodikos procesas	41
12 pav. VS komponentų hierarchija	43
13 pav. VS komponento būsenos kitimas	44
14 pav. VS testo sudėtis	45
15 pav. Testo būsenos kitimas testavimo proceso metu.....	45
16 pav. Sprendimo pateikimo TA veiklos procesas	46
17 pav. Validavimo ir verifikavimo etapo procesas.....	47
18 pav. Testavimo atvejo būsenos kitimas.....	47
19 pav. Sprendimų paieškos etapo veiklos diagrama.....	48
20 pav. Defekto būsenų kitimas defektų verifikavimo etapo metu.....	49
21 pav. Defekto validavimo procesas	50
22 pav. Testuojamų saityno IS VS komponentų aibė	51
23 pav. VS testavimo metodikos prototipo duomenų bazės schema	55
24 pav. VS specifikavimo metu išskirtų komponentų kiekio palyginimas	59
25 pav. Universalių ir specifinių VS komponentų aibių sankirta	60
26 pav. Naujai priimtų ir pakartotinai panaudotų sprendimų kiekio kitimas defektų atžvilgiu.....	68

LENTELIŲ SĄRAŠAS

Lentelė 1. VS testavimo etapai.....	18
Lentelė 2. Testavimo veikėjų dalyvavimas užduočių vykdyme	20
Lentelė 3. Problemos su kuriomis susiduria vartotojai testuojant VS.....	22
Lentelė 4. Informacinių sistemų testavimo metodų palyginimas.....	29
Lentelė 5. Testavimo metodų ypatumai	30
Lentelė 6. Testavimo procese naudojamų testavimo įrankių palyginimas.....	32
Lentelė 7. Esybių ir jose saugomos informacijos sąrašas	37
Lentelė 8. VS testavimo atvejai/euristikos.....	52
Lentelė 9. Duomenų bazės lentelės	54
Lentelė 10. Tyrimo metu testuotos saityno IS.....	57
Lentelė 11. Testuotojų išskirti VS komponentai.....	57
Lentelė 12. Testuotojų sukurtos euristikos VS komponentams	61
Lentelė 13. Euristikoms priskirti sprendimo būdai	62
Lentelė 14. Defektai ir jų šalinimo sprendimai	63
Lentelė 15. VS komponentų, TA, defektų ir sprendimo būdų kiekiai	67
Lentelė 16. Žinomų testavimo metodų, būdų ir įrankių palyginimas su siūloma metodika	68

ĮVADAS

Šis magistro baigiamasis darbas priklauso informacinių sistemų inžinerijos studijų programai.

Temos aktualumas. Vis didesnę įtaką organizacijų veiklos efektyvumui ir populiarumui darančių saityno informacinių sistemų (IS) efektyvumas ir kokybė tampa vis aktualesni ir vertinami ne tik informacinių sistemų savininkų, jų kūrėjų, bet ir eilinių vartotojų. Nesunku nuspėti, jog tai lemia milžiniška saityne egzistuojanti konkurencija ir galimybė vienu pelės paspaudimu iš nepatikusios svetainės persikelti į patrauklesnę konkurentų sistemą. Ypač svarbus veiksnys, darantis įtaką vartotojų požiūriui į sistemą ir jos efektyvumą – tai didelis sistemos panaudojamumas, kurį labai veikia (grafinės) vartotojo sąsajos (VS) charakteristikos. Mažas panaudojamumas gali lemti tai, jog sistema bus tiesiog ignoruojama vartotojų, todėl efektyvus saityno informacinių sistemų vartotojo sąsajos testavimas yra ne mažiau svarbus nei funkcinis šių sistemų testavimas.

Tiriama problema. Testuojant tokio tipo informacines sistemas, taikyti tik funkcinio testavimo metodikas negana, tačiau žinomi tradicinės programinės įrangos vartotojo sąsajos testavimo būdai nėra visiškai tinkami testuoti IS sąsajai [10]. Saityno informacinių sistemų testavimas apskritai traktuotinas kaip besiformuojanti, neišplėtota informatikos sritis. Nors saityno IS vartotojo sąsajai testuoti galima pasirinkti šiuo metu žinomus vartotojo sąsajos testavimo ir vertinimo metodus, atlikta analizė atskleidė, kad šiam uždaviniui įgyvendinti taikant tradicinius testavimo būdus testavimo procesui keliami reikalavimai bus tenkinami tik iš dalies. Atkreiptinas dėmesys į tai, jog saityno svetainių ir informacinių sistemų sąsaja pasižymi keletu darbo būdo (saityne *naršoma*) ir techninės aplinkos lemiamų ypatybių [25]. Dėl to formuojasi specifinė, tik tokio pobūdžio sistemoms būdingų sąsajos defektų ir galimų jų sprendimo būdų aibė [9][21].

Problemos aktualumas. Saityno informacinių sistemų grafinės vartotojo sąsajos testavimo metodika aktuali saityno portalus kuriančioms įmonėms, kurios siekia taikant efektyvias grafines vartotojo sąsajos testavimo priemones užtikrinti kuriamo produkto kokybę, išvengiant dažniausiai pasitaikančių grafinės vartotojo sąsajos klaidų ir trūkumų.

Pagrindinis **tyrimo objektas** yra saityno informacinių sistemų grafinės vartotojo sąsajos, sistemos panaudojamumo testavimas.

Tikslas: sukurti prielaidas kokybiškesnių ir geresniais panaudojamumo parametrais pasižyminčių saityno savitarnos svetainių kūrimui, sudarant specializuotą tokio tipo sistemų grafines vartotojo sąsajos testavimo metodiką.

Darbo tikslui pasiekti sprendžiami **uždaviniai:**

1. Apžvelgti sistemų testavimo sąvokas ir dažniausiai pasitaikančias grafinės vartotojo sąsajos (VS) klaidas realizuotose programiniuose produktuose.
2. Ištirti esamus sistemų VS testavimo metodus jų pritaikymo galimybes saityno informacinių sistemų VS testuoti.
3. Išanalizuoti saityno informacinių sistemų VS kokybės problemas ir pasiūlyti VS testavimo metodiką, kuri padėtų aptikti vartotojo sąsajos defektus bei, esant galimybėms, gauti šių defektų šalinimo rekomendacijas.
4. Suformuluoti išvadas ir rekomendacijas tolimesniam metodikos vystymui.
5. Atlikus dalykinės srities analizę apibrėžti pagrindinius VS testavimo metodikos konceptus ir sudaryti dalykinės srities esybių modelį.
6. Pagal atliktą esamų testavimo procesų analizę ir sudarytą VS testavimo metodikos taikymo diagramą, suformuoti VS testavimo etapus, juos detalizuoti.
7. Suprojektuoti ir realizuoti siūlomai metodikai pritaikytą sistemos prototipą.
8. Eksperimentiškai ištirti ir įvertinti siūlomą metodiką ir sistemos prototipą.

Darbo struktūra.

Analitinė dalis. Apžvelgti susiję sprendimai, aptarti testavimo tipų, metodų privalumai ir trūkumai. Analizuojamas saityno informacinių sistemų VS ir sistemos panaudojamumo testavimo procesas ir jo pagrindiniai principai, bei testavimo metodai. Įvairiais aspektais palyginti egzistuojantys funkcinio testavimo metodai ir testavimo procese naudojami įrankiai susiję su testų kūrimu, testuojamų modulių, testavimo atvejų ir klaidų registravimu, sekimu, bei saugojimu duomenų saugyklose. Aptartos dažniausiai pasitaikančios VS ir panaudojamumo klaidos bei išanalizuotos vartotojų savybės, atsakomybės, tikslai ir problemos. Čia taip pat pateikiama testavimo procesų bei priemonių, kurias galima taikyti testavimo metu, apžvalga.

Siūlomo sprendimo metodikos dalis. Antra darbo dalis skirta siūlomai saityno informacinės sistemos grafinės vartotojo sąsajos testavimo metodikai pristatyti. Metodikos sudarymo dalyje pateiktas formalus saityno informacinės sistemos VS testavimo metodikos aprašas ir jos aprašas, kuris paaiškina kaip atlikti grafinės vartotojo sąsajos testavimą vadovaujantis tokia metodika. Pagal sudarytas veiklos procesų ir objektų būsenų kitimų diagramas gali būti kuriami įrankiai, kurie bus panaudoti saityno informacinės sistemos VS testavimo metodikos taikyme.

VS testavimo metodikos realizacijos dalis. Trečioje dalyje pateikiamos metodikos įgyvendinimo ir bei taikymo galimybės. Šioje dalyje pateikiama sudaryta standartinių komponentų aibė ir bazinis testavimo atvejų rinkinys, kurie bus naudojami tyrimo metu kaip standartiniai metodikai

siūlomi rinkiniai. Siūlomai metodikai kuriamas įrankis, kurio pagalba metodikos pritaikymas bus efektyvesnis. Šiame skyriuje pateikiama siūlomo sprendimo realizacijos, sukurtos programinės įrangos prototipo duomenų bazės schema, duomenų bazės naudojamų serverio procedūrų aprašymas.

Eksperimentinio tyrimo dalis. Paskutinėje dalyje, pateikiami realizuoto sprendimo, sukūrus prototipą, eksperimento rezultatai ir jų apibendrinimas. Darbo pabaigoje pateikiamos išvados.

Naudota literatūra. Analizės dalyje daugiausiai naudotasi užsienio ir Lietuvos autorių moksliniais darbais, knygomis, taip pat straipsniuose ir internete pateikta informacija.

Darbo sudėtis. Be šio įvado darbą sudaro keturi dėstymo skyriai, išvados, literatūros sąrašas: 87 puslapiai, jame pateikti 26 paveikslėliai, 16 lentelių. Pabaigoje pridėti du priedai.

1 ŽINOMŲ VARTOTOJO SĄSAJOS TESTAVIMO SPRENDIMŲ ĮVERTINIMAS

1.1. Analizės tikslas ir metodai

Analizės metu siekiama geriau suprasti tyrimo objektą ir sritį, nustatyti vartotojus, jų tikslus ir problemas, bei išanalizuoti esamus problemos sprendimus.

Analizės tikslui pasiekti išskelti uždaviniai:

1. Aptarti sistemos panaudojamumo sąvoką, sudėtį, svarbą ir identifikuoti ryšius su VS.
2. Apibrėžti saityno informacinės sistemos sąvoką, jos sudėtį ir paskirtį.
3. Išanalizuoti saityno informacinių sistemų VS ir sistemos panaudojamumo testavimo procesą ir pagrindinius principus.
4. identifikuoti dažniausiai pasitaikančias VS ir panaudojamumo klaidas, būdingas tiek saityno informacinėms sistemoms, tiek programinėms įrangoms (PI).
5. Išanalizuoti vartotojų tipus, kvalifikacijas, jų atsakomybes, tikslus ir problemas.
6. Išanalizuoti esamus problemos sprendimus, pasiūlytus būtent saityno informacinių sistemų VS testuoti.

Pirmiausia svarbu aptarti kas tai yra sistemų panaudojamumas, iš ko jis susideda, kodėl jis svarbus ir kaip jis siejasi su VS. Apibrėžus saityno informacinės sistemos sąvoką, sudėtį ir paskirtį, bus nustatyta VS svarba ir reikšmingumas tokio tipo informacinėje sistemoje.

Labai svarbu susipažinti su saityno informacinių sistemų VS, sistemos panaudojamumo testavimo vykdymo procesu, pagrindiniais principais, bei dažniausiai pasitaikančiomis VS klaidomis, būdingoms tiek saityno informacinėms sistemoms, tiek tradiciniai programinei įrangai.

Būtina identifikuoti vartotojus, kurie naudosis sukurta testavimo metodika. Išsiaiškinti jų pagrindinius tikslus, bei problemas, kylančias atliekant saityno informacinės sistemos VS testavimą.

Taip pat reikalinga išnagrinėti esamus sprendimus. Analizuojant egzistuojančius metodus taikomus saityno informacinių sistemų VS testuoti, nustatyti jau pasiūlytų VS metodų privalumus ir trūkumus, bei įvertinti jų tinkamumą saityno informacinių sistemų VS testuoti.

Tyrimo objekto, srities ir esamų sprendimų analizei pasirinktas mokslinės literatūros (publikacijos, moksliniai straipsniai, knygos, su sprendžiama problema susijusių metodų specifikacijos ir t.t.). analizės, apibendrinimo ir palyginimo metodai. Mokslinės literatūros analizės ir apibendrinimo metodu siekiama nustatyti dažniausiai pasitaikančias VS klaidas, kurios būdingos saityno informacinėms sistemoms. Esamų ir literatūros šaltiniuose aprašytų VS testavimo metodų analizei

atlikti taikytas palyginimo, interpretavimo moksliniai tyrimai ir eksperimento (dėl eksperimentų dar neaišku kiek tokių prieinamų priemonių bus) metodai.

Tiriant saityno savitarnos svetainių VS testavimo metodikos sudarymo galimybes, vadovautasi autorės patirtimi, įgyta testuojant savitarnos svetainės funkcionalumą ir panaudojamumą.

1.2. Tyrimo sritis ir objektas

Sprendžiamos problemos pagrindinis **tyrimo objektas** yra saityno informacinių sistemų grafinės vartotojo sąsajos, sistemos panaudojamumo testavimas, todėl svarbu atlikti testavimo proceso analizę, siekiant nustatyti tyrimo objekte egzistuojančias problemas.

Vykdam vieną brangiausių ir daugiausiai darbo išteklių reikalaujančių programinės įrangos kūrimo proceso etapų – testavimą – paprastai svarbiausi uždaviniai yra įvertinti, atlikti patikrą (angl. *verification*) ar sistema atitinka specifikacijoje numatytus reikalavimus, ir ar ji atitinka vartotojo poreikius, ar tai, ką daro sistema, yra teisinga (atestavimas, angl. *validation*). Grafinė vartotojo sąsaja (angl. *GUI - graphical user interface*) dažnai testuojama minimaliai.

Taigi, darbo **tyrimo sritis** – programinės įrangos kokybės testavimo metodai. Sistemų grafinės sąsajai testuoti taikomos įvairios metodikos, principai, bet pagrindinis dėmesys bus skiriamas testavimo metodams, tinkantiems būtent saityno informacinės sistemos grafinei vartotojo sąsajai testuoti.

1.3. Saityno savitarnos svetainė

Saityno savitarnos svetainė pasirinkta kaip pavyzdinė svetainė, kuriai bus pritaikyta kuriama VS testavimo metodika. Saityno savitarnos svetainės plačiai naudojamos ryšio su klientais valdymui. Saityno savitarnos svetainės yra elektroninės paramos savitarnos, kurios suteikia klientams tiesioginę prieigą prie informacijos ir galimybę atlikti kasdienes operacijas per internetą bet kuriuo paros metu. Darbas su savitarnos svetaine nereikalauja jokio įmonės atstovo įsikišimo, todėl labai svarbus tinkamos ir korektiškos sąsajos užtikrinimas tarp vartotojo ir sistemos. Siekiant, kad savitarnos svetainė, turėtų teigiamą naudą jos vartotojui, ji turi pasižymėti funkcionalumu, informatyvumu, geru dizainu, navigacija ir suderinamumu su skirtingomis vartotojo naršyklėmis.

Saityno savitarnos svetainei būtinas nuoseklus stilius, kad svetainė veiktų be klaidų, patiktų ir būtų patraukli vartotojui. Galiausiai, saityno savitarnos svetainės sėkmė priklauso nuo esamos informacijos kokybės ir nuo to kaip paprastai ji gali būti prieinama. Visos šios savitarnos svetainės

savybės ne tiek svarbios vartotojų išlaikymui, bet kiek sužadinti vartotojų norą naudotis saityno savitarnos svetaine.

1.4. Sistemos panaudojamumas ir jo ryšys su vartotojo sąsaja

ISO/IEC 9126-1 išskiria 6 sistemos kokybės charakteristikas:

- Funkcionalumas;
- Patikimumas (*angl. reliability*);
- Panaudojamumas (*angl. usability*);
- Veiksmingumas (*angl. efficiency*);
- Palaikomumas (*angl. maintainability*);
- Perkeliamumas (*angl. portability*).

Viena svarbiausių charakteristikų, sistemos panaudojamumas, parodo kaip lengvai vartotojui gali būti suprantama ir naudojama vartotojo sąsaja [12]. Šis sistemos kokybės rodiklis paprastai nenurodomas reikalavimų specifikacijoje. Vartojimo patogumas – tai naudotojo veiklos veiksmingumas, našumas ir jaučiamas pasitenkinimas, su kuriuo konkretus naudotojas gali pasiekti konkrečių tikslų konkrečiose aplinkose [7]. Panaudojamumą sudaro penkios pagrindinės sub-charakteristikos [6]:

- suprantamumas (*angl. understandability*);
- veiksmingumas (*angl. operability*);
- patrauklumas (*angl. attractiveness*);
- naudingumas (*angl. usability compliance*) – apibudina sistemos išvaizdos funkcionalumą;
- išmokstamumas (*angl. learnability*) – kaip lengvai vartotojai atlieka užduotis pirmą kartą susidūrę su sistemos išvaizda;
- įsimenamumas (*angl. memorability*) – kaip greitai vartotojas gali adaptuotis prie sistemos išvaizdos po ilgesnio laiko tarpo;
- pasitenkinimas (*angl. satisfaction*) – kiek vartotojui maloni sistemos išvaizda [12].

Sistemos panaudojamumą sudaro ir daugelis kitų ne mažiau reikšmingų kokybinių komponentų. Sistemos vartotojui aktualu, kad tai kas jam reikalinga, būtų ir lengvai suprantama. Sistemos vartotojui jokios naudos iš to, kas lengvai suprantama, bet nenaudinga funkciškai. Klaidų kiekis ir jų

sudėtingumas, kurias gali padaryti vartotojas, taip pat vienas iš galimų panaudojamumo įvertinimo komponentų. Įvertinus su kokio sudėtingumo klaidomis susiduria sistemos vartotojas ir kaip lengvai vartotojas pats gali jas išspręsti, galima spręsti apie sistemos panaudojamumo kokybę.

Sistemos panaudojamumas, ypač internete, yra vienas iš svarbiausių sistemos populiarumo ir priimtino vartotojui aspektų. Šie aspektai labai dažnai lemia sėkmę rinkoje, nes konkurencija tarp interneto puslapių labai didelė. Prasta sistemos vartotojo išvaizda sumažina naudotojo veiklos našumą, didina mokymosi laiką ir klaidų skaičių. Visi šie faktoriai iššaukia naudotojo nusivylimą ir galimą atsisakymą naudoti programų sistema [11]. Vartotojas susidūręs su sistemos panaudojimo sunkumais vengia juos spręsti, todėl dažniausiai imasi kitų būdų norimai situacijai išspręsti. Dėl sistemos panaudojamumo neefektyvumo, jos vartotojai gali sistemą labai greitai ignoruoti ir pakeisti ją kita, labiau draugiškesnę vartotojui sąsają turinčia sistema.

Panaudojamumas atlieka svarbų vaidmenį kiekviename sistemos projektavimo proceso etape. Sistemos VS projektavimas atliekamas atsižvelgiant į apribojimus: sukurta sistema turi būti realizuotina, projektavimo procesas turi neviršyti biudžeto ir sistema turi veikti. Panaudojamumas yra taip pat svarbus egzistuojantis rodiklis į kurį būtina atsižvelgti projektuojant VS. Testuojant sistemos panaudojamumą siekiama patikrinti sistemos veikimą ir patogumą galutiniam vartotojui. VS – tai vartotojui matoma sistemos dalis, kuri ir sudaro didžiąją dalį sistemos panaudojamumo. Panaudojamumo testavimo tikslui pasiekti dažniausiai yra testuojama VS, o funkcionalumo testavimas dažnai padengia VS ir panaudojamumo charakteristikas. Vartotojo sąsają ženkiai veikia panaudojamumo charakteristikos. Tinkamai sutvarkytas panaudojamumas, daro didelę teigiamą įtaką vartotojo sąsajai.

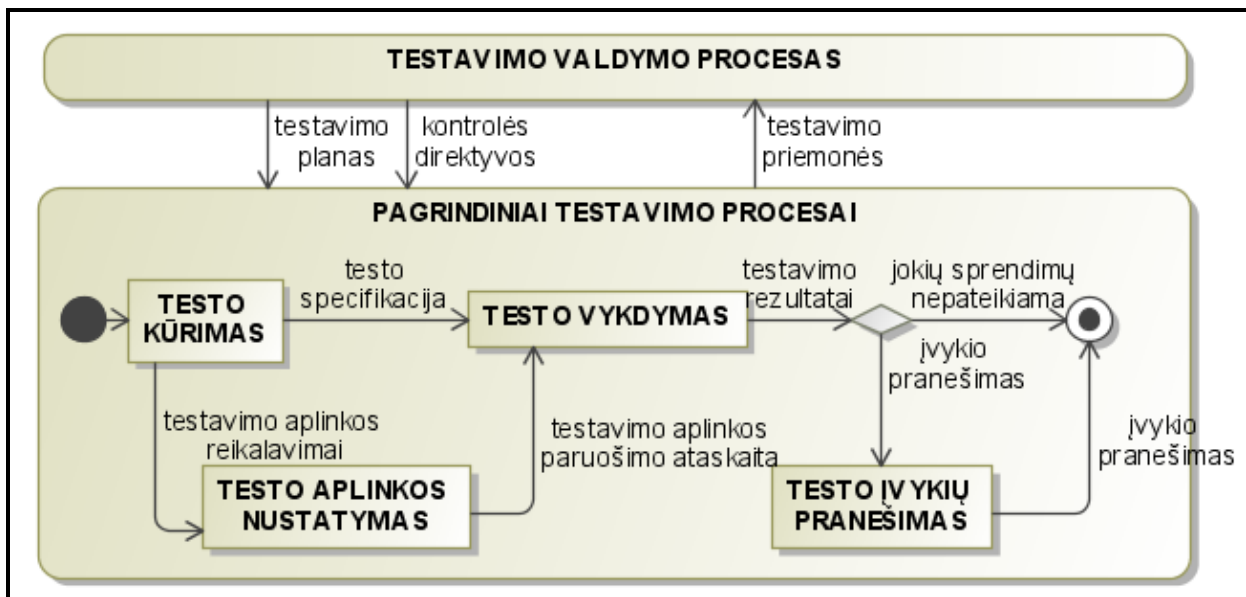
1.5. Programinės įrangos testavimas

Testavimas - tai bet kokia veikla, kurios tikslas - įvertinti programos sistemos charakteristiką ar gebėjimus, bei nustatyti ar ji tinkama reikalaujamiems rezultatams gauti [5]. Bendras testavimo tikslas (PĮ inžinerijoje) yra nustatyti, ar kuriamas produktas gali būti sėkmingai panaudotas pagal jo kūrimo paskirtį. Testavimas paprastai atliekamas šiais tikslais [2]:

- siekiant pagerinti kokybę;
- verifikuoti ir validuoti;
- įvertinti patikimumą.

Žemesnio lygio potiksliai – rasti defektus ir užtikrinti, kad PĮ veiks be sutrikimų.

Testavimas yra sudėtinis procesas, kuris susideda iš dviejų stambių procesų. Testavimą sudaro testavimo valdymo ir pagrindinis testavimo procesai [23] (1 pav.).



Šaltinis: ISO/IEC 29119 Software Testing (2009) Fundamental Test Process [23].

1 pav. Testavimo proceso pagrindas

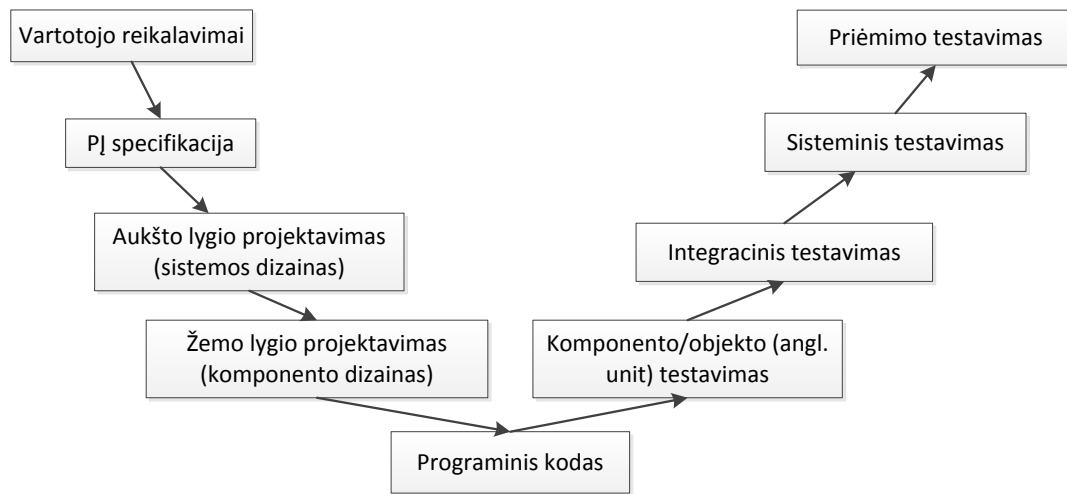
Sudėtinių testavimo procesų vykdymas ir veiklos sudėtis priklauso nuo pasirinktos testavimo technikos, metodų ir priemonių, kurios gali būti parenkamos atsižvelgiant į testuojamą objektą ir siekiamus rezultatus.

Sistemų vartotojo sąsajai testuoti reikalinga unikali metodika, tačiau jos įgyvendinimas turi būti grindžiamas tradicinės programinės įrangos funkcinio testavimo eiga [10]. Galima daryti prielaidą, jog, remiantis standartiniu programinės įrangos testavimo proceso modeliu, gali būti atliktas ir saityno informacinės sistemos tiek funkcinis, tiek vartotojo sąsajos ir panaudojamumo testavimas, tačiau būtina šį procesą pritaikyti ir detaliau apibrėžti. Specializavus procesą saityno informacinės sistemos grafines vartotojo sąsajai testuoti, bus sukurtos prielaidos tokio tipo sistemų testavimo efektyvumui (rastų defektų skaičius/esamų defektų skaičius) ir produktyvumui (defektų skaičiaus radimas per kuo trumpesnę laiką) padidinti.

Testavimo procesus būtina tobulinti ir pritaikyti VS testuoti, kad būtų įgyvendinamas efektyvumas (rastų defektų skaičius/esamų defektų skaičius) ir produktyvumas (defektų skaičiaus radimas per kuo trumpesnę laiką). Egzistuojančius PĮ testavimo procesus reikia taikyti ir saityno informacinės sistemos VS testavime. PĮ testavimo procesai turi būti patobulinti siekiant juos efektyviai pritaikyti vartotojo sąsajos ir sistemos vartojimo patogumui testuoti. Specializuojant testavimo procesą,

gali būti parenkamos skirtingos testavimo priemonės, būdai ir technikos, atsižvelgiant į tai, ko siekiama testavimu.

Keičiantis pagrindinio testavimo proceso sub-procesams, keičiasi ir testavimo tikslai. Atliekant testų vykdymą (angl. *test execution*) jo eigoje nuolat kinta ir testavimo objektas. Sistemos testų vykdymas prasideda nuo modulio (vieneto), atskirų sistemos dalių ar komponentų testavimo. Geriausiai ryšį tarp kiekvieno programinės įrangos kūrimo ciklo ir su juo susijusio testavimo etapo parodo Paul Rook pasiūlytas „V modelis“ [28] 73 (2 pav.).



2 pav. „V modelis“. Testavimo objekto kitimas testuojant sistemą.

VS testavimas gali būti klasifikuojamas į keturis etapus [15]: žemo lygio, aplikacijų, integracijos ir nefunkcinių klaidų testavimą (1 lentelė).

Lentelė 1. VS testavimo etapai

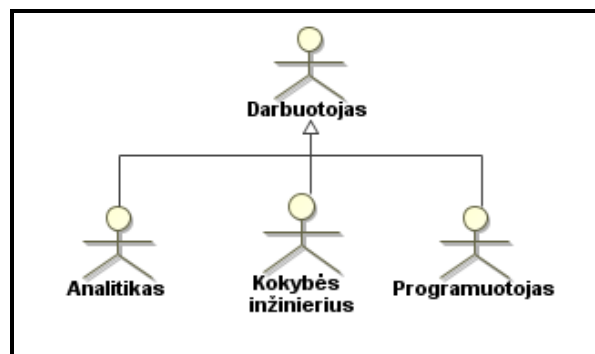
Žemo lygio testavimas (angl. Low Level)	Komponentų sąrašo testai (angl checklist testing)
	Navigacijos (angl. navigation)
Taikomosios programos testavimas (angl. Application)	Ekvivalentaus sudalinimo (angl. equivalence partitioning)
	Ribinių reikšmių (angl. boundary values)
	Sprendimų priėmimo lentelės (angl decision tables)
	Būsenų perėjimo (angl state transition testing)
Integracijos testavimas (angl .Integration)	Darbastalio integracijos (angl desktop integration)
	Komunikacijos (angl. communication)
	Sinchronizacijos (angl. Synchronization)
Nefunkcinis testavimas (angl Non –Functional)	Sistemos patvarumo (angl. soal testing)
	Suderinamumo (angl. Sompatibility testing)
	Aplinkos (angl. platform/environment)

Šaltinis: sudaryta autorės pagal Proposed GUI test stages [15]

Žemo lygio testavimas – komponentinis testavimas. Testuojamos smulčiausios sudėtinės sistemos dalys. Sistemos testavimo metu yra testuojama atskirų modulių sąveika tarpusavyje ir ar sistema tenkina specifikacijoje keliamus reikalavimus. Sistemos sąsajos su kitomis kompiuterinėmis sistemomis testavimas atliekamas sistemos integracijos testavimo metu. Priėmimo testavimas atliekamas paskutiniame testavimo etape. Įvertinant tai, jog defektų aptikimas ankstyvose kūrimo stadijose padeda sumažinti jų sprendimo sąnaudas [29], vartotojo sąsają testuoti būtų galima pradėti jau nuo atskiro komponento testavimo. Taip pat turėtų būti palikta galimybė taikyti testus ir paskutiniuose programinės įrangos kūrimo etapuose arba realizuotai sistemai.

1.5.1. Testavimo procese dalyvaujantys veikėjai ir ryšiai tarp jų

Testavimas išskiriamas kaip atskiras procesas. Testavimo procese dažniausiai dalyvaujantys, pagrindiniai yra trijų sričių specialistai (3 pav.), tarp kurių yra glaudus darbinis informacinis ryšys.



Šaltinis: sudaryta autorės.

3 pav. Testavimo procese dalyvaujantys veikėjai

Testavimo procese dalyvaujantis ir pagrindinis jo vykdytojas yra kokybės inžinierius. Kokybės inžinierius, iš kitų testavimo procese dalyvaujančių veikėjų, gauna visą svarbią ir testavimui reikalingą informaciją, kurią panaudoja testavimo procese, bei dalinasi ja su analitiku ir programuotoju. Tokia informacija gali būti verslo poreikiai, funkciniai ir nefunkciniai reikalavimai, sistemos prototipų langai, iš programuotojų ir patys prototipai. Ši medžiaga dažniausiai yra naudojama prieš pasiruošiant testavimui, o kartais ir testavimo užduočių vykdymo metu.

Prie skirtingų testavimo sudedamųjų užduočių dalyvauja skirtingi specialistai, įnešantys savo darbinį indelį atliekant užduotis. Testavimo procese dalyvaujančių specialistų dalyvavimas skirtinguose testavimo užduotyse pateiktas 2 lentelėje.

Lentelė 2. Testavimo veikėjų dalyvavimas užduočių vykdyme

Užduotis	Aprašymas	Kokybės inžinierius	Analitikas	Programuotojas
Testų sudarymas ir įgyvendinimas (<i>angl. test design&implementation</i>)	Sistemos reikalavimų analizė; optimaliausio sistemos testavimo būdo parinkimas; testavimo duomenų surinkimas; sistemos testavimo specifikacijos sukūrimas.	+	+	+
Testų vykdymas (<i>angl. test execution</i>)	Sistemos testavimo atlikimas pagal testavimo specifikaciją.	+	-	+
Testavimo komponentų kūrimas	Testavimo komponentų sukūrimas pagal sistemos testavimo specifikaciją.	+	-	+

Šaltinis: sudarytas autorės

Kuriant saityno informacinių sistemų VS testavimo metodiką labai svarbu atsižvelgti į visų testavimo procese dalyvaujančių specialistų įtaką testavimo procesui.

1.6. Dažniausiai pasitaikančios VS ir panaudojamumo klaidos

Problemos, susijusios su saityno informacinių sistemų VS testavimo procesu, kyla dėl metodikų nebuvimo arba jų nepritaikymo būtent VS testuoti. Saityno informacinės sistemos VS testavimas vyksta neefektyviai, nes testavimo procesas nėra papildytas sub-procesais apimančiais VS testavimą. To pasėkoje lieka daug VS ir panaudojamumo klaidų sistemoje.

VS klaidos, kurios yra būdingos programinei įrangai, ne visada pasitaiko saityno informacinėse sistemose. Saityno informacinėse sistemose pasitaiko tik joms būdingos ir specifinės klaidos.

Jeff Johnson išskiria tokias saityno VS klaidų grupes [21]:

- turinio klaidos (*angl. content*);
- užduoties palaikymo klaidos (*angl. task-support*);
- navigacijos klaidos;
- formų klaidos;
- paieškos klaidos;
- teksto ir rašymo klaidos;
- nuorodų išvaizdos klaidos;
- grafikos ir išdėstymo klaidos.

Dvi dažniausiai saityno informacinėms sistemoms būdingos klaidos yra susijusios su tekstu ir spalvomis. Viena iš tokių VS klaidų yra teksto esančio ant spalvinio fono skaitymo sunkumai. Šios klaidos keliamas nepatogumas yra toks, kad net ir didelio šrifto tekstas ant margo arba su teksto spalva nekontrastuojančio fono, yra sunkiai skaitomas. Išskiriamas trys pagrindiniai klaidos atvejai [9]:

- a) tekstas ant margo fono;
- b) teksto spalva nekontrastinga fono spalvai;
- c) tekstui ir fonui panaudojamos dvi konkuruojančios spalvos.

Kita klaida dažniausiai pasitaikanti saityno informacinėse sistemose – spalvinių skirtumų nesilaikymas. Ši klaida dažniausiai pasitaiko tada, kai nepaisoma taisyklės, kad geriausiai produktas atrodo jeigu jam yra naudojamos 2-3 spalvos, bet ne daugiau. Saityno informacinėse sistemose, kuriose pateikiami dideli informacijos kiekiai, ir stengiamasi vartotojui kuo aiškiau nurodyti, kokius veiksmus jam atlikti, gausaus kiekio spalvų panaudojamumo galimybė neretai tampa sunkiai išvengiama.

Šių VS klaidų, pasitaikančių saityno informacinėse sistemose, galima išvengti sudarius ir vadovaujantis saityno informacinės sistemos/prekės ženklo knyga (angl. *brand book*). Tokio koncentruoto VS žodyno ir taisyklių rinkinio trūkumas arba nesivadovavimas juo, dažniausiai ir iššaukia kritines situacijas programiniuose produktuose.

1.7. Vartotojų aibė, tipai ir savybės

Kuriamos saityno savitarnos svetainių VS testavimo metodikos vartotojai – tai kokybės inžinieriai ir eiliniai testuotojai, kurie užtikrina PĮ kokybę. Testavimo komandoje, tarp VS metodikos vartotojų, testavimo profesionalai sudarys tik mažumą. Kuriant testavimo metodiką būtina atsižvelgti į tai, kad eilinių testuotojų patirtis mažesnė, nei kokybės inžinierių.

Kokybės inžinieriai – tai savo srities specialistai, turintys didelę PĮ testavimo patirtį, puikiai išmanantys PĮ testavimo technologijas ir metodus, testavimo atvejų rašymo specifikacijas bei testavimo procesą visose PĮ kūrimo stadijose. Tokie vartotojai yra aukštos kvalifikacijos, sertifikuoti, turi “ISEB Foundation in Software Testing” arba lygiaverčius sertifikatus ir yra išklause specializuotus PĮ kokybės užtikrinimo kursus.

Įmonėse, kurios turi pilnai suformuotą programinės įrangos kūrimo komandą, tokių specialistų, testuotojų, būna bent vienas.

Kokybės inžinierių atsakomybė yra labai didelė, nes testavimas yra bene svarbiausias PĮ kūrimo etapas. Kuo klaida ankščiau pastebėta, tuo pigiau atsieina jos taisymas. Paskutinėje testavimo stadijoje problemos ištaisymas yra pats brangiausias, o kai PĮ klaida yra aptinkama PĮ įdiegus, klaidos taisymas

reikalauja dar daugiau išteklių. Kokybės inžinierius duoda garantiją, kad produktas yra kokybiškas ir tinkamas naudojimui.

1.7.1. Vartotojų tikslai ir problemos

Norint pasiekti globalų PĮ testavimo tikslą, būtina pasiekti jį sudarančius potikslius. PĮ testavimo potiksliai, tai kokybės inžinierių siekiami įgyvendinti tikslai testavimo proceso metu. Kokybės inžinieriai atlikdami testavimą turi konkrečius tikslus, kuriuos bandoma pasiekti atliekant testavimo uždavinius.

Kokybės inžinierių tikslai testavimo proceso metu:

- parodyti, kad programinės įrangos produktas teisingai atlieka visas numatytas funkcijas;
- darbo produkte rasti visas įmanomas klaidas (angl. *bugs*), trūkumus (angl. *faults*), defektus (angl. *defects*), silpnumus;
- sąmoningai mėginti elgtis neteisingai, kad nustatyti, ar programa veikia taip, kaip turėtų.

Visose PĮ kūrimo stadijose, testavimo proceso stadijose kokybės inžinieriai susiduria su problemomis. Atliekant saityno informacinių sistemų testavimą dažniausiai su problemomis susiduriama dėl to, kad specialistų, testuotojų, kvalifikacija būna keliama funkcinio testavimo srityje ir didžiąją dalį įgytos patirties taip pat sudaro PĮ produkto funkcijų testavimo patirtis. Nustatytos problemos pateiktos 3 lentelėje.

Lentelė 3. Problemos su kuriomis susiduria vartotojai testuojant VS

Problema	Aprašymas
Nėra specializuotos testavimo metodikos saityno savitarnos svetainių VS testuoti	Kiekvieną kartą saityno informacinės sistemos VS testavimas atliekamas pagal kokybės inžinieriaus sugalvotą scenarijų.
PĮ testavimo metodai nėra pritaikyti saityno informacinėms sistemoms testuoti	Testuotojai taiko universalias PĮ testavimo metodikas saityno informacinėms sistemoms testuoti.
Saityno savitarnos svetainių VS ir panaudojamumas testuojamas minimaliai	Nesant saityno VS testavimo technologijų ir metodų saityno VS testavimas atliekamas įvertinant tik pagrindinius VS aspektus.
Saityno savitarnos svetainių VS testavimas yra neefektyvus	Taikant egzistuojančias testavimo metodikas, kurie yra nespecializuoti ir nepritaikyti saityno savitarnos svetainėms testuoti, nėra užtikrinama geriausia saityno savitarnos svetainės kokybė.
VS testavimo efektyvumas priklauso nuo testuotojo patirties	VS testuoti nesudarytos arba nesivadovaujama prekinio ženklo knygomis (angl. <i>brand-book</i>). Dažniausiai VS testavimas priklauso nuo pačios testuotojo kompetencijos ir žinių, nes nėra sudaromi taisyklių/reikalavimų rinkiniai pagal kuriuos būtų galima nespręsti VS korektiškumą.

Šaltinis: sudaryta autorės.

Sukurta saityno savitarnos svetainių VS testavimo metodikas šias problemas padės išspręsti. Metodika apjungs žinomus sprendimus, o konkrečioms problemoms, su kuriomis susiduria vartotojai atlikdami VS testavimą, bus pritaikyti specifiniai sprendimai joms spręsti.

1.8. Egzistuojančios priemonės VS testuoti

Svarbu nuodugniai išnagrinėti ir įvertinti, jau egzistuojančius PĮ funkcionalumo ir VS testavimo sprendimus, tinkamus saityno IS vartotojo sąsajai testuoti.

Vienas dažniausiai naudojamų būdų, vartotojo sąsajai ir sistemos vartojimo patogumui testuoti yra VS kontroliniai sąrašai (angl. *checklist*). Kontroliniai sąrašai sudaromi atskiriems sistemos komponentams testuoti. Labai svarbu, kad VS testavimo kontrolinio sąrašo punktai, padengtų atskirų panaudojamumo sub-charakteristikų dedamąsias dalis. Tokie testai nurodo tikslus, kuriuos reikia pasiekti, o ne veiksmų sekas, kurias reikia atlikti.

Kitas labai dažnai naudojamas testų tipas VS prototipai. Pirmiausi yra sukuriami aukštos kokybės prototipas su imituotu funkcionalumu, kiekvienam VS komponentui. Vadovaujantis šiais sukurtais prototipais atliekamas kelis kartus kartojamas atskirų komponentų testavimas. VS komponentų prototipų kūrime dalyvauja skirtingų sričių specialistai. VS prototipų sudaryme dalyvauja asmenys atsakingi už dizaino kūrimą ir yra įtraukiami asmenys atsakingi už būsimos sistemos vartojimo patogumo ir VS kokybę.

VS testavimo atvejuose (angl. *test cases*) būtina apsibrėžti testavimo objektus ir tikslus, todėl labai dažnai testavimas atliekamas naudojantis testavimo atvejais. Testavimo atvejis yra sudarytas iš aibės jėjimo duomenų, veiksmų ir tikėtinų rezultatų aibės. Visi testavimo atvejai sudaromi pagal funkcinis reikalavimus, todėl testavimo atveju siekiama patikrinti sistemos atitikimą specifikacijoje keliamiems reikalavimams. VS testuoti testavimo atvejai taikomi kartu su kitais pagalbinais metodais, pavyzdžiui kontroliniais komponentų sąrašais ar prototipais.

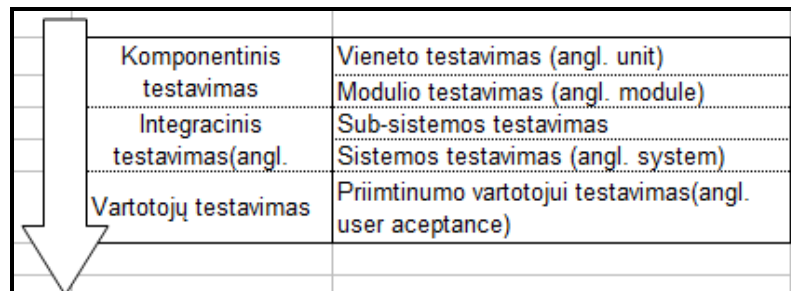
1.9. Egzistuojantys VS testavimo būdai ir metodai

Pasirinkti tokie VS testavimo būdai ir metodai, kurie plačiai taikomi panašioms problemoms, susijusioms su VS testavimu spręsti.

1.9.1. Rankinis VS testavimo būdas

Rankinis (angl. *manual*) testavimas yra seniausias ir griežčiausio tipo PĮ testavimo metodas, kuris VS testavime taikomas dažniausiai.

Metodas daugiausiai naudojamas siekiant aptikti vartojimo patogumo (angl. *usability*) klaidas, nes pagrindinės ir kritiškiausios VS klaidos randamos testuojant rankiniu būdu. Rankinis VS testavimas, kaip ir funkcinis testavimas sudarytas iš kelių etapų (4 pav.).



Komponentinis testavimas	Vieneto testavimas (angl. unit) Modulio testavimas (angl. module)
Integracinis testavimas(angl.	Sub-sistemos testavimas Sistemos testavimas (angl. system)
Vartotojų testavimas	Priimtino vartotojui testavimas(angl. user acceptance)

Šaltinis: sudaryta autorės.

4 pav. Rankinio testavimo etapai

VS testavimas pradedamas nuo smulkių sistemos dalių testavimo, modulių testavimo. Testuojant šiuo būdu, atliekant komponentinį ir integracinį testavimą yra naudojami prototipai, kontroliniai sąrašai arba testavimo atvejai, o kartais rankinis testavimo metodas derinamas su automatinio. Paskutinis testavimo etapas yra susijęs su galutinės sistemos priimtumu, patogumu ir vartotojiškumu. Šiame testavimo etape testavimą atlieka ir sistemą įvertina sistemos vartotojai, o ne sistemos kūrėjai ar testuotojai.

Rankinis testavimas yra sudėtingas ir labai daug nuo testuotojo patirties ir savybių priklausantis procesas. Taikant šį VS testavimo metodą, pats testavimo procesas nėra sistemingas ir neužtikrina visų testavimo atvejų padengiamumo, o įvykus pasikeitimams VS, testas turi būti atliktas dar kartą.

1.9.2. Tradicinių PĮ funkcionalumo testavimo metodų taikymas VS testuoti

Programinės įrangos VS testavimas jau yra įgyvendinamas ir tam skiriama vis daugiau lėšų – išlaidos sudaro nuo 50 iki 60 procentų visų PĮ plėtros išlaidų. VS testavimas toliau kelia sunkumus, parodydamas, kad tradicinės PĮ testavimo technikos daro nepakankamą įtaką VS testavime [10].

Egzistuojantys VS testavimo metodai yra neišsamūs ir daugiausia rankiniai [10]. VS testavimo praktika žymiai skiriasi nuo PĮ testavimo. VS testavimas aprėpia daug skirtingų atvejų, kurie labai skiriasi nuo bendros PĮ testavimo praktikos. Daug tradicinių testavimo metodų yra bandoma taikyti VS testuoti, tačiau efektyvumo lygis nėra toks aukštas kokio tikimasi ir siekiama.

1.9.3. Atsitiktinės įvesties VS testavimo metodas

Atsitiktinės įvesties (angl. *random input*) testavimo metodas taip pat žinomas kaip stochastinis ar beždžionės testas [18]. Šis VS testavimo metodas grindžiamas atsitiktinės sąsajos su kompiuteriu, todėl jo visų galimų testavimo atvejų padengiamumas nėra didelis.

Taikomi dviejų tipų beždžionių testų tipai [18]:

- kvailių (angl. *dumb*) beždžionių;
- sumanių (angl. *smart*) beždžionių.

Taikant šį savalaikišką metodą yra sunku užtikrinti jo pakartojamumą, yra brangus ir reikalauja didelių laiko išteklių, o ir testavimo atvejų padengiamumas nėra pakankamas (Nyman, 2000).

1.9.4. „Įrašymo atkūrimo“ VS testavimo metodas

VS testuoti dažniausiai naudojami įrankiai remiasi įrašų atkūrimo (angl. *capture/replay*) technika [10]. „Įrašymo atkūrimo“ metodas – daug darbo reikalaujanti technika, kuri įrašo VS navigacijos kelią. Tokios testavimo technikos veikimas grindžiamas principu, kad testų kūrėjas sąveikauja su VS, generuodamas pelės ir klaviatūros įvykius. Įrankis įrašo vartotojo veiksmus, išsaugo VS sesijos vaizdus ir kaupia jos saugykloje, kaip scenarijų, kurį kaip testą galima naudoti kelis kartus. Toks sudarytas scenarijus naudojamas kelis kartus, atkuriant įrašytą sesiją ir nurodant vis skirtingus įėjimo kintamuosius.

Automatiniai VS testai nėra visiškai tinkami VS testuoti. Prieš pradėdant testuoti naudojant automatinius testus, jie turi būti tinkamai sudaryti, apgalvojant visus galimus testavimo atvejus. Tokių testų sukūrimas reikalauja didelio įdirbio prieš pradėdant testuoti. Automatinių testų atvejų generatorius

gali padėti pasiekti aukšto lygio rezultatus, tačiau jų efektyvumo sudarymas priklauso nuo testų kūrėjo, programuotojo, įgūdžių sąveikaujant su VS. Programuotojas, testų kūrėjas, turi sukurti kodą visiems galimiems sprendimo taškams VS, nes dėl neplanuotų ar neapgalvotų įvykių sudėtinga parašyti išbaigtus testavimo scenarijus.

Įrašų atkūrimo testavimo metodas yra gerai žinomas ir vienas iš populiariausių automatinių VS testavimo metodų, tačiau jo pagrindinis trūkumas yra jautrumas pasikeitimams testuojamoje sistemos dalyje. Dar nėra sukurto ar įgyvendinto pavyzdinio šablono kaip atlikti VS testavimą. pakeitimams ir priklausomybė nuo platformos, verčia ieškoti efektyvesnių VS testavimo metodų. Automatiniai testai dažnai duoda didelę naudą, kai testuojamas produktas nekinta ir turi konkretų scenarijų, todėl pakartotinai naudoti šio tipo testus yra beveik neįmanoma. Taip pat ši metodika gali būti taikoma tik tada kai PĮ kūrimas jau yra pasibaigęs. Automatiniai testai tik sąlyginai yra pranašesni už rankinius, nes pagrindinės ir kritiškiausios VS klaidos randamos testuojant rankiniu būdu.

1.9.5. Išbaigtų sąveikų sekų VS testavimo metodas

Išbaigtų sąveikų sekų VS testavimo metodas grindžiamas dažniausiai testavime naudojamasi testavimo atvejais, kurie šiame metode turi būti pili ir išbaigti. Siekiant, kad testavimo atvejai būtų išbaigti ir suprantami, reikalinga nurodyti įvesties apibūdinimą ir tikimosi sistemos išvesties rezultato tai įvesčiai.

VS testų įvestys dažniausia aprašomos kaip VS objektų aktyvavimo ar pasirinkimo sekos, kurios interaktyviai veikia su objektais. CIS (angl. *complete interaction sequences*) – išbaigtų sąveikų sekų metodas, kuris ir yra orientuotas į VS objektus, kurie yra veikiami vartotojo veiksmų sekos [19]. Vadovaujantis CIS yra kuriamas baigtinių būsenų modelis (angl. *Finite State Model*), kuriam sukurti reikalinga didelė patirtis, kad sukurtas modelis tiktų visiems išbaigtų sekų atvejams. Sukurtas modelis panaudojamas testų generavimui, kurie vėliau ir naudojami automatiniam testavimui.

1.9.6. Pažintinės peržvalgos VS vertinimo metodas

Pažintinė peržvalga – tai vartotojo sąsajos įvertinimo metodas, kurio centre yra specifikuojant reikalavimus apibrėžtų vartotojų užduočių vykdymas [25]. Šis būdas leidžia kuo anksčiau gauti dizaino sprendimo įvertinimus [26] ir jau projektavimo etape įvertinti specifiškai užduočių vykdymo sekas. Kuo anksčiau dizaino klaidos aptinkamos, tuo lengviau jas spręsti. Vis dėlto taikant metodą ankstyvose sistemos kūrimo stadijose, naudojant grafinius ar programinius sąsajos prototipus, neatsižvelgiama į

galimus interaktyvius vartotojo veiksmus [26]. Todėl išskyla poreikis imituoti interaktyvius procesus, įtraukiant ir specifines užduotis. Toks pasiruošimas testuoti ir testavimas vadovaujantis formaliu metodo aprašu reikalauja intensyvaus darbo. O taikant tik supaprastintas priemones gaunama paviršutiniška ir siaura analizė, apimanti tik teksto ir grafinio dizaino, sąsajos elementų išdėstymo problemas.

1.9.7. Atitikties rekomendacijoms ir standartinio vertinimo metodas

Atitikties rekomendacijoms ir standartinio vertinimo metodas – tai sąsajos peržiūra, kuri atliekama vadovaujantis iš anksto griežtai apibrėžtomis rekomendacijomis. Rekomendacijose labai tiksliai nurodomi reikalavimai arba taisyklės, kurių reikėtų laikytis kuriant konkrečius sąsajos elementus [25]. Vadovaujantis šiomis rekomendacijomis, aptinkamos bendrinės ir dažniausiai pasikartojančios klaidos, taip pat įvertinami ne tik teksto ir grafinių vaizdų defektai, bet ir dizaino ar pačių objektų išdėstymo problemos. Defektų šalinimo sprendimų priėmimas taip pat vykdomas vadovaujantis rekomendacijomis. Vis dėlto ne visos situacijos gali būti numatytos ar apibrėžtos rekomendacijose, tarptautiniuose arba įmonių naudojamuose standartų rinkiniuose.

1.9.8. Euristinis vertinimo metodas

Euristinis vertinimas – tai kruopštus sistemos vertinimas, kurį atlieka vartotojo sąsajos specialistų ar kvalifikuotų IS kūrimo komandos nariai [25]. Taikant šį metodą vertinama pagal asmeninę patirtį, bet dažniau pagal tai, kaip sistema atitinka žinomus panaudojamumo principų rinkinius – euristikas. Euristikos gali būti ne tik universalios, bet ir:

- pritaikytos konkrečiam tipo sistemoms: stacionarioms IS, internetinėms IS, mobilioms IS, medicininėms IS, savitarnos svetainėms ir t.t.;
- pritaikytos konkrečiam projektui, pvz., kuriamai miškotvarkos instituto informacinei sistemai.

Tokios universalios, dinaminės euristikos, kurios būdingos tik konkrečiam testuojama produktui, turi būti papildomai įtraukiamos į universalių euristikų aibę kiekvieno projekto pradžioje.

Saityno svetainių/sistemų panaudojamumo euristikos pagal (Gaffney G. Usability evaluation checklist for web sites. Infodesign, 1998).

- **Navigacija:** yra aiškiai nurodyta, kurioje svetainės vietoje yra vartotojas, yra nuoroda į pradinį puslapį, svetainės struktūra yra paprasta ir kt.;

- **Funkcionalumas:** visas funkcionalumas yra aiškiai pažymėtas, prieinamas nepaliekant tinklapio, nėra nereikalingų įskiepių;
- **Valdymas:** vartotojas gali atšaukti operacijas, kiekviename puslapyje yra aiškus išėjimo ženklas, palaikomos įvairios naršyklės ir kt.;
- **Kalba:** vengiama žargono vartojimo, naudojama lengvai suprantama kalba;
- **Grižtamasis ryšys:** visada yra aišku, kas vyksta tinklapyje, vartotojai gali gauti informaciją el. paštu, pagalba yra prieinama ir kt.;
- **Nuoseklumas:** kiekvienas dalykas sistemoje turi tik vieną pavadinimą, nuorodoms naudojamos standartinės spalvos, terminija yra nuosekli ir kt.
- **Klaidų vengimas ir taisymas:** klaidos neatsiranda be reikalo, klaidų pranešimai nurodo, ką reikia atlikti, klaidų pranešimai nurodo tikslų išėjimo tašką ir kt.;
- **Matymas:** išdėstymas yra aiškus, visi paveikslai turi alternatyvų tekstą, yra vengiama nereikalingos animacijos ir kt.

Euristikos dažniausiai pateikiamos kaip atvirų kontrolinių klausimų ar pasiūlymų sąrašas. Šis sąrašas nėra veiksmų, kuriuos turi atlikti testuotojas, sąrašas; jis naudojamas ne kontroliuoti atliekamiems veiksams, o leidžia įvertinti daugiau galimų ir įdomių problemų aspektų [27]. Tokiu būdu gali būti aptikta daug įvairaus tipo defektų, įskaitant ir panaudojamumo problemas, tačiau neretai gali būti sunku aptikti globalias sistemos problemas ir trūkstamus sąsajos elementus. Vadovaujantis euristikomis ne visada lengva spręsti iškilusias problemas, todėl problemų sprendimų paieška neretai tampa dideliu kūrybiniu iššūkiu testavimo, analitikų ar programuotojų komandai.

1.10. Informacinių sistemų testavimo būdų palyginimas

Palyginus saityno informacinės sistemos VS testuoti tinkamus testavimo metodus, būtina atsižvelgti į tai, kad daugelis metodų labiau pritaikyti testuoti funkcionalumui ir labai maža dalimi apima VS testus, o tie, kurie pritaikyti VS testuoti, dažniausiai testavimą siūlo atlikti skirtingais aspektais.

Egzistuojančių VS testavimo sprendimus galima suskirstyti į dvi pagrindines grupes:

- rankiniai testai;
- automatiniai testai.

Šių testų panaudojamumas ir efektyvumas priklauso nuo to kokių tikslu siekiama juos naudojant, kiek pastangų įdedama juos sudarant ir tobulinant, bei kokiame PĮ kūrimo etape (kūrimo ar testavimo) jie naudojami.

Palyginimui pasirinkti keturi testavimo metodai: du rankiniai ir du automatiniai (lentelė 4). Rankinis, atsitiktinės įvesties, įrašo atkūrimo testavimo metodas ir išbaigtų sąveikų sekų testavimo metodai yra daugiausiai pritaikyti VS testuoti ir tinkami saityno informacinėms sistemoms testuoti. Aptarus pagrindinius šių metodų principus, galima juos palyginti pagal šešis aspektus: testavimo trukmę, testų pakartotinio panaudojamumo galimybę, testavimo atvejų padengiamumą, papildomo pasiruošimo reikalavimą, identifikuoti testavimo metodo principą (rankinis ar automatinis) ir įvertinti metodo panaudojimo testavimui sąnaudas.

Lentelė 4. Informacinių sistemų testavimo metodų palyginimas

Lyginimo kriterijai	Metodas			
	RANKINIS		AUTOMATINIS	
	Rankinis (su priemonėmis)	Atsitiktinės įvesties	Įrašymo atkūrimo	Išbaigtų sąveikų sekų (CIS)
Testavimo greitis	lėtas	lėtas	greitas	greitas
Pakartotinas panaudojamumas	ne	ne	ne	taip
Testavimo atvejų padengiamumas	didelis	didelis	nedidelis	didelis
Reikalauja didelio papildomo pasiruošimo	ne	ne	taip	taip
Testavimo kaštai	maži	dideli	didelis	dideli

Šaltinis: sudaryta autorės.

Dauguma egzistuojančių testavimo metodų yra nukreipti į atskirus testavimo etapus, bet ne į visą procesą. Dažniausiai automatiniai testai naudojami programinės įrangos kūrimo etape. Tokie metodai kaip „įrašymo atkūrimo“ arba CIS, dažniausiai ir yra naudojami programuotojų, produkto kūrimo procese, nors neretai „įrašymo atkūrimo“ metodas testuotojų komandos yra derinamas su rankiniais testais. Automatiniai testai padeda aptikti ir išvengti kritinių klaidų, kurios pasitaiko kūrimo etape, todėl reguliariai juos atliekant defektai aptinkami ankščiau. Įvertinant tai, kad pagrindinės ir kritiškiausios VS klaidos randamos testuojant rankiniu būdu, geriausia testavimo etape rinktis rankinius testus, kombinuojant su automatiniais, kurie nereikalauja didelio įdirbio.

Automatiniai testavimo metodai, visada yra greitesni už rankinius ir gali būti panaudojami keletą kartų. Testavimo automatizavimas leidžia ženkliai padidinti testų skaičių. Tuo pačiu padidėjęs testų skaičius nereikalauja didesnių testavimo išteklių. Siekiant, kad automatinė testų panaudojamumas būtų maksimalus, reikalingas papildomas pasiruošimas. Papildomas pasiruošimas ir žinios reikalingos ne tik pirmą kartą sukūriant testus, bet ir nuolat juos atnaujinant ir tobulinant.

Automatinių testų kūrimui būtinos išskirtinės programavimo ar modeliavimo žinos, todėl šių metodų taikymas testavimo etape, neretai pareikalauja papildomų išteklių ir sukelia diskomfortą testavimo komandai.

Rankinių testavimo metodų taikymas VS testuoti nors yra lėtas ir dažniausiai nesuteikiantis pakartotino panaudojamumo galimybės, tačiau testavimo atvejų padengiamumas yra didesnis, nei taikant automatinius testus, todėl ir pagrindinės ir kritiškiausios VS klaidos randamos testuojant rankiniu būdu.

Pasirenkant testavimo metodą VS testuoti, būtina atsižvelgti į tai, kokio rezultato norima pasiekti, kiek mums yra svarbūs vieni ar kiti testavimo aspektai, bei kiek metodas yra priimtinas ir tinkamas konkrečiam testavimo atvejui ar procesui.

1.11. Vartotojo sąsajai testuoti ir vertinti taikomų metodų palyginimas

Atsižvelgiant į tai, kad defektų aptikimas ir jų pataisymas ankstyvose kūrimo stadijose padeda sumažinti sistemos kūrimo išlaidas, svarbu išskirti ir pasirinkti tokius pagrindinius sąsajos testavimo ir įvertinimo metodus, taikytinus nuo ankstyvos testavimo stadijos [25]:

- Pažintinė peržvalga;
- Atitikties rekomendacijoms ir standartams vertinimas (formalus);
- Euristinis vertinimas.

Šie metodai nereikalauja pašalinių vertintojų dalyvavimo ir gali būti atlikti pačių kūrėjų (ar kuriančios kompanijos testuotojų) rankiniu būdu. Tai lemia ir santykinai neaukštą tokio testavimo kainą.

Apibendrinti testavimo metodų ypatumų analizės rezultatai pateikiami 5 lentelėje.

Lentelė 5. Testavimo metodų ypatumai

	Pažintinė peržvalga	Atitikties rekomendacijoms ir standartams vertinimas	Euristinis vertinimas
Testavimo atvejai	Užduotys, kurias reikia atlikti	Reikalavimų pagrindu sukurtos rekomendacijos arba taisyklės	Euristikos pagal žinomus panaudojamumo principus
Neaptinkamos klaidos	Bendros ir dažnos klaidos	Rimtos koncepcinės, navigacijos ir operacinės klaidos	Globalios problemos ir trūkstami sąsajos elementai
Aptinkamos	Išskirtos problemos gali būti ne esminės	Bendrinės ir pasikartojančios objektų išdėstymo ir dizaino klaidos	Daug įvairių, įskaitant ir panaudojamumo klaidas
Problemų sprendimai	Nėra (sprendimus priima projektuotojai)	Yra (vadovaujamosi priimtomis rekomendacijomis ir standartais)	Ne visada apibrėžtas būdas problemų sprendimams gauti

Šaltinis: sudaryta autorės.

Remiantis atlikta analize galima daryti prielaidą, jog egzistuojantys metodai netenkina visų saityno informacinės sistemos grafinės vartotojo sąsajos testavimo procesui keliamų reikalavimų. Idealiu atveju testavimo metodas turėtų ne tik leisti aptikti esminius saityno informacinės sistemos grafinės vartotojo sąsajos defektus, bet ir padėti juos efektyviai šalinti. Tokį sprendimą galima tikėtis sukurti sujungus ir pagal poreikį praplėtus egzistuojančius metodus.

1.12. Informacinių sistemų testavime naudojamų įrankių palyginimas

Testavimo įrankiai, naudojami įvertinti sistemos panaudojamumui skirtingais būdais, pagal taikymo principus gali būti suskirstyti į keturias grupes [25]:

- stebėjimo;
- palyginimo;
- klausymo;
- įvertinimo.

Esamų testavimo procese naudojamų programinių įrankių analizėje yra lyginamos programos, įrankiai, kurie iš įvairių aspektų (testavimo atveju, testų kūrimo, jų vykdymo, statuso stebėjimo, klaidų ir sprendimų registravimo, kaupimo, jų statuso sekimo ir vykdymo) padėtų vykdyti VS įvertinimą ir testavimą.

Testuff yra įrankis skirtas rankinio testavimo valdymui ir vykdymui, bei defektų valdymui. Įrankio kūrėjas *Testuff Ltd*. Paketas laidžia kurti testus, testavimo scenarijus, registruoti reikalavimus, valdyti testavimo komandą, registruoti defektus ir juos priskirti atsakingiems asmenims, bei pateikia testų, jų testavimo atveju, defektų statistiką. Sprendimas pasirinktas lyginti, kaip geras ryšių tarp testavimo procese egzistuojančių testavimo reikalavimų, testavimo atveju ir defektų įgyvendinimo pavyzdys.

Tosca yra programinės įrangos įrankis labiau skirtas automatizuotų funkcinių ir regresinių testų vykdymui. Įrankis apima ir integracinių testų vykdymą, VS, komandų eilutės ir taikomųjų aplikacijų sąsajų testavimą. Įrankio kūrėjas *TRICENTIS Technology & Consulting GmbH*. Pasirinktas palyginimui, kadangi yra plačiai taikomas įvairių testų formų vykdyme įskaitant ir rankinį testavimą.

EggPlant yra „juodosios dėžės“ principų automatinį VS testavimą atliekantis įrankis. Įrankio kūrėjas *TestPlant*. Įrankis naudoja paveikslėlių atitikimo technologiją kai paieška atliekama testuojamos aplikacijos objektų lygyje. Tokiai technologijai neturi įtakos kokia kalba sukurtas testuojamas produktas ir kokioje aplinkoje jis yra testuojamas. Pasirinktas palyginimui, kadangi yra puikus regresinio VS testavimo pavyzdys rankiniu būdu kuriamiems testavimo atvejams.

TestComplete įrankis gali būti taikomas funkciniam VS testavimui, vieneto testavimui, regresiniam testavimui, "baltos dėžės" testam, duomenimis grindžiamą testavimą, padalintą (angl. *distributed*), praplėstam (angl. *coverage*) ir rankiniam testavimui. Įrankio kūrėjas *SmartBear*. Visiškai pritaikytas WEB sąsajos (angl. *front-end*) testavimui tikrinant jo funkcionalumą ir patikimumą. Kadangi sprendimas plačiai taikomas WEB sąsajai testuoti, būtina jį nuodugniai išanalizuoti.

JIRA yra projektų sekimo įrankis tinkantis komandoms planuoti, kurti ir leisti kokybiškus produktus. Įrankio kūrėjas Šis įrankis turi savyje interneto sąsaja grindžiamą defektų kaupimo, valdymo ir registravimo galimybę. Be to sprendimas yra geras pavyzdys kaip tinkamai užfiksuoti pasiūlymus, sprendimus kilusius komandoje, suteikiantis galimybę dirbti prie konkrečių veiklos elementų. Sprendimas yra gerai taikomas defektų ir sprendimų valdyme ir turi integravimo su kitais įrankiais galimybes.

Bugzilla yra WEB sąsaja skirta klaidų valdymui įrankis. Įrankio kūrėjas *Mozilla Foundation*. Įrankis leidžia ne tik registruoti klaidas, bet ir nustatyti jų prioritetus, sekti pasikeitimus gaunant el. laiškus, nustatyti prieigos teises ir valdyti projektų, komponentų ir registruotų defektų matomumą. Dėl savo unikalių savybių sprendimas yra nepakeičiamas defektų dubliavimo išvengime, nes registruojant naują defektą, pagal jo aprašymą duomenų bazėje ieškoma ar toks defektas jau nėra registruotas.

Pasirinktų esamų programų paketų palyginimas pagal kriterijus, kurie yra svarbūs VS testavime, yra pateiktas 6 lentelėje.

Lentelė 6. Testavimo procese naudojamų testavimo įrankių palyginimas

Lyginimo kriterijai	Testuff	Tosca	eggPlant	Test Complete		JIRA	Bugzilla
Testų pakartotinis panaudojamumas	ne	ne	ne	ne		ne	ne
Testavimo atvejų valdymas	taip	taip	taip	taip		ne	ne
Testavimo atvejų DB	ne	ne	ne	ne		ne	ne
Testavimo atvejų ryšiai su komponentais	ne	taip	taip	ne		ne	ne
Rankinis testavimas	taip	taip	taip	taip		ne	ne
Defektų valdymas	taip	taip	ne	ne		taip	taip
Defektų ryšiai su komponentais	ne	ne	ne	ne		taip	taip
Sprendimų valdymas	ne	ne	ne	ne		ne	ne

Šaltinis: sudaryta autorės.

Palyginus visus pasirinktus paketus, galima daryti prielaidą, kad šiuo metu nėra rankiniu testavimu būdu grindžiamo sprendimo kuriame būtų realizuotas pilnas VS testuoti reikalingas funkcionalumas. Taip pat kuris būtų pritaikytas saityno IS VS testavimui pritaikytą metodiką. Nuspręsta kurti savo sistemą, kuri būtų grindžiama sukurta saityno IS VS testavimo metodika.

1.13. Siekiamas sprendimas

VS testuoti reikalinga unikali metodika, tačiau šios metodikos įgyvendinimas turi būti grindžiamas tradicinės PĮ funkcinio testavimo eiga [10]. Specializuota VS testavimo metodika turėtų leisti specifikuoti testavimo eigą ir pasiekti išsamius rezultatus, operatyviai aptinkant saityno informacinėje sistemoje egzistuojančius VS trūkumus ir klaidas. Galiausiai metodika turėtų įtraukti priemones ir metodus, kurie yra pakankamai platūs įvairiomis VS testuoti. Be abejo, reikėtų pritaikyti ir tam tikrus egzistuojančių PĮ VS testavimo metodikų aspektus.

Metodika turėtų apjungti jau egzistuojančius metodus ir turėti savyje sprendimus pašalinančius žinomus apjungtų metodų trūkumus. Pagrindu kuriamoje metodikoje gali būti pasirinktas rankinis testavimo būdas, kuris papildomas pagrindinėmis testavimo taisyklėmis ir grindžiamas dažniausiai pasitaikančių klaidų sąrašu.

Darbe numatoma specifikuoti ir sudaryti būtent saityno (tikslinis sprendimo poaibis – savitarnos svetainės) VS testuoti pritaikytą metodiką, todėl didelis dėmesys bus skiriamas saityno informacinių sistemų VS klaidoms identifikuoti.

1.14. Analizės išvados

1. Testavimo procesą vykdančių kokybės inžinierių kvalifikacija būna keliami funkcinio testavimo srityje ir didžiąją dalį įgytos patirties taip pat sudaro programinės įrangos funkcijų testavimo patirtis. Taigi, sukurta saityno informacinės sistemos VS testavimo metodika turi būti grindžiama pagrindinėmis VS testavimo taisyklėmis ir pagrindiniais testavimo aspektais.

2. Dažniausiai pasitaikančios VS klaidos daro didelę neigiamą įtaką vartotojo darbo kokybei, sistemos panaudojamumo charakteristikoms, bei sistemos efektyvumui. Todėl į jas turi būti atsižvelgiama kuriant saityno informacinės sistemos VS testavimo metodiką, kurios taikymas padėtų aptikti dažniausiai pasitaikančių VS klaidas.

3. Ne visos VS klaidos, pasitaikančios PĮ, yra būdingos ir saityno informacinėms sistemoms. Todėl reikalinga tokia metodika, kuri padėtų išskirti VS klaidas, pasitaikančias saityno informacinėse sistemose, o tradicinių PĮ funkcinio testavimo metodikų panaudojamas testuojant saityno informacinės sistemos VS, padės ieškant kitokio pobūdžio VS klaidų, kurios pasitaiko ir PĮ.

4. Panaudojamumo testavimo tikslui pasiekti dažniausiai yra testuojama VS, o funkcionalumo testavimas dažnai padengia VS ir panaudojamumo charakteristikas. Vartotojo sąsają ženkliai veikia panaudojamumo charakteristikos, todėl tinkamai sutvarkytas panaudojamumas daro didelę teigiamą įtaką vartotojo sąsajai.

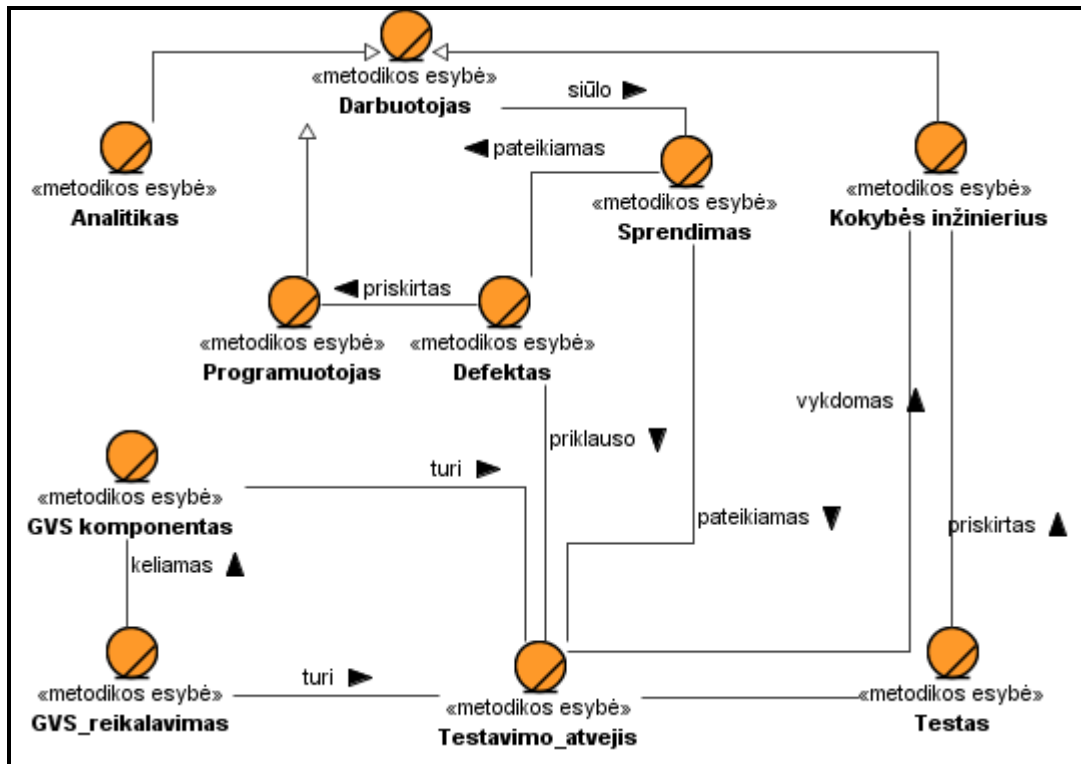
5. Tipiniai funkcinio testavimo etapai ir procesai tik iš dalies gali būti pritaikyti VS testuoti, todėl reikalinga specializuoti kai kuriuos funkciniam testavimui pritaikytus etapus, modifikuojant jų funkcijas ir įtraukiant VS testavime egzistuojančius objektus. Taip specializuoti etapai ir testavimo procesas padės užtikrinti, kad VS testavimas vyktų efektyviau ir teiktų kokybiškesnius rezultatus.

6. Žinomi VS testavimo metodai yra taikomi skirtingiems produkto kūrimo testavimo etapams, todėl juos pasirenkant svarbu nuspręsti, kokio rezultato norima pasiekti ir kokie yra metodo taikymo tikslai, bei rizikos vienu ar kitu atveju. VS testavimo metodika turėtų vadovautis rankinio testavimo principais, kadangi automatinis VS testavimas negarantuoja pakankamai aukšto testavimo atvejų padengiamumo.

2 SIŪLOMOS SAITYNO IS VS TESTAVIMO METODIKOS SUDARYMAS

2.1 Metodikos konceptai

Sukurtas teorinis (konceptualus) testavimo metodikos vaizdas pateiktas 5 paveikslėlyje. Konceptų modelyje pateikti visi pagrindiniai VS testavimo procese dalyvaujantys konceptai (su atitinkamais atributais, kurie jį apibūdina ir identifikuoja).



Šaltinis: sudaryta autorės.

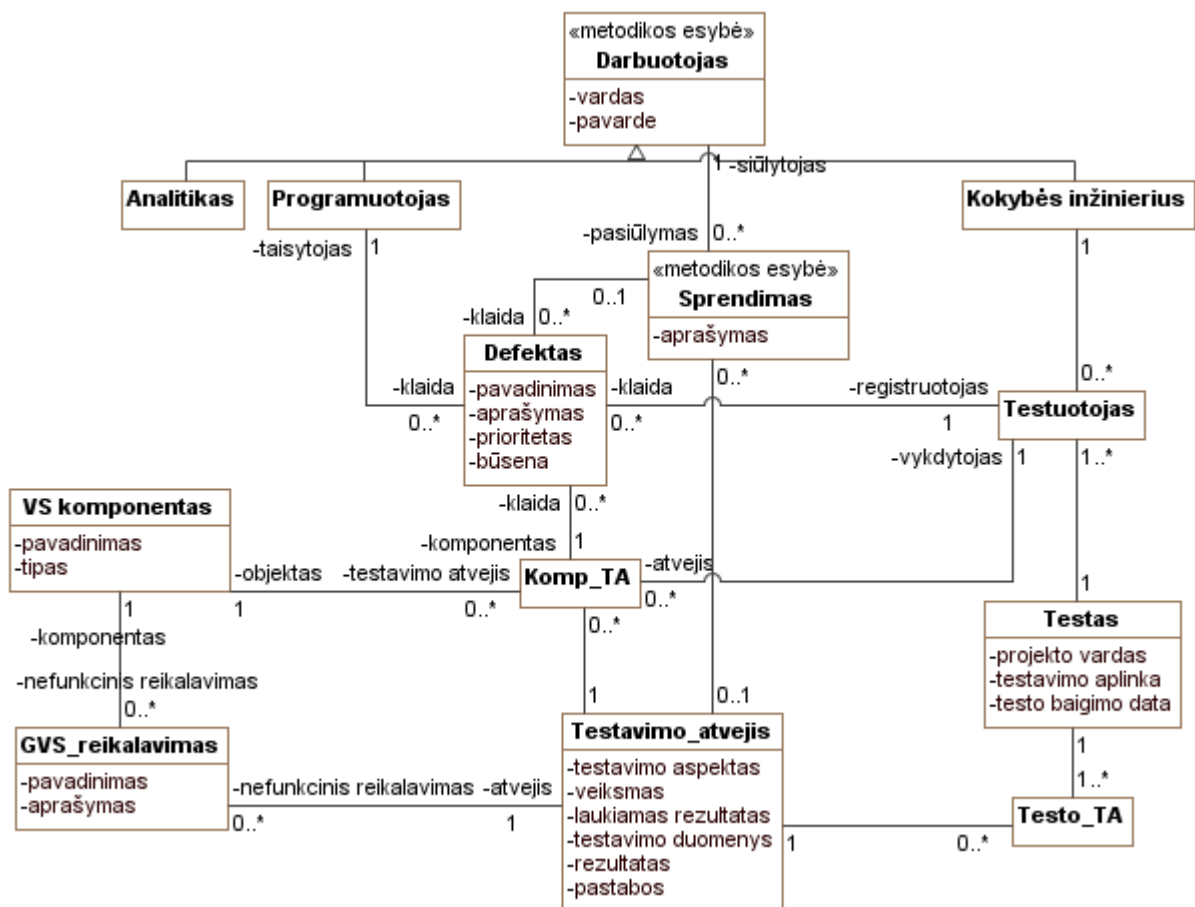
5 pav. Konceptualus testavimo metodikos modelis

Modelyje nurodyti esminiai testavimo metodikos konceptai tarpusavyje sujungti binariniais ryšiais. Iš testavimo procese dalyvaujančių veikėjų analizės išskirti tokie konceptai kaip programuotojas, analitikas ir kokybės inžinierius. Šių konceptų ryšys su kitais konceptais identifikuotas pagal testavimo procese dalyvaujančių veikėjų atsakomybių sritis, užduotis, kurias jie atlieka. Pagrindinis metodikos vykdytojas yra kokybės inžinierius, kuris gauna visą procesui reikalingą informaciją, kurios nebereikia papildomai analizuoti.

Konceptualaus modelio testavimo metodikai sudarymas padeda apibrėžti visus faktorius ir ryšius tarp jų saityno informacinės sistemos VS testavimo metodikos rėmuose. Atliekant tolimesnį testavimo metodikos modeliavimą, modeliuojama šių konceptų aplinka ir sąveika su ja.

2.2 Dalykinės srities klasių modelis

VS testavimo metodikos konceptų modelis transformuotas į kompiuterizuojamos srities esybių modelį (6 pav.). Šio modelio pagrindu bus formuojama duomenų bazės schema, kuris bus taikoma VS testavimo metodikos vykdyme.



Šaltinis: sudaryta autorės.

6 pav. Dalykinės srities klasių modelis

Kiekvienoje esybėje saugoma esybę apibūdinanti informacija, kuria manipuluojama VS testavimo metodikos metu. Kokioje esybėje kokia informacija saugoma pateikiama 7 lentelėje.

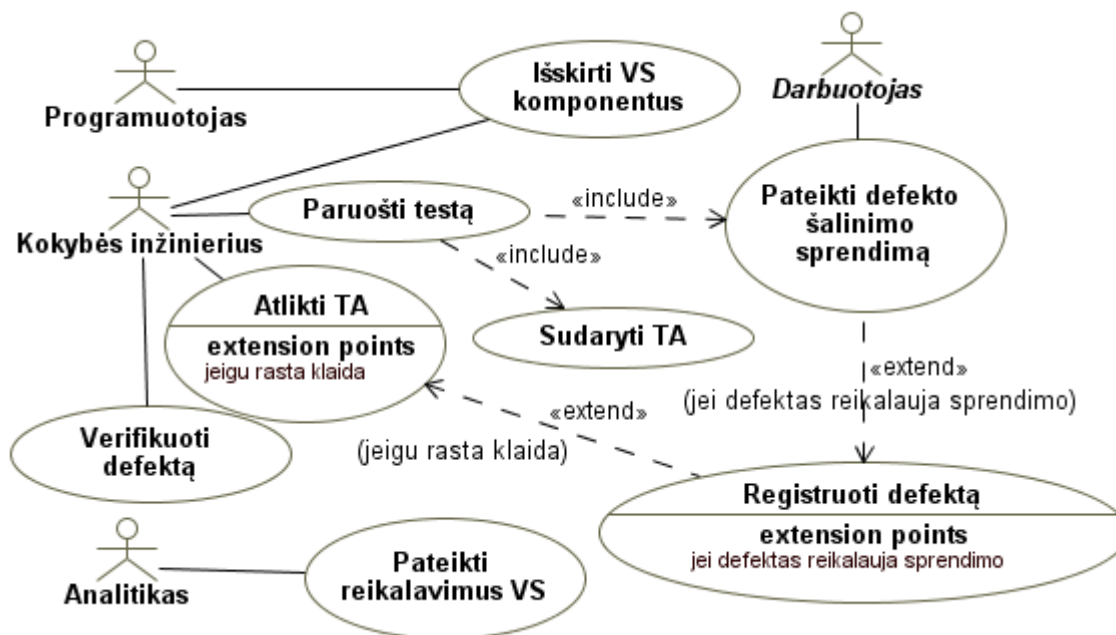
Lentelė 7. Esybių ir jose saugomos informacijos sąrašas

Nr.	Esybė	Saugoma informacija
1.	Darbuotojas	Informacija apie visus su duomenų baze galinčius dirbti darbuotojus
2.	Defektas	Informacija programuotojui apie kokybės inžinieriaus rastą VS defektą
3.	VS_komponentas	Informacija apie išskirtą saityno IS VS komponentą
4.	Testavimo_atvejis	Informacija apie tai, kokiais žingsniais turi būti testuojamas VS komponentas
5.	VS_reikalavimas	Informacija apie VS komponentui keliamus VS reikalavimus
6.	Testas	Testuojamo projekto testavimo plano ir rezultatų dokumentas
7.	Sprendimas	Informacija apie defekto sprendimo būdą arba VS patobulinimo galimybes
8.	Komp_TA	Konkreto komponento testavimo atvejis
9.	Testo_TA	Konkreto testo testavimo atvejis
10.	Testuotojas	Testą atliekantis kokybės inžinierius

Šaltinis: sudaryta autorės.

2.3 Metodikos taikymo panaudojimo atvejų modelis

Metodikos taikymas iliustruojamas panaudojimo atvejų modeliu (7 pav.). Atspindėti tipiniai metodikos vartotojai ir metodikos panaudojimas metodikos rėmuose.



Šaltinis: sudaryta autorės.

7 pav. VS testavimo metodikos taikymo veiklos

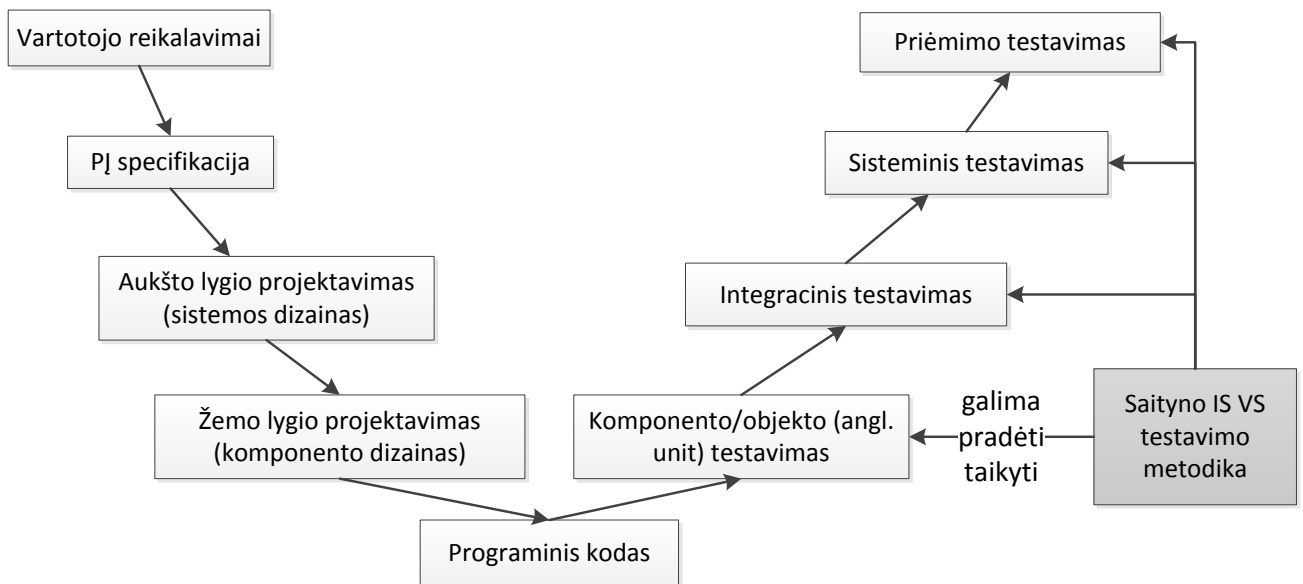
Iš testavimo procese dalyvaujančių veikėjų analizės išskirti tokie kaip programuotojas, analitikas ir kokybės inžinierius, panaudojimo atveju modelyje atitinka aktorius. Apibrėžti aktoriai yra

VS testavimo metodikos vykdytojai, jų roles atlieka testavimo procesą dalyvaujantys specialistai. Kiekvienas pagal VS testavimo metodika dirbantys vykdytojai, turi skirtingas atsakomybės sritis.

VS testavimo metodikoje išskiriama bendra funkcija „Pateikti sprendimą“, kurią gali atlikti visi testavimo procese dalyvaujantys aktoriai. Tokiam atvejui sukurtas bendras abstraktus aktorius „Darbuotojas“, kuriam ir priskirtas bendras panaudojimo atvejis.

2.4 VS testavimo metodikos etapai

Siūlomą saityno IS VS testavimo metodiką galima taikyti jau ankstyvose sistemos kūrimo stadijose. Metodika grindžiama rankinio testavimo principais, todėl priešingai nei taikant automatinį testavimo būdą būdu, testuoti galima pradėti jau tada kai vartotojo sąsaja nėra dar pakankamai stabili. Siūloma metodika testavimo etapų „v modelyje“ būtų komponento/objekto (angl. *unit*) testavimo lygyje (8 pav.).

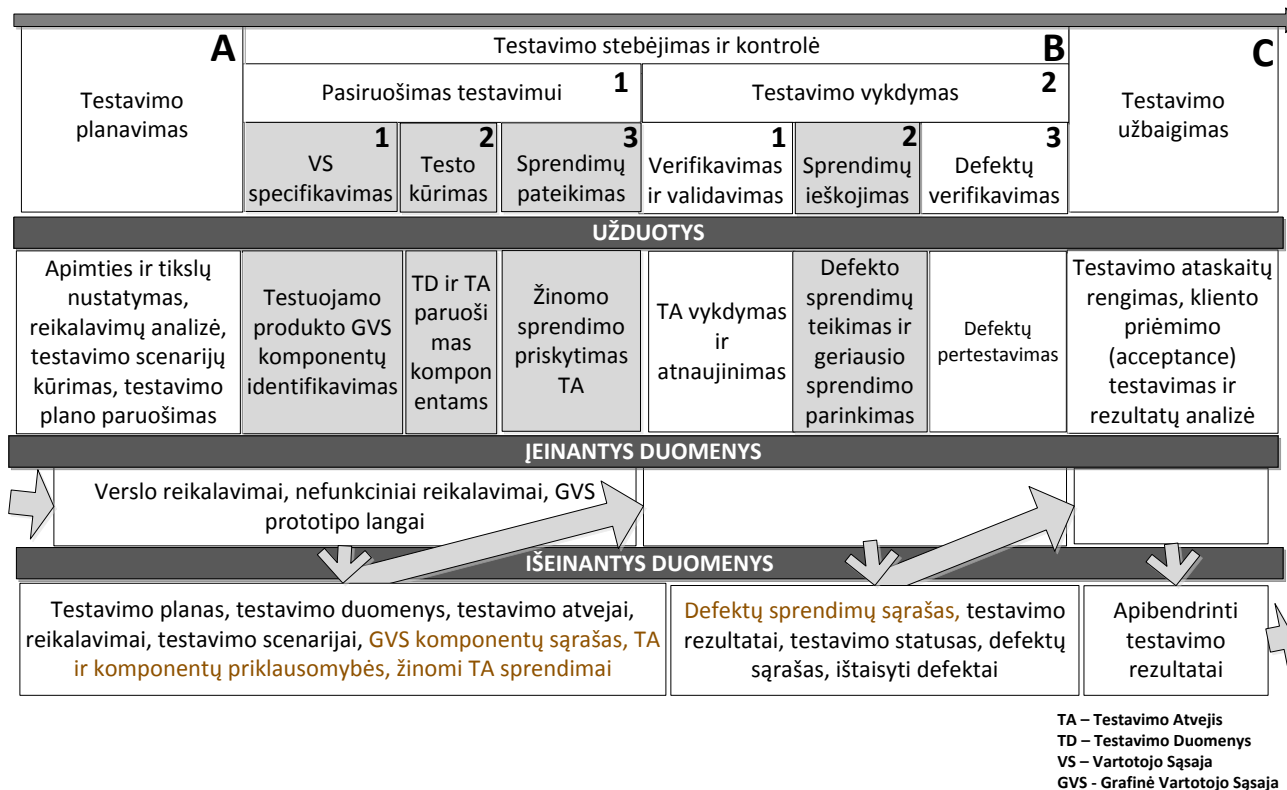


8 pav. Metodikos taikymas testavimo lygiuose

Panaudojimo atvejų diagramoje išskirti panaudojimo atvejai suskirstyti į sudarytus VS testavimo metodikos etapus. Kuriant saityno informacinių sistemų VS testavimo metodikos etapus (9 pav.) buvo vadovautasi programinės įrangos testavimo procesu, kurį apibrėžia tarptautinis programinės įrangos testavimo standartas „ISO/IEC 29199 Software testing“ [23]. Pagal šį standartą programinės įrangos testavimo procesas yra išskaidomas į procesų lygius ir jų sub-procesus.

Sudarytas VS testavimo metodikos etapų modelis, palyginti su programinės įrangos testavimo procesų modeliu, skiriasi tik dinaminių testavimo procesų lygyje esančių sub-procesų skaičiumi.

Dinaminis testavimo proceso lygis papildytas trimis etapais – vartotojo sąsajos specifikavimo (1.1), sprendimų pateikimu (1.3) ir sprendimų ieškojimo (2.2). Testavimo valdymo procesų lygis išlieka nepakitęs, toks koks yra nurodytas tarptautiniame programinės įrangos testavimo standarte. Testavimo valdymo procesų lygį sudaro testavimo planavimas (A), testavimo stebėjimas ir kontrolė (B), bei testavimo užbaigimas (C).



Šaltinis: sudaryta autorės.

9 pav. Vartotojo sąsajos testavimo metodikos etapai

Likę standartiniai PĮ testavimo procesų etapai, kurie naudojami VS testuoti, išlieka nepakitę arba skiriasi jų vykdymo principai. Vienas iš tokių modifikuotų etapų, kad būtų labiau pritaikytas VS testuoti, yra testų kūrimo (1.2) etapas. Šio ir kitų VS testavimo etapų, kuriais buvo papildytas modelis, užduotys, kurios atliekamos po VS specifikavimo etapo, glaudžiai susiejant su sprendimų pateikimo etapu (1.3).

Išskirti pagrindiniai VS testavimo etapai:

A Testavimo planavimas

B Pasiruošimas testuoti

- Vartotojo sąsajos specifikavimas.
- Testų kūrimas.
- Sprendimų pateikimas.

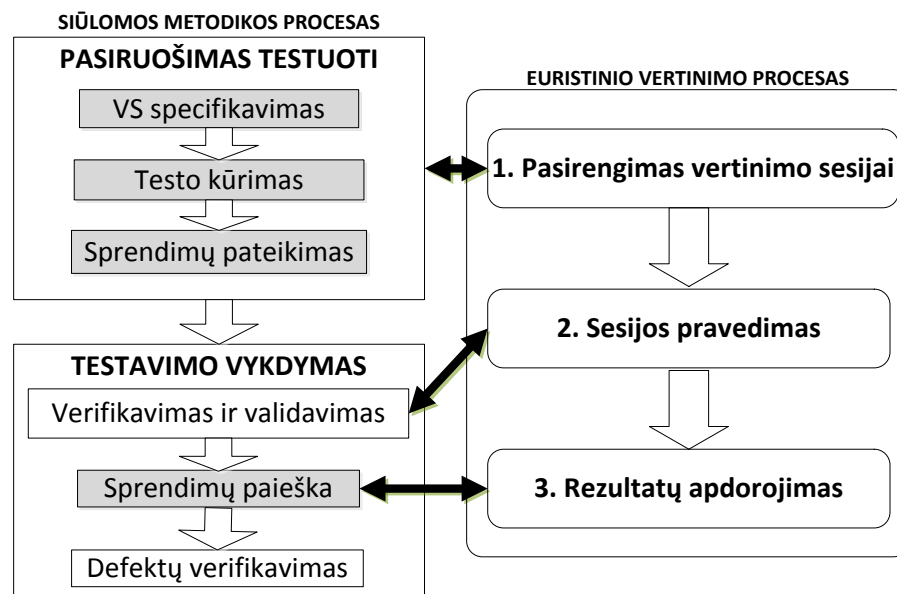
C Testavimo vykdymas

- Validavimas ir verifikavimas.
- Sprendimų paieška.
- Defektų verifikavimas.

D Testavimo užbaigimas

Kiekvieno testavimo etapo vykdymui, kad būtų atlikta etapo užduotis ir gautas proceso rezultatas, yra reikalingi įeinantys dokumentai. Pirminiai įeinantys duomenys (nurodyti modelyje) yra verslo reikalavimai, nefunkciniai reikalavimai ir grafinės vartotojo sąsajos prototipas. Po kiekvieno etapo užbaigimo yra gaunami išeinantys dokumentai, kurie gali būti panaudojami tolesniuose testavimo etapų procesuose kaip priemonės arba pagalbinių dokumentai. Testavimo planavimo ir pasiruošimo testuoti etapų išėjimo dokumentai yra naudojami testų vykdymo etape, o testų vykdymo etapo išeinanti informacija naudojama testavimo užbaigimo etape.

Didžiausias dėmesys atliekant VS testavimą turėtų būti skirtas dinaminių procesų lygio etapams – pasiruošimo testuoti ir testavimo vykdymo etapui. Šie etapai yra pritaikyti VS testuoti (10 pav.).



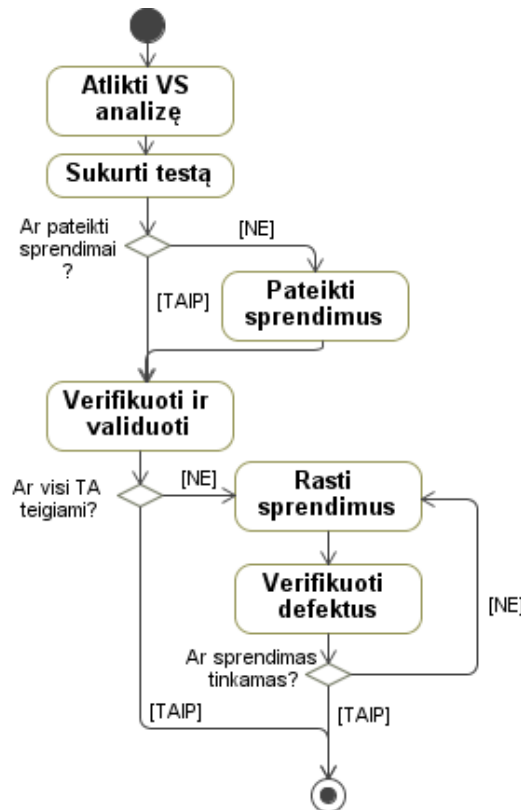
Šaltinis: sukurta autorės

10 pav. Siūlomos metodikos ir euristinio vertinimo metodo procesų sulyginimas

Testavimo vykdymo etapas sudarytame modelyje eina po pasiruošimo testuoti etapo. Šių dviejų etapų sub-etapų vykdymo tvarkai nėra griežtai apibrėžta. Kai kurie etapai gali būti persidengiantys, vykdomi kelis kartus pakartotinai pagrindinio etapo ribose arba nesant būtinybei praleidžiami. Pasiruošimo testavimo etapo sub-etapai gali būti vykdomi lygiagrečiai, tačiau tikslingiausia būtų perdengti šiuos etapus – pirmiausiai pradėti specifikuoti vartotojo sąsają, o atliekant testų kūrimo etapą,

tuo pat metu atlikti ir sprendimų pateikimą. Testavimo vykdymo etape verifikavimo ir validavimo etapas gali būti persidengęs su sprendimų ieškojimo ir defektų verifikavimo etapais, kaip ir defektų sekimo etapas gali persidengti su sprendimų ieškojimo etapu.

Naudojant veikos diagramą sumodeliuotas bendras VS testavimo metodikos procesas (11 pav.), kuris apima pasiruošimo testuoti ir testavimo vykdymo etapus.



Šaltinis: sudaryta autorės.

11 pav. Bendras VS testavimo metodikos procesas

Bendrame VS testavimo metodikos procese iliustruojami visi pagrindiniai šeši etapai. Identifikuoti trys sprendimo taškai – ar pateikti sprendimai, ar visi TA teigiami ir ar sprendimas tinkamas. Šie sprendimo taškai lemia ar visi VS testavimo metodikos etapai/procesai bus vykdomi ir kiek kartų jie bus vykdomi.

2.4.1 Testavimo planavimas

Testavimo planavimo etapas yra pagrindinis pasiruošimo testuoti etapas. Šiame etape tiksliai įvertinta testavimo apimtis (angl. *estimation*) ir testuoti reikalingi išteklių padeda nuspręsti kiek laiko

ir kiek išteklių pareikalaus visas testavimo procesas. Etapas yra vienas iš testavimo valdymo etapų labiau reikalingas darant didelius projektus, todėl tiesiogiai su testavimu nėra glaudžiai susijęs, tačiau jo metu testavimo vadovo parengiami testavimo planai atsižvelgiant į testavimo metodiką palengvina testuotojų komandos darių bendradarbiavimą.

Testavimo planas kokybės inžinieriams yra kaip gairės padedančios susikcentruoti ties testuojamo produkto testavimo specifiška. Testavimo plane numatoma, kokius uždavinius taikyti testavimo metu, kokie turėtų būti rezultatai, bei būdą, kuriuo testavimo rezultatai įvertinami ir pakartotinai naudojami regresinio testavimo metu. Pagrindinės testavimo plano sudedamosios dalys [24] gali būti tokios:

- tikslų, apimties apibrėžimas, rizikos identifikavimas;
- testavimo būdas (metodai, komanda, aplinka, įrankiai);
- grafiko sudarymas;
- testavimo pabaigos kriterijai.

Ruošiantis VS testavimui ir sudarant testavimo planą didžiausias dėmesys turėtų būti skiriamas laukiamo rezultato apibrėžimui, testavimo aplinkos skyriui ir testavimo pabaigos kriterijams. Laukiamo rezultato apibrėžimas turėtų nurodyti, kokio rezultato tikimasi pabaigus VS testavimą. Kriterijus gali būti nurodomas procentine išraiška ar skaičiumi, kuris parodytų kiek testavimo atvejų turėtų būti sėkmingai įvykdytų ar kiek VS defektų sistemoje gali likti, kad jos kokybė išliktų puiki.

Saityno informacinės sistemos VS kokybės užtikrinimui ir testavimo sėkmei didžiausia įtaką daro aplinka, kurioje bus atliktas sistemos testavimas. Šioje testavimo plano skiltyje būtina nurodyti aplinkas, kuriose reikalinga atlikti VS testavimą. Vartotojai sąsają su saityno informacinėmis sistemomis užmezga per naršyklės. VS kokybei įtakos turi tai, kad dauguma laisvų, atviro kodo ir komercinių naršyklių nevienodai palaiko interneto tinklapių standartus ir technologijas, todėl tinklapių VS yra atvaizduojama nevienodai. Pasirinkti, kurias naršyklės ir jų versijas naudoti saityno testavimui, galima pagal naršyklių naudojimo statistiką. Kai testuojama jau egzistuojančios saityno informacinės sistemos VS, atlikus besilankančių naudotojų naudojamų naršyklių analizę, nustatomos prioritėtinės naršyklės. Jeigu ruošiamasi testuoti dar neegzistuojančios saityno informacinės sistemos VS, pravartu pasinaudoti nepriklausomais tyrimais.

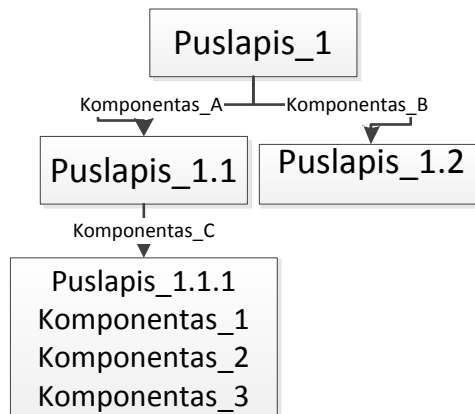
Testavimo pabaigos kriterijai nurodo, kada testavimo procesas gali būti laikomas baigtu. Čia nurodoma kiek ir kokių testų, bei jų testavimo atvejų, turi būti atlikta ir koks turi būti defektų statusas.

Kitos testavimo plano dalys yra bendrinės, todėl jos pagal savo struktūrą ir prasmę puikiai tinka ir VS testavimo plano aprašymui.

2.4.2 Vartotojo sąsajos specifikuojimo etapas

Pasiruošimą testavimui sudaro trys etapai: vartotojo sąsajos specifikuojimo, testo kūrimo ir sprendimų pateikimo etapai. VS testavimo metodikai pritaikytas etapas, kurio metu yra identifikuojami VS komponentai ir sukuriama ryšys tarp VS komponentų ir testavimo atvejų. Saityno informacinės sistemos VS komponentų išskyrimas atliekamas vadovaujantis dizaino šablonais ar prototipais. Pirmiausia yra išskiriami saityno informacinės sistemos langai, o vėliau ir tų langų komponentai. Komponentų išskyrimas leidžia atlikti su jais susijusius testavimo atvejus, o langų išskyrimas reikalingas testuojant sąveikas tarp komponentų.

Komponentų išskyrimui gali būti naudojama VS komponentų hierarchija (12 pav.).



Šaltinis: sudaryta autorės

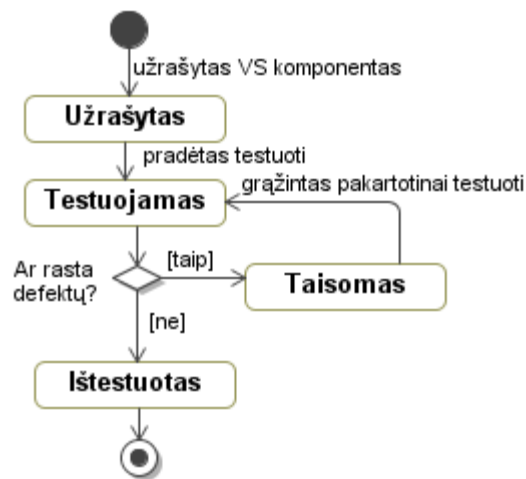
12 pav. VS komponentų hierarchija

VS komponentų hierarchija saityno informacinės sistemos sudaroma tokiu principu:

1. išskiriami saityno puslapiai;
2. nustatomi navigacijos komponentai tarp puslapių;
3. nustatomi puslapyje esantys komponentai.

Pagal sudarytą VS komponentų hierarchiją galima užtikrinti sąveikas tarp komponentų testavimą. VS testavimui svarbu ne tik kiek kodo yra ištestuota, bet ir kiek skirtingų įmanomų PĮ būsenų yra testuojama kiekvienoje kodo dalyje. Siekiant užtikrinti testavimo pilnumą, kuris bus atliekamas pagal komponentus, labai svarbu komponentus suskirstyti pagal modalinius langus, testuoti kiekvieną komponentą. Nepamesti testuojamų komponentų sekos, galiam sudarius įvykių sekos grafą.

VS komponentas, kaip konceptas, gali įgyti keturias skirtingas būsenas (13 pav.).



Šaltinis: sukurta autorės.

13 pav. VS komponento būsenos kitimas

Sėkmingo scenarijaus metu galutinė VS komponento būseną yra ištestuotas, nesėkmingo – taisomas.

2.4.3 Testo kūrimo etapas

Šio etapo metu paruošiami testavimo duomenys ir testavimo atvejai, kurie sudaro testuojamos saityno informacinės sistemos VS testą. Tinkamai parašyti testavimo atvejai turi išliekamąją vertę tolimesniems sistemų testavimams. VS testavimui reikalingi testavimo duomenys gali būti tokių kategorijų:

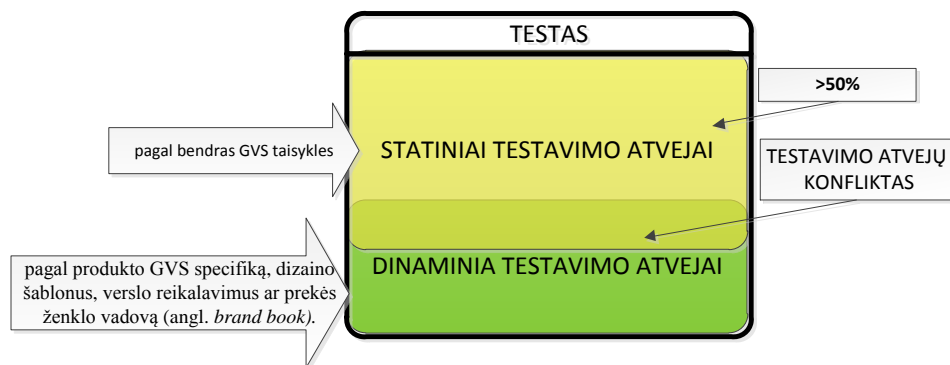
1. prisijungimo duomenys, skirtingas teises ar funkcijas matančių saityno informacinės sistemos naudotojų.
2. specifiniai verslo duomenys reikalingi įvedimo formoms, kurie bus reikalingi sėkmingai veiksmų seką.

VS testavimo atvejai sukuriama pagal VS reikalavimus, iš ankstesnių projektų sukauptų duomenų arba remiantis kokybės inžinieriaus patirtimi. Testavimo atvejai yra kaip testavimo rekomendacijos, kurios yra kaip nurodymas, o ne griežta taisyklė. Metodikos testavimo atvejai yra dviejų skirtingų tipų (14 pav.):

1. statiniai testavimo atvejai;
2. dinaminiai testavimo atvejai.

Statiniai testavimo atvejai, tai tokie testavimo atvejai, kurie yra bendri visų saityno VS testavimui. Tokie testavimo atvejai yra kas kart pakartotinai panaudojami testų kūrimui, todėl būtina turėti tokių testavimo atvejų sąrašą. Testavimo atvejai kaupiami pagal ankstesnių projektų testavimui

sudarytus komponentus ir jų testavimo atvejus. Šie testavimo atvejai turėtų būti VS testavimo metodikos pagrindas. Šio tipo testavimo atvejų teste turėtų būti ne mažiau kaip 50%.



Šaltinis: sukurta autorės.

14 pav. VS testo sudėtis

Dinaminiai testavimo atvejai yra tokie testavimo atvejai, kurie kaskart yra kuriami nauji atsižvelgiant į testuojamo produkto VS specifiką, dizainą ar prekės ženklo vadovą (angl. *brand book*). Tokio tipo testavimo atvejai yra grindžiami verslo reikalavimų arba dizaino šablonais, todėl kartais gali būti prieštaraujantys statinio tipo testavimo atvejams. Kilus konfliktui tarp skirtingo tipo testavimo atvejų jis yra sprendžiamas nelaukiant testavimo atvejų vykdymo etapo. Konfliktą būtina išspręsti jau sudarinėjant testą, todėl, kad vėliau atliekant testavimo atvejus, VS kokybę nulems tas testavimo atvejis, kuris bus atliktas paskutinis, tačiau jis ne visada gali būti tinkamiausias.

Objektas testas viso testavimo proceso metu įgyti gali tris būsenas (15 pav.).



Šaltinis: sukurta autorės.

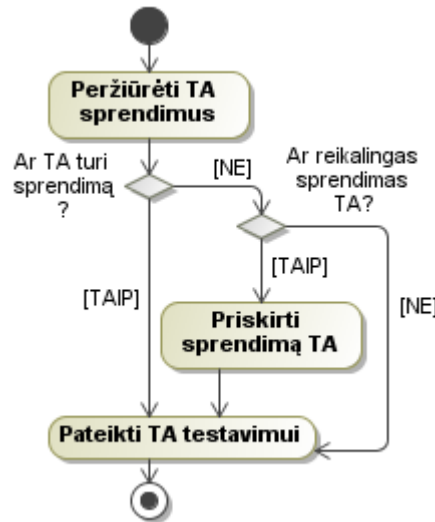
15 pav. Testo būsenos kitimas testavimo proceso metu

Sudarytas testas yra tada kai kokybės inžinierius jį sudaro testo kūrimo etape atrinkdamas testavimo atvejus. Vykdomas testas susijęs su vykdomais testavimo atvejais testo vykdymo etape iki kol tampa atliktu testu. Atliktas testas yra susijęs su jo testavimo atvejais. Testas atliktas sėkmingai kai

visi testavimo atvejai yra būsenoje „teigiamas“, nesėkmingai atliktas testas turi būseną tada kai bent vienas testavimo atvejis yra būsenoje „neigiamas“.

2.4.4 Sprendimų pateikimo etapas

Kiekvienam testo sudarymo etape sudarytam VS testavimo atvejui kokybės inžinierius gali priskirti iš anksto žinomą VS testavimo atvejo sprendimą (16 pav.), kuris turėtų būti taikomas radus defektą VS komponente.



Šaltinis: sukurta autorės.

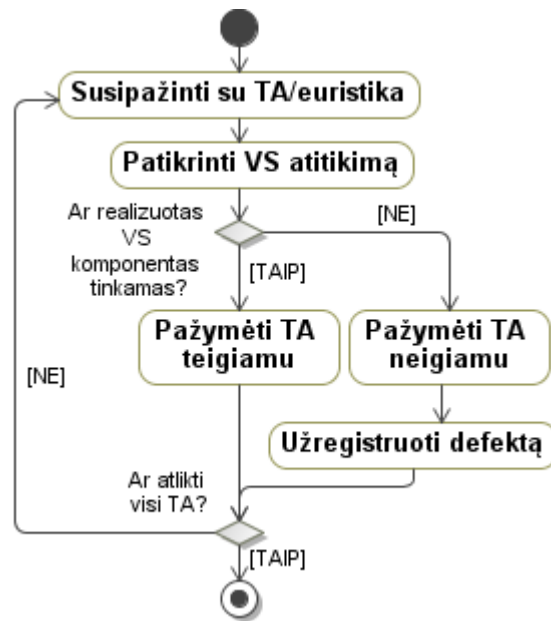
16 pav. Sprendimo pateikimo TA veiklos procesas

Sprendimai taip pat gali būti gaunami pakartotinai panaudojant jau turimus testavimo atvejus iš ankščiau atliktų testų. Sprendimai turėtų būti tik rekomendacinio pobūdžio, kad esant būtinybei būtų galima priimti sprendimą defektui taisyti atsižvelgiant į situaciją testuojamame produkte.

Po pasiruošimo testavimui, testavimo vykdymą sudaro trys etapai – validavimas ir verifikavimas, sprendimų ieškojimas ir defektų verifikavimas.

2.4.5 Validavimo ir verifikavimo etapas

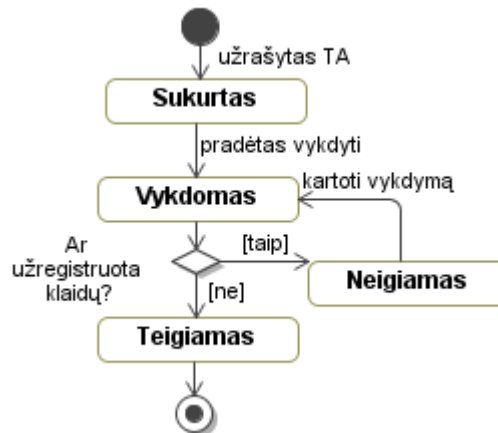
Vartotojo sąsajos verifikavimą, validavimą ir visas kitas užduotis šiame metodikos etape vykdo kokybės inžinierių komanda. Validavimo ir verifikavimo etape siekiama patikrinti sistemos vartotojo sąsają atliekant dvi veiklas – įsitikinti ar vartotojo sąsajos komponentai teisingai veikia ir ar ji atitinka reikalavimus. Validavimo ir verifikavimo procesas susideda iš šešių pagrindinių veiklų. Validavimo ir verifikavimo etape vykstantys procesai iliustruojami veiklos diagramoje (17 pav.).



Šaltinis: sukurta autorės.

17 pav. Validavimo ir verifikavimo etapo procesas

Validavimo ir verifikavimo etapo procesas atliekamas tol kol yra neatliktų testavimo atveju/euristikų. Po šio etapo visi teste esantys testavimo atvejai turi įgyti kitas būsenas. Konceptas testavimo atvejis turi keturias būsenas (18 pav.). Sėkmingo scenarijaus metu testavimo atvejis įgyja būseną teigiamas, o nesėkmingo – neigiamas.



Šaltinis: sukurta autorės.

18 pav. Testavimo atvejo būsenos kitimas

Radus neatitikimus tarp realizuoto VS komponento ir euristikos, testavimo atvejis pažymimas kaip neigiamas ir yra atliekamas defekto registravimas. Užregistruotas defektas yra priskiriamas programuotojui, kad būtų atliktas jo pataisymas. VS komponento programuotojas turėtų būti ir jo defektų taisytojas, todėl geriausia, kad užregistravus defektą būtų nurodomas atitinkamas

programuotojas arba defektas registruojamas prie atitinkamo komponento už kurį būtų atsakingas jį kūręs programuotojas.

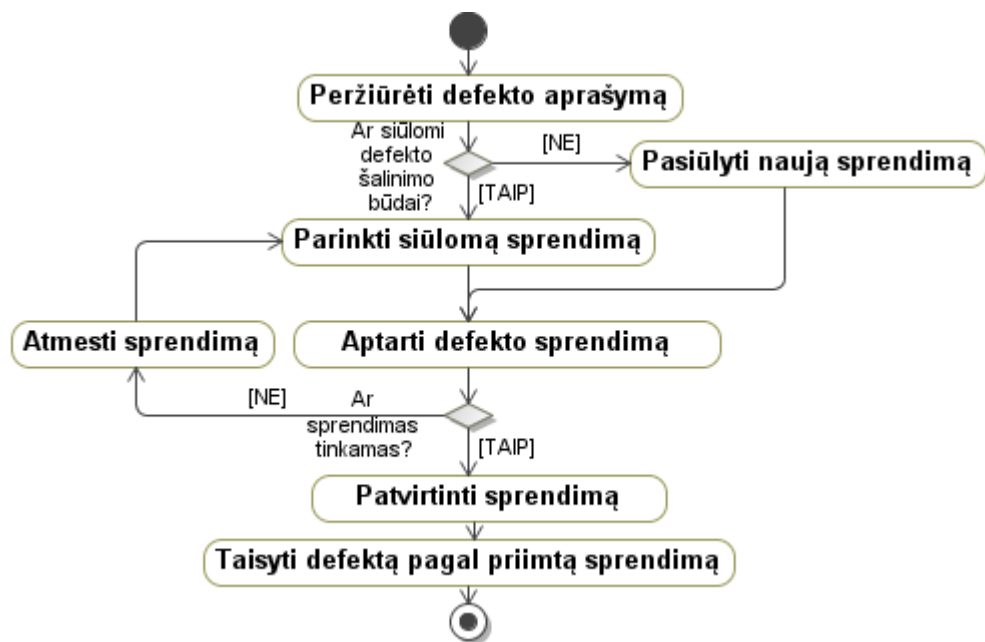
Siekiant užtikrinti atsekamumą tarp atlikto VS komponento testavimo atvejo ir užregistruoto defekto turi būti sukurtas ryšys. Tai būtina užtikrinti tam, kad sekantį kartą testuotojas atsidaręs testą galėtų matyti atlikimo statusą arba atliekant defektų pakartotinį testavimą, būtų aišku, kurio testavimo atvejo vykdymo metu buvo rastas defektas ir koks jis. Prie testavimo atvejo pateikiamas sprendimas padeda ne tik identifikuoti defektą, bet yra kaip defekto aprašymas, padedantis jį ištaisyti.

VS testavimo metodikos procesuose objektų būsenos yra labai svarbios. Objektų, kurie išskirti iš konceptų modelio, būsenos VS testavimo metodikoje yra svarbesnės negu sistemos procesų būsenos, nes pagal objektų būsenas gali būti apibrėžta testavimo kokybė ir pagal konkrečių objektų būsenas, kai jos tenkina tam tikrą sąlygą, nustatoma kokie veiksmai pagal metodiką turi būti atliekami toliau vykdant testavimo procesą.

2.4.6 Sprendimų paieškos etapas

Užregistravus defektą jam gali nereikti papildomo sprendimo arba jam gali būti reikalingas sprendimas. Defektas sprendimą gali turėti jeigu jis buvo priskirtas testavimo atvejui arba defektui gali reikėti priimti sprendimą pagal kurį bus atliktas defekto pataisymas.

Sudarytas sprendimo ieškojimo veiklos diagrama (19 pav.) atspindi visus sprendimų ieškojimo etape vykdomus procesus.



Šaltinis: sukurta autorės.

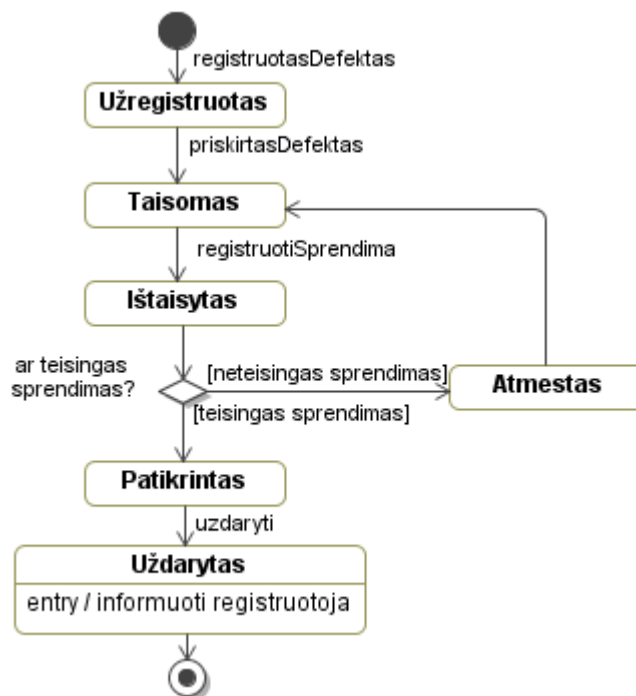
19 pav. Sprendimų paieškos etapo veiklos diagrama

Į defekto sprendimo teikimo ir priėmimo procesą gali būti įtraukti ne tik kokybės inžinierius ir programuotojas, bet ir analitikas. Analitikas reikalingas tuo atveju, kai sprendžiamas defektas glaudžiai susijęs su kliento poreikiais. Iš defektui pateiktų sprendimų turi būti išrinktas konkrečiu atveju tinkantis sprendimas, paliekant galimybę matyti ir kitus siūlytus sprendimus.

2.4.7 Defektų verifikavimo etapas

Paskutinis VS testavimo metodikos etapas, kurio metu atliekamas neigiamų testavimo atvejų pakartotinis testavimas ir ištaisytų komponentų defektų verifikavimas. Neigiami testavimo atvejai pakeičiami į teigiamus jeigu priimtas sprendimas yra tinkamas, o jeigu sprendimas nėra tinkamas defekto pataisymas yra atmetamas. Atmetus defektą pritaikomas kitas iš ankščiau siūlytų ir svarstytų defekto sprendimų variantų.

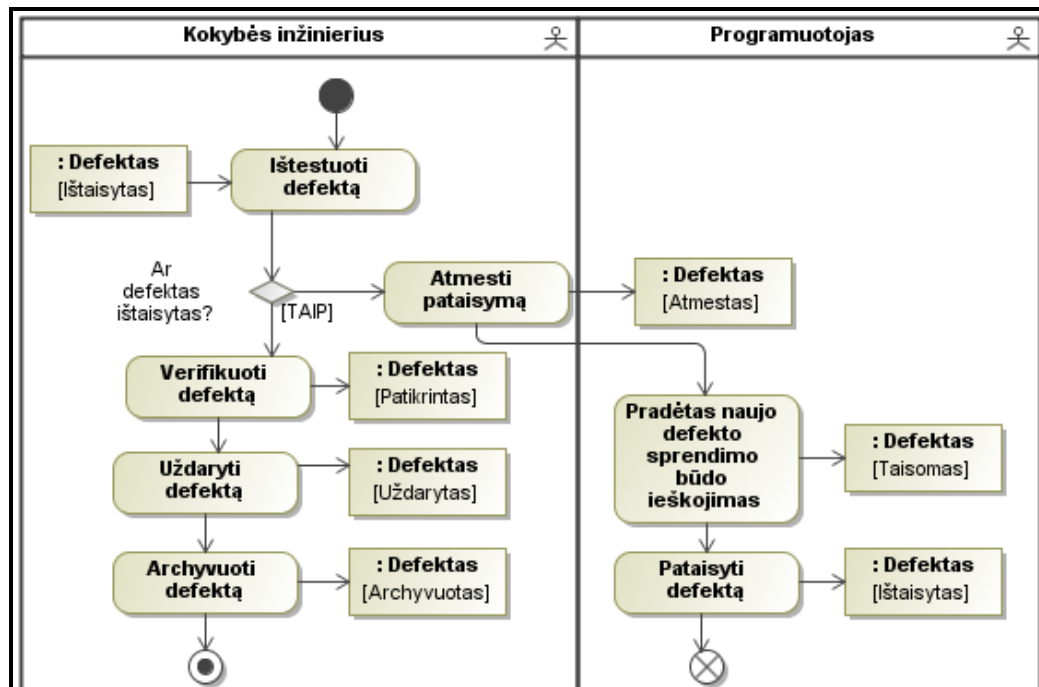
Konceptas defektas šio etapo metu gali įgyti šešias būsenas (20 pav.).



Šaltinis: sukurta autorės.

20 pav. Defekto būsenų kitimas defektų verifikavimo etapo metu

Defekto verifikavimo procese dalyvauja du specialistai – kokybės inžinierius ir programuotojas (21 pav.).



Šaltinis: sukurta autorės.

21 pav. Defekto validavimo procesas

Sėkmingo veiklos srauto atveju procese gali dalyvauti tik kokybės inžinierius, o jeigu pasukama nesėkmingo proceso srautu, tada reikalingas ir programuotojo įsikišimas į procesą. Defektų verifikavimo etapas atliekamas tol, kol visi defektai yra ištaisomi ir verifikuojami, o testavimo atvejai tampa teigiami.

2.5 Siūlomo sprendimo išvados

1. Šioje darbo dalyje buvo pateiktas formalus saityno informacinės sistemos VS testavimo metodikos aprašas ir aprašas, kuris paaiškina kaip atlikti grafinės vartotojo sąsajos testavimą vadovaujantis tokia metodika.

2. Taikant sudarytą VS testavimo metodiką būtų galima sukurti specializuotus įrankius, pritaikytus atskiriems VS testavimo metodikos etapams, kurių veiklos procesai yra žinomi ir smulkiau detalizuoti veiklos ir būsenų diagramomis.

3. Pagal VS testavimo metodiką sukurtas ir jai pritaikytas įrankis turėtų užtikrinti pakartotinį testavimo proceso metu sukurtų komponentų panaudojamumą, tai padėtų metodiką padaryti efektyviau pritaikomą testavimo procese ir sukurti prielaidas kokybiškesnio VS testavimo atlikimui.

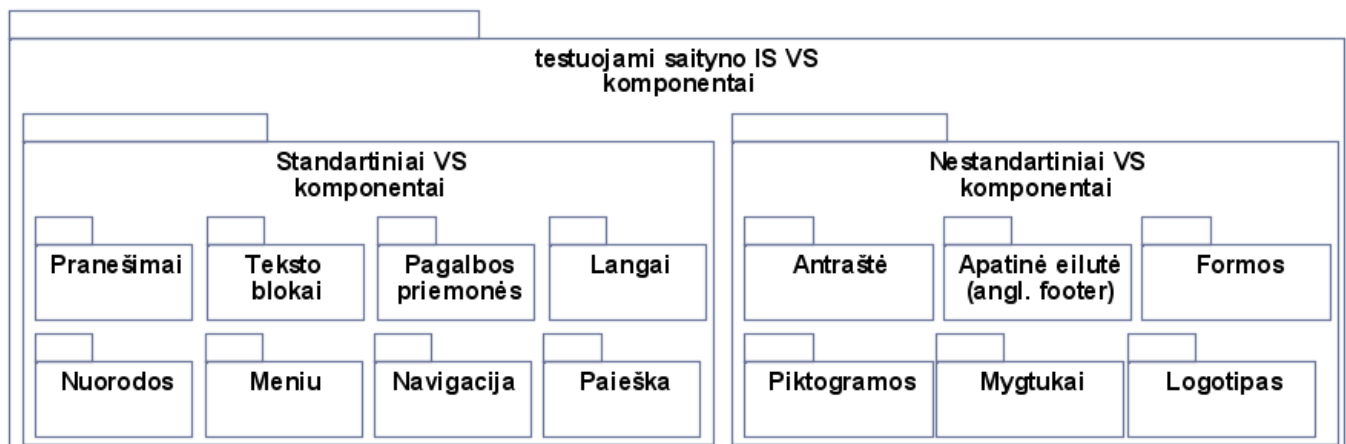
3 VARTOTOJO SĄSAJOS TESTAVIMO METODIKOS REALIZACIJA

3.1 VS komponentų aibė

Vienas iš pirmųjų ir svarbiausių pasiruošimo etapų, kuris yra numatytas sudarytoje saityno VS testavimo metodikoje, yra vartotojo sąsajos specifikavimo etapas (2.4.2 skyrius). Šio etapo metu pagal turimus dizaino šablonus ar prototipus yra sudaroma saityno VS komponentų aibė (22 pav.). Pagal sudarytą komponentų aibę yra nusprendžiama kokie testavimo atvejai bus reikalingi testuojamai saityno informacinės sistemos VS.

Saityno informacinės sistemos skiriasi savo struktūra ir turiniu, tačiau visose pasitaiko standartinių saityno informacinės sistemos VS komponentų. Šių standartinių komponentų sąrašas sudarytas atsižvelgiant į dažniausiai pasitaikančių saityno IS VS klaidų grupes, kurios apibrėžtos J. Johnson knygoje [21]. Klaidų grupės turi joms būdingą klaidų pasitaikymo vietą saityno informacinėse sistemose, kurios ir gali būti identifikuojamos kaip komponentai.

Kita stambi komponentų grupė, kuri yra būdinga kiekvienai saityno informacinės sistemos struktūrai atskirai, yra nestandartiniai VS komponentai. Siekiant sudaryti tokį specifinių komponentų sąrašą yra tikslinga pasirinkti tikslinį metodikos sprendimo poaibį. Pasirinktas tikslinis sprendimo poaibis – savitarnos svetainės. Turima savitarnos svetainių VS testavimo patirtis padeda tiksliau išskirti tipinį, šioms ir kitokio tipo svetainėms būdingą, nestandartinių VS komponentų sąrašą.



Šaltinis: sukurta autorės.

22 pav. Testuojamų saityno IS VS komponentų aibė

Rekomenduojamas standartinių ir nestandartinių komponentų rinkinys, gali būti panaudotas kaip aibė su kuria galima dirbti pagal sukurtą metodiką.

VS komponentų aibė, kuri bus naudojama VS testavime:

- langai (langų struktūra, prisijungimo langas, laukimo langai);
- nuorodos (išorinius ir vidinius puslapius);
- teksto blokai;
- mygtukai;
- meniu;
- navigacija;
- logotipas (svetainės ar kompanijos);
- viršutinė eilutė (antraštė);
- paieška;
- formos;
- apatinė eilutė (poraštė);
- piktogramos;
- pranešimų tekstai (klaidų ir informaciniai pranešimai);
- pagalbos priemonės (laukų paaiškinamieji tekstai, šabloniniai pavyzdžiai, mygtukų etiketės).

Išskirtiems VS komponentams turi būti sudaryti ir priskirti jų testavimo atvejai.

3.2 VS testavimo atvejų rinkinys

Pagal sudarytą metodiką, VS testavimo atvejai priklauso išskirtiems VS testavimo komponentams. Statiniai testavimo atvejai suformuoti kiekvienam komponentui pateikiami 8 lentelėje.

Lentelė 8. VS testavimo atvejai/euristikos

VS komponentas	TA nr.	Testavimo atvejis (TA) – euristika
langai	01	Tame pačiame lange pateikiami skirtingi funkcionalumai.
	02	Langų dizainas visoje sistemoje turi vienodą/vientisą stilių.
	03	Lango elementai nėra "sulipę", jie yra atskiriami vienodo dydžio tarpais.
	04	Po kiekvieno vartotojo atlikto veiksmo, kuris inicijuoja kažkokio rezultato pateikimą, arba

VS komponentas	TA nr.	Testavimo atvejis (TA) – euristika
		kuris inicijuoja perėjimą veiksmų seka (angl. <i>flow</i>), vartotojui aišku ką veikia sistema.
	05	Vartotojui atliekant veiksmų seka turi būti rodoma kuriame žingsnyje jis yra.
nuorodos	06	Nuorodos į išorinius puslapius neturi įtakos vartotojo likimo saityno IS.
	07	Nuoroda į vidinius puslapius atidaroma tame pačiame naršyklės lange.
	08	Nuorodoms visoje saityno informacinėje sistemoje taikomas vienodas stilius.
	09	Nuorodos turi tris būsenas – normali (angl. <i>normal</i>)/užvestas (angl. <i>hover</i>)/paspaustas (angl. <i>pressed</i>)
teksto blokai	10	Tekstas lygiuojamas pagal abi kraštines išlaikant vienodus tarpus tarp eilučių ir žodžių
	11	Teksto spalva neturi būti šviesesnė nei teksto fono.
mygtukai	12	Mygtukai turi tris būsenas – normali (angl. <i>normal</i>)/užvestas (angl. <i>hover</i>)/paspaustas (angl. <i>pressed</i>)
	13	Mygtukams visoje saityno informacinėje sistemoje taikomas vienodas stilius.
	14	Mygtukų pateikimo vieta visada yra intuityvi vartotojui.
	15	Vartotojo sąsajos elementai turintys skirtingas paskirtis skiriasi savo išvaizda.
menu	16	Vartotojas menu mato visada, nepriklausomai nuo to, kurioje saityno IS vietoje būtų.
	17	Menu turėtų būti kuo arčiau puslapio viršaus.
	18	Menu neturėtų būti gilesnis nei trijų lygių.
	19	Yra aiškiai suprantama kuris menu punktas yra pasirinktas/atidarytas.
navigacija	20	Yra aiškiai nurodyta, kurioje svetainės vietoje yra vartotojas.
kompanijos ar svetainės logotipas	21	Logotipas turėtų būti toje pačioje kiekvieno atskiro interneto svetainės puslapio vietoje.
	22	Vartotojas iš bet kurio puslapio gali grįžti į pradinį puslapį.
viršutinė eilutė	23	Vartotojas mato viršutinę eilutę visada, nepriklausomai nuo to, kurioje saityno IS vietoje jis yra.
	24	Viršutinėje eilutėje pateikiama vartotojo informacija ir jo prisijungimo būseną.
	25	Viršutinėje eilutėje pateikiamas atsijungimo mygtukas.
paieška	26	Paieškos įrankius vartotojas turi turėti galimybę pasiekti iš bet kurios saityno vietos, nepriklausomai nuo to, kurioje vietoje jis būtų.
formas	27	Formose visada aišku, kuris mygtukas kokį veiksmą atlieka.
	28	Paspaudus mygtuką, kuris inicijuoja pakeitimą, kurio vartotojas negali atšaukti ar keisti, vartotojo yra paprašoma patvirtinti veiksmą.
	29	Įvykus klaidai, vartotojui yra aišku, kurioje vietoje reikalinga atlikti pataisymą.
apatinė eilutė	30	Visada pateikiama, nepriklausoma nuo to, kurioje saityno IS vietoje yra vartotojas.
	31	Pateikiamas kontaktinė, teisinė informacija.
piktogramos	32	Paspaudžiamos piktogramos turi tris būsenas – normali (angl. <i>normal</i>)/užvestas (angl. <i>hover</i>)/paspaustas (angl. <i>pressed</i>).
	33	Piktogramoms visoje saityno informacinėje sistemoje turi būti taikomas vienodas stilius.

VS komponentas	TA nr.	Testavimo atvejis (TA) – euristika
pranešimų tekstai	34	Visada yra aišku kuris klaidos ar informacinis pranešimas kuriam funkcionalumui priklauso.
	35	Informaciniai pranešimai visada pateikiami zonoje į kurią žiūri vartotojas.
	36	Pateikiami konkretūs ir vartotojui suprantami pranešimai.
	37	Klaidų, paaiškinamiesiems ir informaciniams pranešimams taikomi skirtingos teksto spalvos.
	38	Vartotojui visada yra suprantama, kodėl jis nemato informacijos ar kodėl jo teisės sistemoje yra apribotos.
pagalbos priemonės	39	Teksto įvedimo laukas, kuriam taikoma kaukė (angl. <i>mask</i>), savyje turi turėti pavyzdinį tekstą neryškia spalva, kuris išnyksta pažymėjus įvedimo lauką.
	40	Vartotojui visada yra aišku, kodėl funkcionalumas ar jo mygtukai yra neaktyvūs.
	41	Vartotojui visada aišku, kokios informacijos tikimasi iš jo įvedimo laukuose.
	42	Vartotojas gali pasiekti pagalbą iš bet kurios sistemos vietos.

Šaltinis: sukurta autorės.

Šie universalūs, statiniai testavimo atvejai/euristikos turi būti įtraukti į duomenų bazę kaip tipiniai testavimo atvejai tinkami visoms sistemoms. Pritaikyti konkrečiam projektui, specifiniai testavimo atvejai turėtų būti įtraukiami į duomenų bazę kiekvienai testuojamai vartotojo sąsajai atskirai.

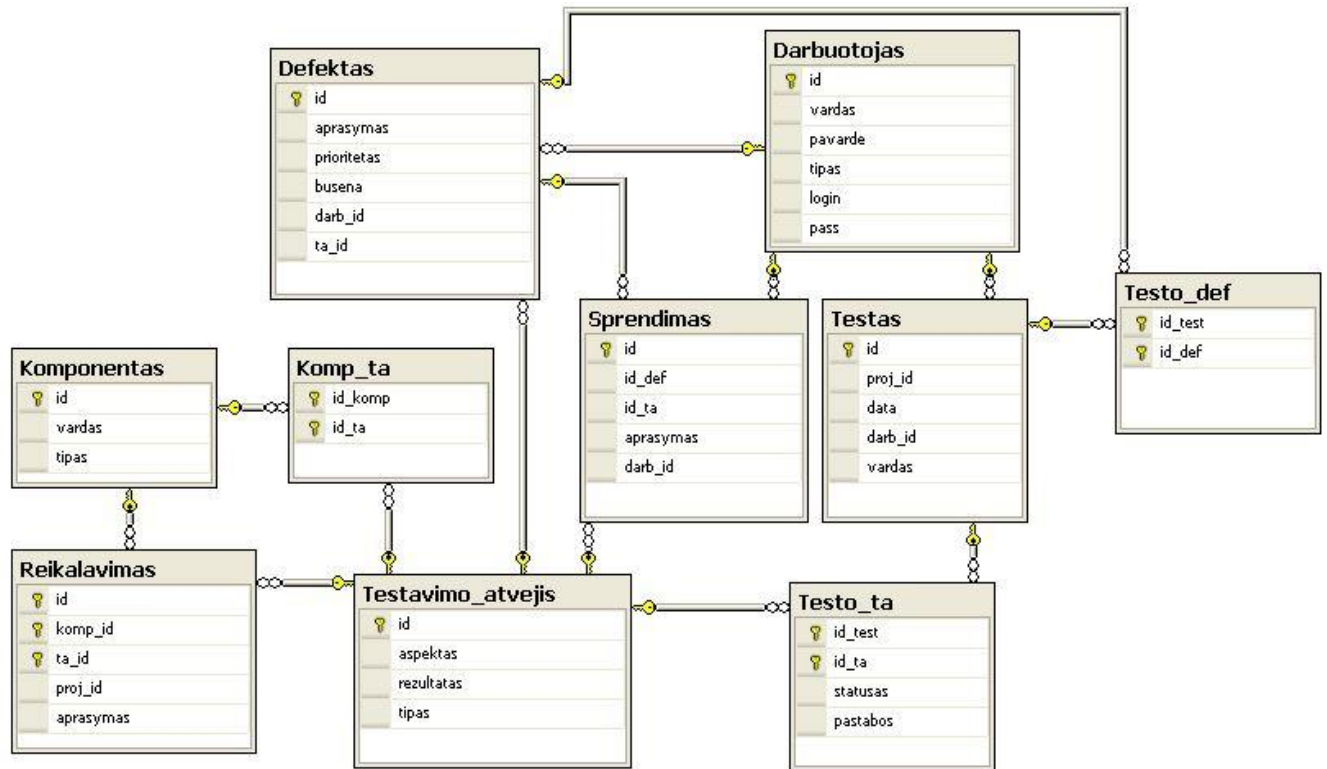
3.3 Duomenų bazės modelis

Pagal dalykinės srities klasių modelį sudarytas duomenų bazės modelis. Saityno IS VS testavimo metodikos prototipo realizacijoje naudojama duomenų bazė yra sudaryta iš dešimties lentelių. Lentelėse saugoma dalykinės srities informacija, su kuria dirbama sistemoje (9 lentelė).

Lentelė 9. Duomenų bazės lentelės

Lentelė	Saugoma informacija
Darbuotojas	Sistemos naudotojo, dirbančio su sistema duomenys.
Defektas	Duomenys apie rastą VS defektą.
Komponentas	Duomenys apie VS komponentą.
Komp_ta	Duomenys apie komponentus ir jų testavimo atvejus.
Reikalavimas	Specifinis užsakovo VS reikalavimas .
Sprendimas	Duomenys apie siūlomus ir priimtus sprendimus defektui spręsti.
Testas	Duomenys apie testą.
Testo_ta	Duomenys apie konkretaus testo testavimo atvejus.
Testo_def	Duomenys apie konkretaus testo defektus.
Testavimo_atvejis	Duomenys apie testavimo atvejus/euristikas.

Sudaryta duomenų bazės struktūra yra pateikta 23 paveikslėlyje.



Šaltinis: sukurta autorės.

23 pav. VS testavimo metodikos prototipo duomenų bazės schema

Kiekvienoje duomenų bazės lentelėje yra saugoma skirtingo objekto svarbiausia informacija apie jį. Atliekant sprendimo realizaciją svarbu išlaikyti numatytą duomenų bazės struktūrą ir ryšius tarp lentelių.

3.4 Realizacijos išvados

1. Metodikoje apibrėžta rekomendacinė, universali standartinių komponentų aibė ir sudarytas tipinių testavimo atvejų (euristikų) rinkinys leidžia be papildomo pasirengimo pradėti bazinius saityno IS vartotojo sąsajos testavimo darbus.
2. Siūlomos saityno IS VS testavimo metodikos etapai detalizuoti išskiriant procesus. Parengta testavimo proceso žingsnių specifikacija sukurianti prielaidas standartizuoti procesą bei kurti specializuotus įrankius.

4 VS TESTAVIMO METODIKOS EKSPERIMENTINIS TYRIMAS

4.1 Eksperimento planas

Siekiant eksperimentiškai įvertinti sudarytą saityno IS vartotojo sąsajos testavimo metodiką, buvo sukurtas ja grindžiamas testavimo sistemos prototipas. Magistrinio darbo metu buvo siekiama sukurti tokį įrankį, kuris būtų pranašesnis už kitus analizuotus testavimo įrankius ir padėtų atskleisti siūlomos metodikos privalumus, kuriais nepasižymi egzistuojantys vartotojo sąsajos testavimo metodai. Kaip jau buvo minėta, tyrimo metu buvo siekiama sukurti sprendimą ir jį įgyvendinti taip, kad jis būtų specializuotas būtent saityno informacinių sistemų (grafinei) vartotojo sąsajai testuoti. Pasirinktas rankinio testavimo principas, kuriam būdingas didesnis testavimo laiko sąnaudas buvo siekiama sumažinti sprendimais priimtais siūlomoje metodikoje ir sistemos prototipe. Metodika kurta euristinio vertinimo metodo pagrindu, tačiau sprendžiant šiam metodui būdingas trūkstamų sąsajos elementų nepastebimumo ir sprendimų priėmimo problemas. Taigi, eksperimentas susideda iš kelių dalių, kurių metu buvo siekiama atlikti:

- elementų nepastebimumo problemos išsprendimo įvertinimą;
- metodikos siūlomo rankinio testavimo laiko sąnaudų sumažinimo įvertinimą;
- defektų šalinimo sprendimų priėmimo proceso pagreitinimo įvertinimą;

Eksperimento metu operuojama metodikos realizacijos dalyje apibrėžtomis komponentų ir testavimo atvejų (euristikų) aibėmis, kad didesnis dėmesys būtų skiriamas testavimo vykdymo, o ne pasiruošimo jam etapui. Šios duomenų aibės yra tipiniai vartotojo sąsajos komponentai ir testavimo atvejų rinkiniai tinkantys įvairaus tipo saityno informacinėms sistemoms.

Siekiant išanalizuoti ir įvertinti metodiką bus naudojama anketinė apklausa ir atliktas praktinis metodikos taikymas. Panaudojant sukurtą prototipą metodikos tyrimas atliekamas realiuose saityno projektuose. Tyrimui pasirinktas tikslinis sprendimo poaibis – savitarnos svetainės. Pasirinktų savitarnos svetainių testavimą vadovaujantis siūloma saityno IS vartotojo sąsajos testavimo metodika atlikti ir metodikai įvertinti pasirinkti kokybės inžinieriai. Kokybės inžinieriai turės atsakyti į anketoje pateiktus klausimus įvertinant metodikos apibrėžtą testavimo procesą ir darbą su sistemos prototipu. Bus atlikta anketinės apklausos analizė, o gauti rezultatai bus panaudoti tolimesniam metodikos vystymui.

4.2 Eksperimento eiga ir rezultatai

Tyrimui pasirinktos trys realios saityno IS – mobiliųjų komunikacijų bendrovių savitarnos svetainės (lentelė 10), šių bendrovių klientams Lietuvos rinkoje.

Lentelė 10. Tyrimo metu testuotos saityno IS

Suteiktas kodas	Pavadinimas	Nuoroda internete	Įmonė
<i>Savitarna1 (s1)</i>	Mano BITĖ	https://www.manobite.lt	UAB „Bitė Lietuva“
<i>Savitarna2 (s2)</i>	Omnitel savitarna	https://savitarna.omnitel.lt	UAB „Omnitel“
<i>Savitarna3 (s3)</i>	Mano Tele2	https://kt.tele2.lt	UAB „Tele2“

Šaltinis: sukurta autorės.

Šių savitarnos svetainių vartotojo sąsajas ištestuoti vadovaujantis siūloma metodika buvo paprašyta 3 kokybės inžinierių (suteikti kodai – T1, T2 ir T3). Kokybės inžinieriai turi patirties saityno savitarnos svetainių vartotojo sąsajos testavime taikant įvairias metodikas, todėl jie galės palyginti siūlomą metodiką su anksčiau jų praktikoje taikytomis.

Pirma tyrimo dalis buvo skirta įvertinti kiek padidėja vartotojo sąsajos elementų pastebimumas siūlomoje metodikoje, kuri grindžiama vartotojo sąsajos išskaidymu į komponentus. Šioje tyrimo dalyje testuotojai pirmiausiai turėjo nurodyti kokius vartotojo sąsajos komponentus, kuriuos būtina testuoti, sudaro *Savitarna1*, *Savitarna2* ir *Savitarna3* prisijungimo puslapius. Antra užduotis, kurią kokybės inžinieriai atliko šioje tyrimo dalyje, buvo pakartotinai pažiūrėti ir išskirti iš metodikos siūlomos universalių komponentų aibės komponentus, kurie nebuvo nurodyti testuojamoms savitarnos svetainėms atliekant pirmą užduotį. Komponentai, kurie buvo išskirti atliekant šias užduotis, pateikti 11 lentelėje.

Lentelė 11. Testuotojų išskirti VS komponentai

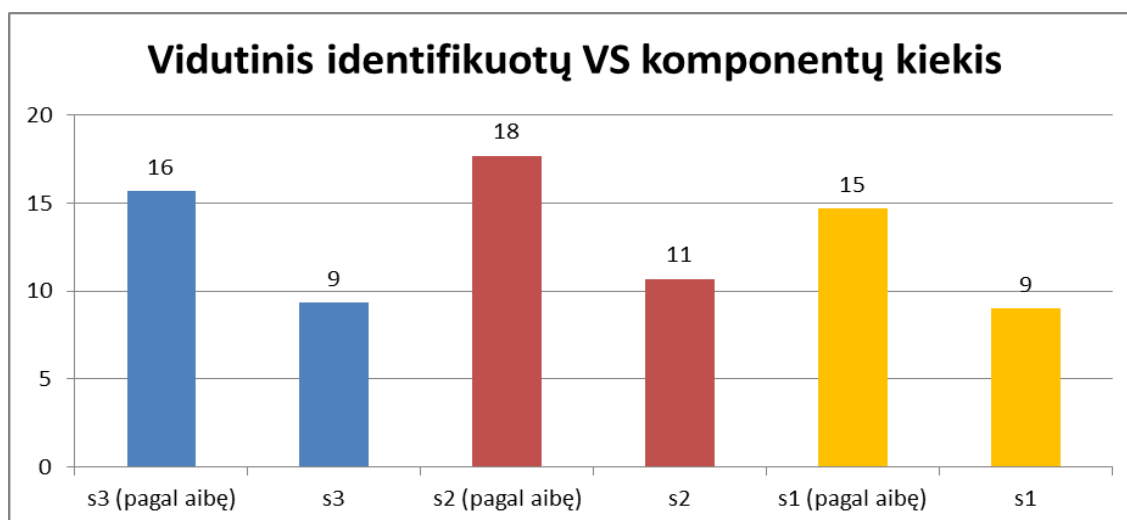
VS komponentai	nesivadovaujant universalių komponentų aibe			pagal universalių komponentų aibę		
	T1	T2	T3	T1	T2	T3
<i>Savitarna1</i>						
langai		+	+	+		
nuorodos	+	+	+			
teksto blokai	+	+	+			
mygtukai	+	+	+			
menu		+	+	+		
navigacija		+		+		+
svetainės logotipas				+	+	+
viršutinė eilutė						+

	nesivadovaujant universalių komponentų aibe			pagal universalių komponentų aibę		
VS komponentai	T1	T2	T3	T1	T2	T3
<i>Savitarna1</i>						
paieška				+	+	+
formos		+	+			
apatinė eilutė	+		+		+	
piktogramos	+				+	+
pranešimų tekstai		+	+	+		
pagalbos priemonės			+	+	+	
Testuotojų išskirti						
įvedimo laukai	+	+	+			
punktavimas		+				
spalvų paletė			+			
<i>Savitarna2</i>						
langai		+	+	+		
nuorodos	+	+	+			
teksto blokai	+	+	+			
mygtukai	+	+	+			
menu		+	+	+		
navigacija		+		+		+
svetainės logotipas				+	+	+
viršutinė eilutė				+	+	+
paieška						+
formos			+	+	+	
apatinė eilutė	+	+	+			
piktogramos	+				+	+
pranešimų tekstai		+	+	+		
pagalbos priemonės	+		+		+	
Testuotojų išskirti				Pasinaudojus savitarna1 DB		
įvedimo laukai	+	+	+			
punktavimas		+		+		+
spalvų paletė			+	+	+	
paveiksliukai	+	+	+			
pažymimosios varnelės		+				
langai		+	+	+		
<i>Savitarna3</i>						
nuorodos	+	+	+			
teksto blokai	+	+	+			

	nesivadovaujant universalių komponentų aibe			pagal universalių komponentų aibę		
VS komponentai	T1	T2	T3	T1	T2	T3
mygtukai	+	+	+			
<i>Savitarna3</i>						
menu		+	+	+		
navigacija		+		+		+
svetainės logotipas				+	+	+
viršutinė eilutė						
paieška						+
formos			+	+	+	
apatinė eilutė				+	+	+
piktogramos						
pranešimų tekstai		+	+	+		
pagalbos priemonės	+		+		+	
Testuotojų išskirti				Pasinaudojus savitarna1 ir savitarna2 DB		
įvedimo laukai	+	+	+			
punktavimas		+		+		+
spalvų paletė			+	+		
paveiksliukai	+	+	+			
pažymimosios varnelės		+				+

Šaltinis: sukurta autorės.

Vartotojo sąsajos specifikavimo metu nesivadovaujant ir vadovaujantis pagalbine aibe išskirtų VS komponentų kiekio palyginimas pateiktas 24 paveikslėlyje.



Šaltinis: sukurta autorės.

24 pav. VS specifikavimo metu išskirtų komponentų kiekio palyginimas

Antra tyrimo dalis buvo skirta įvertinti kiek užtrunkama laiko vartotojo sąsajos testavimui pakartotinai panaudojant VS komponentų, testavimo atvejų/euristikų, defektų ir jų sprendimo būdų aibes skirtingoms vartotojo sąsajoms testuoti. Šioje dalyje kiekvienas iš kokybės inžinierių nuosekliai, vadovaujantis metodikos žingsniais atliko visų trijų savitarnos svetainių vartotojo sąsajos testavimą. Siekiant sumažinti testavimo apimtį, buvo testuotai tik savitarnos svetainių prisijungimo ir pagrindinis svetainės puslapis po prisijungimo. Kokybės inžinierių buvo paprašyta įsidėmėti, kiek laiko užtruks kiekvienos vartotojo sąsajos testavimas vadovaujantis metodika ir jos etapai. Skirtingų vartotojų sąsajų testavimą kokybės inžinieriai atliko vadovaujantis šiais etapais:

A Pasiruošimas testuoti

- Vartotojo sąsajos specifikavimas;
- Testų kūrimas;
- Sprendimų pateikimas.

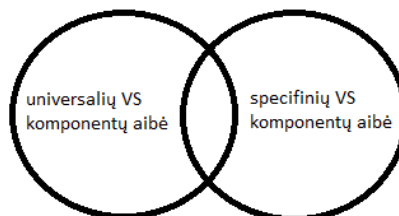
B Testavimo vykdymas

- Validavimas ir verifikavimas;
- Sprendimų paieška.

Defektų verifikavimo etapo kokybės inžinieriai atlikti su realiomis savitarnos sistemomis negalėjo, nes reikalingas defektų pašalinimo procesas. Defektų pašalinimo proceso dalyviai – kiti sistemos kūrimo komandos nariai, kuria tyrime nedalyvavo.

Pirmiausia buvo vykdomas pasiruošimo testavimui etapas. Visi žemiau aprašyti žingsniai buvo vykdomi su testuojamomis saityno *Savitarna1*, *Savitarna2* ir *Savitarna3* informacinėmis sistemomis.

Vartotojo sąsajos specifikavimas. Kokybės inžinieriai vadovaujantis metodikoje siūloma universalių komponentų aibe išskyrė kuo daugiau testuojamos savitarnos svetainės vartotojo sąsajos komponentų, kuriuos planuojama testuoti. Testuotojų išskirtos universalių VS komponentų aibės papildytos specifinių VS komponentų aibėmis – $\{universalūs\ VS\ komponentai\} \cup \{specifinių\ VS\ komponentai\}$ (25 paveikslėlis). Šias aibes sudarė testuotojai įvardindami komponentus, kurių nebuvo universalių komponentų aibėje ir jie buvo būdingi konkrečiai testuojamai savitarnos VS.



Šaltinis: sukurta autorės.

25 pav. Universalių ir specifinių VS komponentų aibių sankirta

Universalūs ir specifiniai VS komponentai buvo įtraukti į duomenų bazę sistemos prototipo pagalba. Svarbu paminėti, kad visi trys testuotojai testuojant pirmos sistemos vartotojo sąsają naudojami skirtingomis komponentų duomenų bazėmis, todėl jie negalėjo matyti vienas kito pastebėtų ir įtrauktų komponentų.

Testų kūrimas. Testuojamose savitarnos svetainėse egzistuojantiems ir į duomenų bazę įtrauktiems vartotojo sąsajos komponentams turi būti priskirti testavimo atvejai (euristikos). Šis etapas, kai sukuriama euristikos ir jos priskiriamos komponentams metodikoje, yra antras. Į duomenų bazę įvestiems universaliems VS komponentams buvo priskirti statiniai testavimo atvejai (euristikos), kurie yra pasiūlyti metodikoje. Trūkštamus ir dinامينius testavimo atvejus sugalvojo ir komponentams priskyrė testuotojai (12 lentelė).

Lentelė 12. Testuotojų sukurtos euristikos VS komponentams

	T1	T2	T3
KOMPONENTAS	EURISTIKOS		
<i>įvedimo laukai</i>	E1. Be klaidos pranešimo yra aišku ar informacijos nurodymas įvedimo lauke yra privalomas.	E3. Yra aišku koks tekstas turi būti nurodytas įvedimo laukuose.	E7. Tarp įvedimo laukų galima „vaikščioti“ klaviatūros pagalba. E8. Be papildomos informacijos yra aišku koku formatu reikia nurodyti informaciją datos ar laiko įvedimo laukuose.
<i>spalvų paletė</i>	-	-	E9. Saityno IS spalvų paletę sudaro ne daugiau 4 spalvos, viena iš jų – dominuojanti.
<i>paveikslukai</i>	E2. Teminis paveikslėlis neturėtų „iškristi“ iš konteksto savo spalvomis.	E4. Yra aiški paveiksluko vieta turinyje.	E10. Informaciniai paveikslukai neturi blaškyti dėmesio, būti pastebėti pirmiau nei funkcionalumas.
<i>punktavimas</i>	-	E5. Iš punktavimo simbolio yra aišku, kad vadinami punktai.	-
<i>pažymimosios varnelės</i>	-	E6. Yra aiški pažymimosios varnelės reikšmė ir jos funkcionalumas.	-

Šaltinis: sukurta autorės.

Testuotojai testuojant pirmos sistemos vartotojo sąsają naudojami skirtingomis euristikų duomenų bazėmis, todėl jie negalėjo matyti vienas kito įtrauktų testavimo atvejų/euristikų.

Sprendimų pateikimo etapas. Kokybės inžinieriai turėjo sugalvotoms euristicoms priskirti sprendimus jeigu jie yra žinomi. Kokybės inžinierių sugalvoti TA (euristikų) defektų sprendimo būdai pateikti 13 lentelėje.

Lentelė 13. Euristicoms priskirti sprendimo būdai

Euristikos nr.	Sprendimo nr.	Sprendimo būdas
E1	S1	Privaloma laukai turi būti pažymėti specialiu simboliu, kuris būtų paaiškintas apie informacijos įvedimo į laukus būtinumą.
E2	S2	Teminiuose paveikslėliuose turėtų būti panaudojamos spalvos artimos saityno IS dizainui.
E3	S3	Prie kiekvieno įvedimo lauko yra nurodoma kokia informacija ir kokių formatu turi būti įvesta.
E4	S4	Paveikslukas turi būti lygiuojamas su kitais kurinio elementais ir jam turi būti pritaikyti mažesni balti tarpai, nei tarp kitų elementų.
E5	S5	Naudoti akivaizdžius punktavimo simbolius, pvz. varneles arba taškus.
E6	S6	Prie pažymimųjų varnelių realizuoti paaiškinamuosius tekstus.
E7	S7	Realizuoti galimybę „vaikščioti“ po puslapį <i>tab</i> mygtuko pagalba.
E8	S8	Datos ar laiko informacijos nurodymo formatas gali būti apibrėžiamas pagalbinio įrankio pagalba (pvz. datą pasirinkti iš kalendoriaus).
E9	S9	Vadovautis „ <i>brandbook</i> “.
E10	S10	Informaciniai paveikslukai neturi būti zonose į kurias pirmiausiai žiūri vartotojas (pvz. turinio centre). Šiose zonose turėtų būti funkcionalumas.

Šaltinis: sukurta autorės.

Pirmos savitarnos svetainės vartotojo sąsajos testavimo metu kokybės inžinieriai euristicoms sugalvotus sprendimus registravo skirtingose duomenų bazėse, todėl jie negalėjo matyti vienas kito sukurtų sprendimų.

Kai visos reikalingos duomenų aibės buvo suvestos į duomenų bazę, kokybės inžinierius prototipo pagalba susikūrė testą testuojamai savitarnos svetaine. Testą sudaro testavimo atvejai (euristikos), kurios įtraukiamos į testą pagal testuotojo nurodytus VS komponentus, kurie jo nuomone egzistuoja testuojamoje savitarnos svetainėje. Kokie VS komponentai buvo testuojami kiekvieno kokybės inžinieriaus savitarnos svetainėse rezultatai pateikti 10 lentelėje.

Verifikavimo ir validavimo etapas. Testavimo vykdymo etapas prasidėjo nuo TA (euristikų) validavimo ir verifikavimo. Kokybės inžinieriai testavo savitarnos svetainės pagal TA (euristikas). Atitikimus euristicoms žymėjo teigiamai, o neatitikimams – registruodavo defektus.

Sprendimų paieška. Registruotiems defektams kokybės inžinieriai galėjo priskirti sprendimo būdus ir pateikiamo sąrašo arba nurodyti naujus atsižvelgiant į situaciją. Šie veiksmai atliekami

sprendimų paieškos etape. Vartotojo sąsajos defektai ir jų šalinimui pritaikyti sprendimai pateikti 14 lentelėje.

Lentelė 14. Defektai ir jų šalinimo sprendimai

			T1	T2	T3
Defekto nr.	Pažeista euristika	Defekto aprašymas	Pritaikytas sprendimas		
<i>Savitarna1</i>					
D1	E1	Ne visai aišku, ką reiškia raudonos žvaigždutės prie įvedimo laukų registracijos formoje.	S1		
D2	E3	Nėra aišku kokie reikalavimai yra taikomi slaptažodžio formatui registracijos formoje.		S3	
D3	E8	Ne visada formoje rodomi pagalbinių informacijos įvedimo įrankiai (pvz. kalendoriukas).			S8
D4	E5	Ne visai aišku, ar pateikiamas tekstas yra punktuojamas ar dialogo kalba.		S5	
D5	TA36	Pamiršto slaptažodžio atveju, nenurodžius el. pašto ir paspaudus mygtuką „siųsti“ pateikiamas vartotojui nesuprantamas klaidos pranešimas – „ERR_INVALID_LOGIN_NAME“.		+	S11. Klaidos pranešimas turi būti pateikiamas vartotojui suprantamo paaiškinimo forma, nurodant ir vietą, kurioje įvyko klaida.
D6	TA08	Ne visada aišku ar tai neaktyvi nuoroda ar nuorodos stilių turintis tekstas, pvz. „Pamiršote slaptažodį?“	S12. Netaikyti neaktyvių nuorodų stilių tekstams ar nuorodoms, kurios yra aktyvios.	S13. Šiuo atveju siūlau keisti į mygtuką arba nuorodos stilių.	S14. Tekstams-nuorodoms atidarančioms vidinius langus naudoti identišką nuorodos stilių.
D7	TA29	Iš klaidos pranešimo nėra aišku kuriame įvedimo lauke įvesta nekorektiška ar neįvesta informacija (pvz. rašo, kad neįvestas slaptažodis, o raudonai		S15. Korektiškas tekstas ar vieta, kurioje neįvyko klaida neturi būti	S16. Duomenų įvedimo formose aptikta klaida turi būti pažymima ta

			T1	T2	T3
Defekto nr.	Pažeista euristika	Defekto aprašymas	Pritaikytas sprendimas		
		pažymimas įvesto el.pašto adreso tekstas).		pažymimas kaip klaida.	spalva, kuria pateiktas klaidos pranešimas.
D8	TA37	Informacinis pranešimas informuojantis apie sėkmingą registraciją nesiskiria nuo paprasto teksto.	S17. Informacinis pranešimas gali būti išskiriamas specialia piktograma pranešimo pradžioje.	S18. Informacinis pranešimas turėtų būti įrėmintas.	S19. Informaciniai ir klaidų pranešimai turėtų būti išskiriami priešingomis spalvomis.
D9	TA35	Informacinis pranešimas informuojantis apie sėkmingą registraciją pateikiamas zonoje į kurią nežiūri vartotojas.	+	S20. Šiuo atveju, kai yra daugiau nei vienas informacinis pranešimas turėtų būti pateikiami toje pačioje turinio vietoje.	+
D10	TA34	Informacinis pranešimas informuojantis apie sėkmingą užsiregistravimą pateikiamas po antrašte, kuri neatitinka pranešimo pateikiamo po ja (pvz.: Savitarnos svetainėje galėsite: Registracijos duomenys sėkmingai išsaugoti!“).	+	S35. Klaidų ir informaciniai pranešimai turėtų būti pateikiami kuo arčiau funkcionalumo su kuriuo jie yra susiję.	
D11	TA01	Pagrindiniame lange po prisijungimo pateikiamas pašto funkcionalumas du kartus, tampa neaišku ar funkcionalumas skirtingas ar vienodas.	S37. Pasikartojantį funkcionalumą pašalinti iš to puslapio.	S29. Vieną iš funkcionalumų galima išskelti į meniu punktus (realizuoti pasiekimą iš kito puslapio).	+
D12	TA20	Ne visada aišku kurioje sistemos vietoje yra vartotojas.	S30. Tarp meniu ir puslapio turinio galima pateikti nuorodų kelią iki		S31. Navigacijos kelias ir šoninis

			T1	T2	T3
Defekto nr.	Pažeista euristika	Defekto aprašymas	Pritaikytas sprendimas		
			vietos kurioje yra vartotojas.		menu visada turi būti pateikiamas vartotojui nepriklausomai nuo to kuriame puslapyje jis yra.
D13	TA19	Ne visada iš elementų stiliaus yra aišku, kuris menu punktas yra pažymėtas, aktyvus.	S33. Visoje sistemoje aktyviems elementams taikyti vienodą stilių. Šiuo atveju aktyviems menu ir įjungtos/siūlomos paslaugos skirtukams reikalinga suvienodinti teksto stilių.		
D14	TA04	Vartotojui inicijavus veiksmą, kuris trunka ilgiau nei puslapio atidarymas, neaišku ką veikia sistema (pvz. paspaudus mygtuką prisijungti ar atsijungti).	S25. Kai sistema <i>back-end</i> sistemose vykdo vartotojo inicijuotą veiksmą arba veiksmas užtrunka ilgiau nei įprastai vartotojas turi būti apie tai informuojamas ir paprašomas palaukti.		
Savitarna2					
D15	E3	Nėra aišku koku formatu turi būti nurodomas prisijungimo vardas prisijungimo formoje.	S3		
D16	TA04	Siunčiant SMS vartotojui neaišku ką veikia sistema.	S25		
D17	TA06	Vartotojui patekus į išorinį puslapį/sistemą ne visai aišku, kaip grįžti į savitarną.	S26. Nuorodos į išorinius puslapius turi būti atidaromos naujuose naršyklės skirtukuose.	S27. Kitų puslapių ar sistemų turinys vartotojo peržiūrai gali būti pateikiamas skirtinguose naršyklės languose.	S28. Informacija iš išorinių puslapių gali būti pateikiama modaliniuose languose.
Savitarna3					
D18	E4	Neaišku kokiam tikslui aplink paveiksluką, kuris neįsikomponuoja į turinį, palikta daug baltos/tuščios vietos.		S4	
D19	E10	Prisijungimo/registracijos			S10

			T1	T2	T3
Defekto nr.	Pažeista euristika	Defekto aprašymas	Pritaikytas sprendimas		
		funktionalumo puslapyje pirmiausia dėmesys skiriamas paveikslui, o tik po to ieškoma prisijungimo galimybės.			
D20	TA36	Nėra aišku, kuris iš pateiktų klaidų ir informacinių pranešimų įvyko konkrečiu atveju.	S11	S11	S21. Konkrečiam klaidos atvejui turi būti pateikiamas tik vienas klaidos pranešimas.
D21	TA29	Nėra aišku kuriame įvedimo lauke įvyko klaida.	S15	S15, S16	S15, S16
D22	TA02	Pamiršus slaptažodį arba naujo kliento atveju nenurodžius tel. nr. ir paspaudus mygtuką „Tęsti“ klaidų pranešimai pateikiami ne vientiso dizaino languose.	S22. Langų stilius turi būti kaip kitų langų esančių sistemoje.	S23. Šiuo atveju klaidų pranešimus galima pateikti ir pranešimo forma o ne atskiruose languose.	S24. Langų stilius turi būti pritaikytas atsižvelgiant į kitus sistemos langus.
D23	TA04	Ne visada aišku ką „veikia“ sistema, ko vartotojui laukti (pvz. atsijungiant neaišku ar jau vykdomas vartotojo atjungimas nuo sistemos ar nepavyko paspausti mygtuko).	S25.		
D24	TA06	Vartotojui patekus į išorines sistemas ar puslapius nesuteikiama galimybė grįžti į sistemą prie kurios buvo prisijungęs (pvz. pasirinkus virtualios pagalbos meniu punktą, norint grįžti į savitarną reikalinga iš naujo prisijungti).	S26	S27, S28, S29	S29
D25	TA20	Ne visada aišku kurioje sistemos vietoje yra vartotojas.	S30		
D26	TA38	Ne visada aišku, kodėl vartotojas mato	S32. Iškreptas meniu su vienu meniu punktu sąraše galėtų būti pakeistas į šoninį meniu.		
D26	TA38	Ne visada aišku, kodėl vartotojas mato	S34. Lentelėse, kuriose nėra pateikiama informacija,		

			T1	T2	T3
Defekto nr.	Pažeista euristika	Defekto aprašymas	Pritaikytas sprendimas		
		tuščias lenteles.	galima nurodyti paaiškinimą, kad joje nėra įrašų arba informacija nėra prieinama dėl tam tikrų priežasčių.		
D27	TA15	Ne visada aiški vartotojo sąsajos elementų paskirtis (pvz. kontaktų pridėjimo lentelėje pažymimosios akutės).	S36. Pažymimųjų akučių elementai naudojami vieno elemento pasirinkimui, pažymimųjų laukelių – keliems elementams pasirinkti.		

Šaltinis: sukurta autorės.

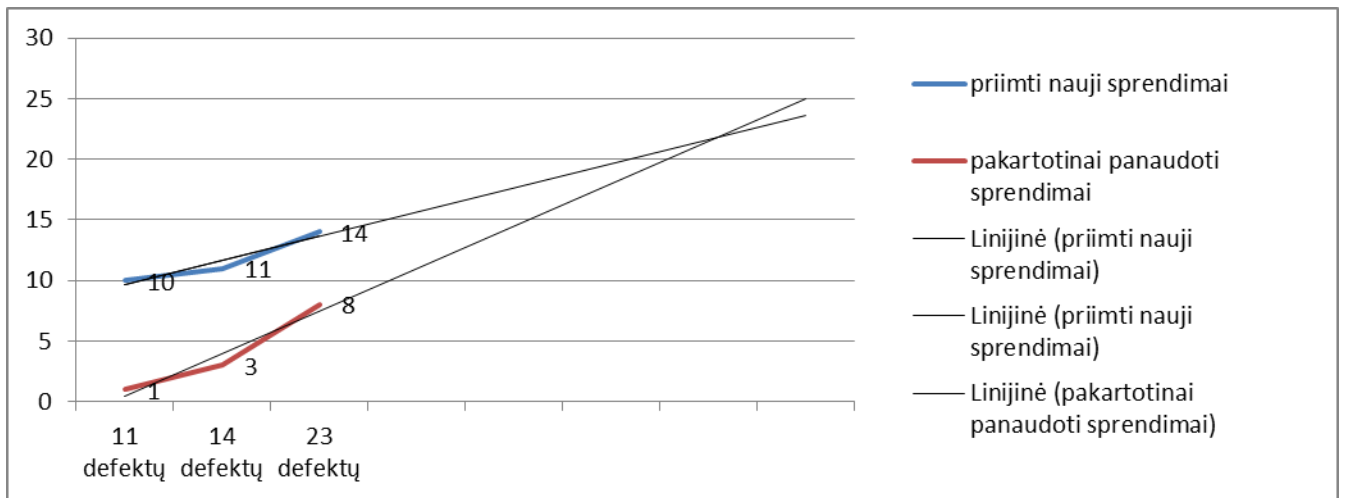
Visus aukščiau aprašytus žingsnius kokybės inžinieriai pirmiausiai atliko su *savitarna1*, po to su *savitarna2* ir galiausiai su *savitarna3* vartotojo sąsajomis. Skirtumas buvo tik tas, kad testuojant *savitarna2* kokybės inžinieriai galėjo naudotis *savitarna1* testavimo metu tiek savo, tiek kitų tyrimo dalyvių sudarytomis komponentų, testavimo atvejų ir defektų aibėmis. Testuojant *savitarna3* buvo naudojamos *savitarna1* ir *savitarna2* testavimo metu sudarytomis komponentų, testavimo atvejų ir defektų aibėmis. Toks pakartotinis aibių panaudojimas skirtingoms vartotojo sąsajoms testuoti turėjo leisti sumažinti testavimo laiko sąnauda skirtas pasiruošimui ir testavimo vykdymui. Kiekvieno testuotojo pakartotinai panaudotų ir naujai sukurtų defektų šalinimo būdų sprendimų skaičiai pateikti 15 lentelėje.

Lentelė 15. VS komponentų, TA, defektų ir sprendimo būdų kiekiai

Testuota VS		Išskirta komponentų	Atlikti testavimo atvejai	Užregistruota defektų	Pateikti nauji sprendimai	Pritaikyti sprendimai
<i>Savitarna1</i>	T1	12	40	10	9	1
	T2	13	44	12	10	2
	T3	15	47	11	10	1
Vidurkis:		13	44	11	10	1
<i>Savitarna2</i>	T1	15	50	3	1	2
	T2	17	51	3	1	2
	T3	19	50	3	1	2
Vidurkis:		17	50	3	1	2
<i>Savitarna3</i>	T1	14	44	8	3	5
	T2	18	44	9	3	6
	T3	16	46	9	4	5
Vidurkis:		16	45	9	3	5

Šaltinis: sukurta autorės.

Naujai priimtų ir pakartotinai panaudotų defektų šalinimo sprendimo būdų kiekio kitimas atsižvelgiant į klaidų kiekį pateiktas 26 grafike.



Šaltinis: sukurta autorės.

26 pav. Naujai priimtų ir pakartotinai panaudotų sprendimų kiekio kitimas defektų atžvilgiu

Iš grafiko matyti, kad didėjant aptiktų defektų skaičiui proporcingai didėja pakartotinai panaudotų sprendimų ir proporcingai mažėja naujai priimamų sprendimų.

4.3 Eksperimento rezultatų apibendrinimas

Euristinio vertinimo metodo, kurio pagrindu buvo kurta metodika, rankinio testavimo būdo, kurio principais buvo kurta metodika, ir egzistuojančių testavimo įrankių palyginimas su siūloma VS testavimo metodika ir sukurtu prototipu pateiktas 16 lentelėje.

Lentelė 16. Žinomų testavimo metodų, būdų ir įrankių palyginimas su siūloma metodika

Lyginimo kriterijai	TESTAVIMO								
	METODAS	BŪDAS	ĮRANKIAI						saityno IS VS testavimo
	Euristinis vertinimas	Rankinis testavimo būdas	Testuff	Tosca	eggPlant	Test Complete	JIRA	Bugzilla	
Rankinis testavimo būdas	taip	taip	taip	taip	taip	taip			taip
Dažnai yra žinomus defektų šalinimo būdai	ne		ne	ne	ne	ne	ne	ne	taip
TA turi ryšį su komponentais	ne		ne	taip	taip	ne			taip
Defektas turi ryšį su komponentais	ne		ne	ne	ne	ne	taip	taip	taip
Defektų valdymas			taip	taip	ne	ne	taip	taip	ne

	TESTAVIMO								
	METODAS	BŪDAS	ĮRANKIAI						
Kaupiama TA duomenų bazė			ne	ne	ne	ne			taip
Testų pakartotinis panaudojamumas			ne	ne	ne	ne			taip
Testavimo atvejų valdymas			taip	taip	taip	taip			taip

Euristinio metodo trūkumas – testuojant nepastebimi trūkstami vartotojo sąsajos elementai. Siūloma metodika grindžiama grafinės vartotojo sąsajos išskaidymu į komponentus. Pirmoje tyrimo dalyje atliktas vartotojo sąsajos komponentų išskyrimas. Palyginus kokybės inžinierių nuožiūra išskirtus komponentus ir komponentus išskirtus remiantis metodikos siūloma universalių ir kitų testavimo metu sudaryta komponentų aibe, parodė, kad pastaruoju būdu komponentų išskiriama ir pastebima daugiau. Nuolatinis komponentų duomenų bazės pildymas naujais komponentais leis sukaupti pakankamą universalių ir specifinių komponentų aibes, kurias bus galima taikyti skirtingoms saityno IS vartotojo sąsajoms testuoti. Nors remiantis komponentų aibe jų ir buvo pastebima daugiau, tačiau kokybės inžinieriai dažnai pasirinkdavo skirtingą komponentų skaičių testavimui. Taip nutikdavo todėl, kad kokybės inžinieriai komponentus pasirinkdavo tuos kuriuos pamatydavo vartotojo sąsajoje arba kurie jų nuomone gali turėti didelę įtaką sistemos panaudojamumui. Siekiant išspręsti šią problemą metodikoje būtina išskirti, kurie komponentai iš universalių komponentų aibės yra privalomi vartotojo sąsajose norint užtikrinti aukštą jos panaudojamumą.

Vienas svarbiausių sukurtosios metodikos ypatumų – galimybė kaupti žinias ne tik apie galimas saityno IS klaidas, bet ir apie šių klaidų taisymo galimybes. Antra tyrimo dalis parodė, kad sukauptos defektų šalinimo būdų žinios gali būti sėkmingai pakartotinai panaudojamos skirtingų vartotojo sąsajų defektams šalinti. Testuojant skirtingas vartotojo sąsajas pakartotinai panaudojant jau turimus komponentus, euristikas, defektus ir jų sprendimo būdus parodė, kad laiko sukūrimo ir sprendimo priėmimo metodikos žingsniuose sugaištama mažiau. Akivaizdu, kad tai padeda sumažinti testavimo rankiniu būdu laiko sąnaudas, kai kiekvienai saityno IS vartotojo sąsajai testuoti reikalaujama mažiau pastangų analitiniuose ir kūrybiniuose sprendimuose.

Preliminarus metodikos tyrimas parodė, jog sprendimas gali būti realizuotas ir turėtų leisti pagerinti ne tik sąsajos defektų aptikimo efektyvumą, bet ir jų sprendimo greitį, pritaikytų pataisymų kokybę.

5 REZULTATŲ APIBENDRINIMAS IR IŠVADOS

1. Analizės metu atskleista, jog saityno svetainių ir informacinių sistemų sąsaja pasižymi keletu darbo būdo (saityne *naršoma*) ir techninės aplinkos lemiamų ypatybių. Dėl to formuojasi specifinė, tik tokio pobūdžio sistemoms būdingų sąsajos defektų ir galimų jų sprendimo būdų aibė.

2. Vartotojo sąsajos testavimo ir vertinimo metodų analizė parodė, kad šiuo metu vyraujantys metodai nėra pakankamai tinkami saityno informacinių sistemų vartotojo sąsajai testuoti. Vis dėlto žinomos metodikos pasižymi tam tikrais privalumais, kuriais verta pasinaudoti. Šio tyrimo metu buvo sukurta metodika, kurioje remiamasi testavimo standartais, panaudojamas rankinio testavimo principas bei pritaikytas euristinio vertinimo metodas.

3. Pristatytoje metodikoje apibrėžta rekomendacinė, universali standartinių komponentų aibė ir sudarytas tipinių testavimo atvejų (euristikų) rinkinys leidžia be papildomo pasirengimo pradėti bazinius saityno IS vartotojo sąsajos testavimo darbus.

4. Siūlomos saityno IS VS testavimo metodikos etapai detalizuoti, išskiriant procesus. Parengta testavimo proceso žingsnių specifikacija, sukurianti prielaidas standartizuoti procesą bei kurti specializuotus įrankius.

5. Sukurtas metodika grindžiamas testavimo sistemos prototipas buvo panaudotas metodikos tyrimui realiuose projektuose. Tyrimo metu gauti rezultatai gali būti panaudoti tolimesniam metodikos vystymui. Tyrimas parodė, jog taikant metodiką pagerėjo vartotojo sąsajos komponentų pastebimumas ir sumažėjo laiko sąnaudos skiriamos defektų šalinimo būdų sprendimų priėmimui.

6. Preliminarus metodikos tyrimas parodė, jog sprendimas gali būti realizuotas ir turėtų leisti pagerinti ne tik sąsajos defektų aptikimo efektyvumą, bet ir jų sprendimo greitį, atliktų pataisymų kokybę. Vienas svarbiausių sukurtosios metodikos ypatumų – galimybė kaupti žinias ne tik apie galimas saityno IS klaidas, bet ir apie šių klaidų taisymo galimybes.

6 LITERATŪRA

- [1] Chang, T. H., Yeh, T., Miller, R. C., „GUI Testing Using Computer Vision“. [interaktyvus] [žiūrėta 2011-09-22]. Prieiga per internetą: <http://groups.csail.mit.edu/uid/projects/sikuli/sikuli-chi2010.pdf>
- [2] Chong, W. H. „RPB in Software Testing“, pp.8, International Multi-Conference on Computing in the Global Information Technology (ICCGI'07), 2007
- [3] Garcia, J. E. Automated Model-based GUI Test Cases Generation from Use Cases. 2008. [interaktyvus] [žiūrėta 2011-09-22]. Prieiga per internetą: http://paginas.fe.up.pt/~aas/pub/Aulas/PI/Avaliacao/Ano07_08/JorgeEsparteiro.pdf
- [4] Gerrard, P., „Testing GUI Application“. 1997. [interaktyvus] [žiūrėta 2011-10-28]. Prieiga per internetą: <http://www.gerrardconsulting.com/sites/default/files/Techgui.pdf>
- [5] Hetzel, W. C., The Complete Guide to Software Testing, 2nd ed. Wellesley, Mass. : QED Information Sciences, 1988. ISBN: 0894352423.
- [6] International Organization for Standardization, „ISO Standard 9126: Software Engineering –Product Quality, parts 1, 2 and 3“, International Organization for Standardization, Geneva, 2001 (part 1), 2003 (parts 2 and 3).
- [7] ISO/IEC, 9241-11 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability. 1998:ISO/IEC 9241-11: 1998 (E).
- [8] Johnson, J., „Designing with the mind in mind: simple guide to understanding user interface design rules“. Morgan Kaufmann, 2010.
- [9] Johnson, J., „GUI Bloopers 2.0, Second Edition: Common User Interface Design Don'ts and Dos. Morgan Kaufman, 2007.
- [10] Memon, A. M., „GUI Testing: Pitfalls and Process, Software Technologies“, 35(8):87-88. 2002.
- [11] Moroz-Lapin, K., „Žmogaus ir kompiuterio sąveika. Metodinė priemonė programų sistemų ir informatikos specialybių bakalauro studijų studentams“, 2004. [interaktyvus] [žiūrėta 2011-12-03] Prieiga per internetą: <http://e-stud.vgtu.lt/users/files/dest/11242/hci.pdf>
- [12] Nielsen J. Nielsen Norman Group. Jakob Nielsen's Alertbox. Usability 101: Introduction to Usability. 1995-2011. ISSN 1548-5552 [interaktyvus] [žiūrėta 2011-12-02]. Prieiga per internetą: <http://www.useit.com/alertbox/20030825.html>

- [13] Norman, K. L. „Cyberpsychology: An Introduction to Human-Computer Interaction“. Cambridge University press, 2008.
- [14] Tucker, I. „Testing Graphical User Interfaces“. 2007 [interaktyvus] [žiūrėta 2011-09-22]. Prieiga per internetą: <http://www.cs.swan.ac.uk/~csmarkus/CS339/dissertations/TuckerI.pdf>
- [15] GUI Types. [interaktyvus] [žiūrėta 2011-12-20]. Prieiga per internetą: <http://www.scribd.com/doc/13887556/GUI-Types>
- [16] Cunningham & Cunningham, Inc. [interaktyvus] [žiūrėta 2011-12-20]. Prieiga per internetą: <http://www.c2.com/cgi/wiki?FunctionalTest>
- [17] Jorge Esparteiro Garcia. „Automated Model-based GUI Test Cases Generation from Use Case“. [interaktyvus] [žiūrėta 2011-12-20]. Prieiga per internetą: <http://paginas.fe.up.pt/~apaiva/PhD/PhDGUITesting.pdf> (40 psl.)
- [18] Nyman, N. „Using Monkey Test Tools, in STQE“ – Software Testing and Quality Engineering Magazine, 2000.
- [19] White, L., Almezen, H. „Generating test cases for GUI responsibilities using complete interaction sequences“. 11th International Symposium on Software reliability engineering, proceedings, pp. 110-121. 2000.
- [20] Rubin, J., „Chisnell, D. Handbook of Usability Testing. How to plan, design and conduct effective tests: Second Edition“. Wiley Publishing, Inc. 2008.
- [21] Johnson, J. „Web Bloopers: 60 Common Web Design Mistakes, and How to Avoid Them“. Morgan Kaufman, 2003.
- [22] Atlassian. JIRA. [interaktyvus] [žiūrėta 2012-02-28]. Prieiga per internetą: www.atlassian.com/JIRA
- [23] ISO/IEC 29119 Software Testing. [interaktyvus] [žiūrėta 2012-05-20]. Prieiga per internetą: <http://www.softwaretestingstandard.org/>
- [24] Software Testing fundamentals. [interaktyvus] [žiūrėta 2012-05-20]. Prieiga per internetą: <http://softwaretestingfundamentals.com/test-plan/>
- [25] W. O. Galitz, „The Essential Guide to User Interface Design“. Third Edition. Wiley Publishing, 2007.
- [26] Lewis, P. Polson, C. Wharton, J. Rieman. „Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces“, CHI '90 Proceedings of the SIGCHI Conference on Human Factors on Computing Systems, ACM New York, pp. 235-242, 1990.

[27] J. Bach, „Heuristic Risk-Based Testing“, Software Testing and Quality Engineering Magazine, 11/1999, prieiga per internetą: <http://nilachakra.org/documents/material/L%20-%20RiskAnalysis.pdf>

[28] P. Rook, „Controlling software project“, IEEE Software Engineer Journal, 1986, pp. 7-16

[29] G. Mujtaba, T. Mahmood, Z. Nasir, „A Holistic Approach to Software Defect Analysis and Management“, Australian Journal of Basic and Applied Sciences, 5(6), pp. 1632-1640, 2011.

[30] E. Šarapnickytė, K. Kapočius, „Saityno informacinių sistemų vartotojo sąsajos testavimo metodika“, Proceedings of the 18th Master and PhD students conference “Information Society and University studies” (IVUS 2013), pp. 147-152, 2013.

7 PRIEDAI

7.1 priedas. Straipsnis

Saityno informacinių sistemų vartotojo sąsajos testavimo metodika

Edita Šarapnickytė
Informacijos sistemų katedra
Informatikos fakultetas
Kauno technologijos universitetas
Kaunas, Lietuva
El. paštas: edita.sarapnickyte@stud.ktu.lt

Kęstutis Kapočius
¹Informacijos sistemų katedra
²Informacinių sistemų projektavimo technologijų centras
Informatikos fakultetas
Kauno technologijos universitetas
Kaunas, Lietuva
El. paštas: kestutis.kapocius@ktu.lt

Santrauka. Svarbus veiksnys, lemiantis vartotojų požiūrį į sistemą ir jos efektyvumą – tai aukštas sistemos panaudojamumas, kurį ženkliai įtakoja vartotojo sąsajos charakteristikos. Dėl milžiniškos konkurencijos pranašumas panaudojamumo srityje ypač aktualus saityne, todėl akivaizdu, jog saityno informacinių sistemų sąsają tikslinga nuodugnai testuoti. Vis tik atlikus žinomų testavimo būdų analizę nustatyta, jog jų pagalba saityno sistemų sąsajos testavimui keliami reikalavimai gali būti tenkinami tik dalinai. Sprendžiant šią problemą buvo sukurtas specializuotas, būtent saityno IS vartotojo sąsajos testavimui pritaikytas sprendimas, apjungiantis ir pagal poreikį praplečiantis egzistuojančius metodus. Atlikta preliminarini sukurtoji metodikos taikymo galimybių analizė leidžia daryti prielaidą, jog metodika turėtų leisti ne tik aptikti daugiau ir esminių vartotojo sąsajos klaidų, bet ir operatyviai priimti sprendimus dėl aptiktų defektų šalinimo.

Reikšminiai žodžiai: testavimas, grafinė vartotojo sąsaja, informacinės sistemos.

I. ĮVADAS

Augančią įtaką organizacijų veiklos efektyvumui ir populiarumui darančių saityno informacinių sistemų (IS) efektyvumas ir kokybė tampa vis labiau aktualūs ir vertinami ne tik informacinių sistemų savininkų, jų kūrėjų, bet ir eilinių vartotojų. Nesunku nuspėti, jog tai lemia milžiniška saityne egzistuojanti konkurencija ir galimybė vienu pelės paspaudimu iš nepatikusios svetainės persikelti į patrauklesnę konkurentų sistemą. Ypač svarbus veiksnys, įtakojantis vartotojų požiūrį į sistemą ir jos efektyvumą – tai aukštas sistemos panaudojamumas, kurį ženkliai įtakoja (grafinės) vartotojo sąsajos (GVS) charakteristikos. Žemas panaudojamumas gali lemti tai, jog sistema bus tiesiog ignoruojama vartotojų, todėl efektyvus saityno informacinių sistemų vartotojo sąsajos testavimas yra ne mažiau svarbus nei funkcinis šių sistemų testavimas.

Testuojant tokio tipo informacines sistemas, taikyti tik funkcinio testavimo metodikas negana, tačiau žinomi tradicinės programinės įrangos vartotojo sąsajos testavimo būdai nėra visiškai tinkami IS sąsajos testavimui [2]. Saityno informacinių sistemų testavimas apskritai traktuotinas kaip besiformuojanti,

neišplėta informatikos sritis. Nors saityno IS vartotojo sąsajos testavimui galima pasirinkti šiuo metu žinomus vartotojo sąsajos testavimo ir vertinimo metodus, atlikta analizė atskleidė, kad šiam uždaviniui taikant tradicinius testavimo būdus testavimo procesui keliami reikalavimai bus tenkinami tik dalinai. Atkreiptinas dėmesys į tai, jog saityno svetainių ir informacinių sistemų sąsaja pasižymi eile darbo būdo (saityne *naršoma*) ir techninės aplinkos lemiamų ypatybių [25]. Dėl to formuojasi specifinė, tik tokio pobūdžio sistemoms būdingų sąsajos defektų ir galimų jų sprendimo būdų aibė [7] [8].

Sukūrus specializuotas, saityno informacinių sistemų vartotojo sąsajos testavimui pritaikytas priemones, pavyktų užtikrinti aukštą produkto kokybę, išvengiant dažniausiai pasitaikančių vartotojo sąsajos klaidų ir trūkumų. Straipsnyje pristatoma būtent tokia metodika, skirta saityno informacinių sistemų vartotojo sąsajos testavimui. Metodika grindžiama standartiniu testavimo procesu, paremta rankinio testavimo būdu ir euristinio vertinimo metodu. Vadovaujantis sukurta metodika ne tik galima tikėtis aptikti daugiau ir esminių vartotojo sąsajos klaidų, bet ir gauti parengtus defektų šalinimo sprendimus, kurie gali būti perpanaudoti arba papildyti.

Antroje straipsnio dalyje apžvelgti susiję sprendimai, aptarti testavimo tipų, metodų privalumai ir trūkumai. Čia taip pat pateikiama testavimo procesų bei priemonių, kurias galima taikyti testavimo metu, apžvalga. Trečia straipsnio dalis skirta siūlomai saityno informacinės sistemos grafinės vartotojo sąsajos testavimo metodikai pristatyti. Ketvirtoje straipsnio dalyje pateikiamos metodikos realizavimo bei taikymo galimybės. Straipsnis baigiamas išvadomis.

II. ŽINOMŲ VARTOTOJO SĄSAJOS TESTAVIMO SPRENDIMŲ ĮVERTINIMAS

Testavimas – tai bet kokia veikla, kurios tikslas yra įvertinti programinės sistemos charakteristikas ar galimybes bei nustatyti, ar ji tinkama reikalaujamiems rezultatams gauti [5]. Tarptautinio programinės įrangos testavimo standarto apibrėžtame testavimo proceso modelyje vienas iš esminių testavimo proceso lygių yra dinaminiai testavimo procesai [5].

Dinaminių testavimo procesų lygis yra sudėtinis ir jį sudaro keturi žemesnio lygio procesai:

- Testo kūrimas.
- Testavimo aplinkos parinkimas ir konfigūravimas
- Testo vykdymas.
- Testavimo rezultatų fiksavimas.

Šių sudėtinių testavimo procesų vykdymas ir veiklos sudėtis priklauso nuo pasirinktos testavimo technikos, metodų ir priemonių, kurios gali būti parenkamos atsižvelgiant į testuojamą objektą ir siekiamus rezultatus.

Sistemų vartotojo sąsajos testavimui reikalinga unikali metodika, tačiau šios metodikos įgyvendinimas turi būti paremtas tradicinės programinės įrangos funkcinio testavimo eiga [2]. Galima daryti prielaidą, jog remiantis standartiniu programinės įrangos testavimo proceso modeliu, gali būti atliktas ir saityno informacinės sistemos tiek funkcinis, tiek vartotojo sąsajos ir panaudojamumo testavimas, tačiau būtina šį procesą pritaikyti ir detaliau apibrėžti. Specializavus procesą saityno informacinės sistemos grafinės vartotojo sąsajos testavimui, bus sukurtos prielaidos tokio tipo sistemų testavimo efektyvumui (rastų defektų skaičius/esamų defektų skaičius) ir produktyvumui (defektų skaičiaus radimas per kuo trumpesnę laiką) padidinti.

Specializuojant testavimo procesą, gali būti parenkamos skirtingos testavimo priemonės, būdai ir technikos, atsižvelgiant į tai, ką siekiama pasiekti testavimu. Taigi, būtina gerai suprasti prieinamus sprendimus ir jų privalumus bei trūkumus. Egzistuojančias programinės įrangos testavimo metodikas pagal testavimo vykdymo būdą galima suskirstyti į dvi pagrindines grupes:

1) *Rankinis testavimas*. Seniausiai taikomas, labiau tradicinis ir daugiausiai nuo testuotojo patirties ir asmeninių savybių priklausantis testavimo būdas [1] [10]. Rankiniu būdu dažniausiai testuojama siekiant aptikti sistemos vartojimo patogumo klaidas, nes pagrindinės ir kritiškiausios vartotojo sąsajos klaidos randamos, tik kuomet testavime dalyvauja realūs žmonės.

2) *Automatinis testavimas*. Testavimo būdas, kai testavimo atvejams vykdyti naudojamos programos bei įrankiai, skirti programų ar modulių, kurių kokybę norima įvertinti, darbui patikrinti [10].

Siekiant palyginti ir įvertinti, kuris metodas gali būti tinkamesnis saityno informacinės sistemos grafinėi vartotojo sąsajai testuoti, galima pasinaudoti Smilgytės ir Nenortaitės (2011) atlikta analize [9]. Autorės atliko rankinio ir automatinio testavimo būdų privalumų ir trūkumų palyginimą pagal tokius pagrindinius kriterijus: kainos, testų parengimo ir vykdymo laiko, pritaikymo prie besikeičiančių reikalavimų, išmokstamumo, testavimo aprėpties ir klaidų aptikimo.

Nors rankinių testavimo metodų taikymas grafinės vartotojo sąsajos testavimui yra lėtas, testavimo atvejų padengiamumas gali būti didesnis, nei taikant automatinius testus. Automatiniai testai labiau tinkami testuoti paviršutiniškai, o ne identifikuoti kritines ir specifines klaidas [1]. Automatinis testavimas negali pakeisti rankinio testavimo

metu taikomo analitinio mastymo. Taip pat svarbu įvertinti ir tai, kad atlikus automatinių testavimą vis vien būtina analizė ir sprendimų priėmimas, kuris testuojant rankiniu būdu yra atliekamas iškart. Taigi, tikslinga testavimo etape rinktis rankinius testus, kartais kombinuojant juos su automatiniais, nereikalaujančiais didelio įdirbio testais.

Automatiniai testai padeda aptikti ir eliminuoti tik esmines klaidas, pasitaikančias kūrimo etape. Pagrindinės ir kritiškiausios grafinės vartotojo sąsajos klaidos randamos testuojant rankiniu būdu. Rankinio testavimo metodus galima taikyti jau ankstyvose sistemos kūrimo stadijose, kai vartotojo sąsaja dar nėra pakankamai stabili, automatiniai testai dar nėra veiksnūs ir jų vykdymas neįmanomas arba neturi prasmės [26][10]. Nereikia pamiršti, jog defektų aptikimas ankstyvose kūrimo stadijose padeda sumažinti jų sprendimo kaštus [29].

Analizuojant automatinių testavimo metodų greitį, būtina pažymėti, jog šis pranašumas prieš rankinius metodus išryškėja tik testų vykdymo etape. Testavimo automatizavimas leidžia ženkliai padidinti testų skaičių panaudojant testus keletą kartų, vykdant juos reguliariai. Tai nereikalauja papildomų resursų, tačiau įvertinant reikalavimų pasikeitimus testus kaskart tenka koreguoti, o jų parengimas imlus laikui. Papildomas pasiruošimas ir žinios reikalingos ne tik pirmą kartą kuriant testus, bet ir nuolat juos atnaujinant ir tobulinant. Automatiinių testų kūrimui būtinos išskirtinės programavimo ir modeliavimo žinios [10], todėl tokio tipo metodų taikymas neretai pareikalauja papildomų išteklių ir gali sukelti diskomfortą testavimo komandai.

Testavimo įrankiai, naudojami įvertinti sistemos panaudojamumą skirtingais būdais, pagal taikymo principus gali būti suskirstyti į keturias grupes [25]:

- Stebėjimo.
- Palyginimo.
- Klausymo.
- Įvertinimo.

Atsižvelgiant į tai, kad defektų aptikimas ir jų pataisymas ankstyvose kūrimo stadijose padeda sumažinti sistemos kūrimo išlaidas, galima išskirti tokius pagrindinius sąsajos testavimo/įvertinimo metodus, taikytinus nuo ankstyvos testavimo stadijos [25]:

- Pažintinė peržvalga.
- Atitikimo rekomendacijoms ir standartams vertinimas (formalus).
- Euristinis vertinimas.

Šie metodai nereikalauja pašalinių vertintojų dalyvavimo ir gali būti atlikti pačių kūrėjų (ar kuriančios kompanijos testuotojų) rankiniu būdu. Tai lemia sąlyginai neaukštą tokio testavimo kainą.

Pažintinė peržvalga – tai vartotojo sąsajos įvertinimo metodas, kurio centre yra reikalavimų specifikuojimo metu apibrėžtų vartotojų užduočių vykdymas [25]. Šis būdas leidžia kuo ankščiau gauti dizaino sprendimo įvertinimus [26] ir jau projektavimo etape įvertinti specifikuotas užduočių vykdymo sekas. Kuo ankščiau dizaino klaidos aptinkamos, tuo lengviau jas spręsti. Vis tik taikant metodą ankstyvose sistemos kūrimo

stadijose, naudojant grafinius ar programinius sąsajos prototipus, neatsižvelgiama į galimus interaktyvius vartotojo veiksmus [26]. Todėl iškyla poreikis realizuoti interaktyvių procesų imitaciją, įtraukiant ir specifines užduotis. Toks pasiruošimas testavimui ir testavimas vadovaujantis formaliu metodu aprašu reikalauja intensyvaus darbo. Tuo tarpu taikant tik supaprastintas priemones gaunama paviršutiniška ir siaura analizė, pagrinde apimanti tik teksto ir grafinio dizaino, sąsajos elementų išdėstymo problemas.

Atitikimo rekomendacijoms ir standartinio vertinimo metodas – tai sąsajos peržiūra, kuri atliekama vadovaujantis iš anksto griežtai apibrėžtomis rekomendacijomis. Rekomendacijose labai tiksliai nurodomi reikalavimai arba taisyklės, kurių reikėtų laikytis kuriant konkrečius sąsajos elementus [25]. Vadovaujantis šiomis rekomendacijomis, aptinkamos bendrinės ir dažniausiai pasikartojančios klaidos, taip pat įvertinami ne tik teksto ir grafinių vaizdų defektai, bet ir dizaino ar pačių objektų išdėstymo problemos. Defektų šalinimo sprendimų priėmimas taip pat vykdomas vadovaujantis rekomendacijomis. Vis tik ne visos situacijos gali būti numatytos ar apibrėžtos rekomendacijose, tarptautiniuose arba įmonių naudojamuose standartų rinkiniuose.

Euristinis vertinimas – tai kruopštus sistemos vertinimas, atliekamas vartotojo sąsajos specialistų ar kvalifikuotų IS kūrimo komandos narių [25]. Taikant šį metodą vertinama pagal asmeninę patirtį, bet dažniau pagal tai, kaip sistema atitinka žinomus panaudojamumo principų rinkinius – euristikas. Euristikos dažniausiai pateikiamos kaip atvirų kontrolinių klausimų ar pasiūlymų sąrašas. Šis sąrašas nėra veiksmų, kuriuos turi atlikti testuotojas, sąrašas; jis naudojamas ne kontroliuoti atliekamus veiksmus ir leidžia įvertinti daugiau galimų ir įdomių problemų aspektų [27]. Tokiu būdu gali būti aptinkama daug įvairaus tipo defektų, įskaitant ir panaudojamumo problemas, tačiau neretai gali kilti problemų aptinkant globalias sistemos problemas ir trūkstamus sąsajos elementus. Vadovaujantis euristikomis ne visada lengva spręsti iškilusias problemas, todėl problemų sprendimų paieška neretai tampa dideliu kūrybiniu iššūkiu testavimo, analitikų ar programuotojų komandai.

Apibendrinti atliktos testavimo metodų ypatumų analizės rezultatai pateikiami 1 lentelėje.

Remiantis atlikta analize galima daryti prielaidą, jog egzistuojantys metodai netenkina visų saityno informacinės sistemos grafinės vartotojo sąsajos testavimo procesui keliamų reikalavimų. Idealiu atveju testavimo metodas turėtų ne tik leisti aptikti esmines saityno informacinės sistemos grafinės vartotojo sąsajos defektus, bet ir padėti juos efektyviai šalinti. Tokį sprendimą galima tikėtis sukurti apjungus ir pagal poreikį praplėtus egzistuojančius metodus.

LENTELĖ I. TESTAVIMO METODŲ PALYGINIMAS

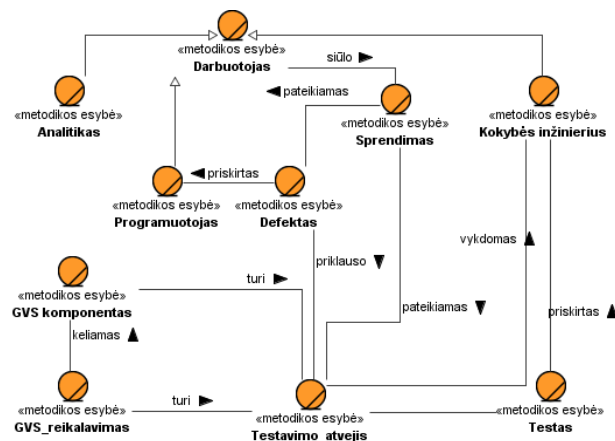
	Pažintinė peržvalga	Atitikimo rekomendacijoms ir standartinio vertinimo	Euristinis vertinimas
Testavimo atvejai	Užduotys, kurias reikia atlikti	Reikalavimų pagrindu sukurtos rekomendacijos arba taisyklės	Euristikos pagal žinomus panaudojamumo principus

	Pažintinė peržvalga	Atitikimo rekomendacijoms ir standartinio vertinimo	Euristinis vertinimas
Neaptinkamos klaidos	Bendros ir dažnos klaidos	Rimtos koncepcinės, navigacijos ir operacinės klaidos	Globalios problemos ir trūkstami sąsajos elementai
Aptinkamos	Išskirtos problemos gali būti ne esminės	Bendrinės ir pasikartojančios objektų išdėstymo ir dizaino klaidos	Daug įvairių, įskaitant ir panaudojamumo klaidas
Problemų sprendimai	Nėra (sprendimus priima projektuotojai)	Yra (vadovaujamasi priimtomis rekomendacijomis ir standartais)	Ne visada apibrėžtas būdas problemų sprendimams gauti

III. SIŪLOMA SAITYNO IS VARTOTOJO SĄSAJOS TESTAVIMO METODIKA

Kuriant čia pristatomą metodiką buvo remiamasi tarptautinio programinės įrangos testavimo standarto apibrėžtu testavimo proceso modeliu, jį detalizuojant ir pritaikant pasirinktam testavimo tipui ir metodui. Kaip jau buvo minėta, šio tyrimo metu buvo siekiama sukurti sprendimą, kuris būtų specializuotas būtent saityno informacinių sistemų (grafinės) vartotojo sąsajos testavimui. Pasirinktas rankinio testavimo principas, kuriam būdingas didelės testavimo laiko sąnaudas buvo siekiama sumažinti. Metodika kurta euristinio vertinimo metodo pagrindu, tačiau sprendžiant šiam metodui būdingas trūkstamų sąsajos elementų nepastebimumo ir sprendimų priėmimo problemas.

Sudarant teorinį testavimo metodikos modelį (Pav. 1) išskirti pagrindiniai konceptai ir jų tarpusavio ryšiai saityno informacinės sistemos grafinės vartotojo sąsajos testavimo metodikos rėmuose.



Pav. 1. Konceptinis testavimo metodikos modelis

Atlikus testavimo procese dalyvaujančių veikėjų analizę išskirti programuotojo, analitiko ir kokybės inžinieriaus konceptai. Šių konceptų ryšys su kitais konceptais identifiukuotas pagal testavimo procese dalyvaujančių veikėjų atsakomybių sritis ir jų vykdomas užduotis.

Vienas iš dažniausiai nurodomų euristinio vertinimo metodo trūkumų yra defektų šalinimo sprendimų būdų pateikimo nebuvimas. Čia pristatomoje metodikoje numatoma,

kad kuriamas testavimo atvejis (euristika, TA) arba testavimo atvejo defektas (TD) gali turėti sprendimą. Kiekvieną kartą perpanaudojant universalias euristikas skirtingoms vartotojo sąsajoms testuoti, aptikus defektą galima pritaikyti anksčiau taikytą konkretaus defekto šalinimo sprendimą, jį modifikuoti arba pridėti naują.

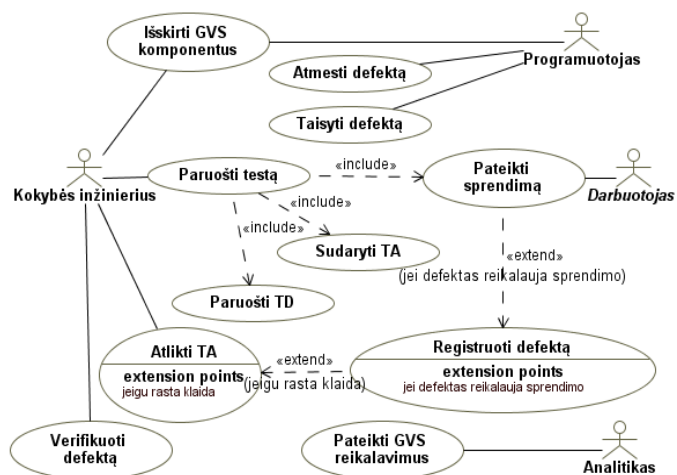
Kitas euristinio metodo trūkumas – testuojant nepastebimi trūkstami vartotojo sąsajos elementai. Siūloma metodika paremta grafinės vartotojo sąsajos išskaidymu į jos komponentus. Sudaromus testavimo atvejus (euristikas) siūloma susieti su vartotojo sąsajos komponentais. Tokio ryšio užmezgimas leidžia ne tik atlikti kiekvieno komponento testavimą, bet ir kaupti jiems pritaikytus testavimo atvejus.

Metodikos taikymas pavaizduotas panaudojimo atvejų modeliu (Pav. 2), kuriame atspindėti tipiniai metodikos vartotojai ir metodikos panaudojimas metodikos rėmuose.

Remiantis tarptautinio programinės įrangos testavimo standarto „ISO/IEC 29199 Software testing“ [5] siūlomu testavimo proceso modeliu ir išlaikant jo struktūrą, išskirti panaudojimo atvejai buvo suskirstyti į grafinės vartotojo sąsajos testavimo metodikos etapus (Pav. 3). Be to, procesai detalizuoti, išskaidant juos į smulkesnius procesus ir pritaikant saityno informacinių sistemų vartotojo sąsajos testavimui.

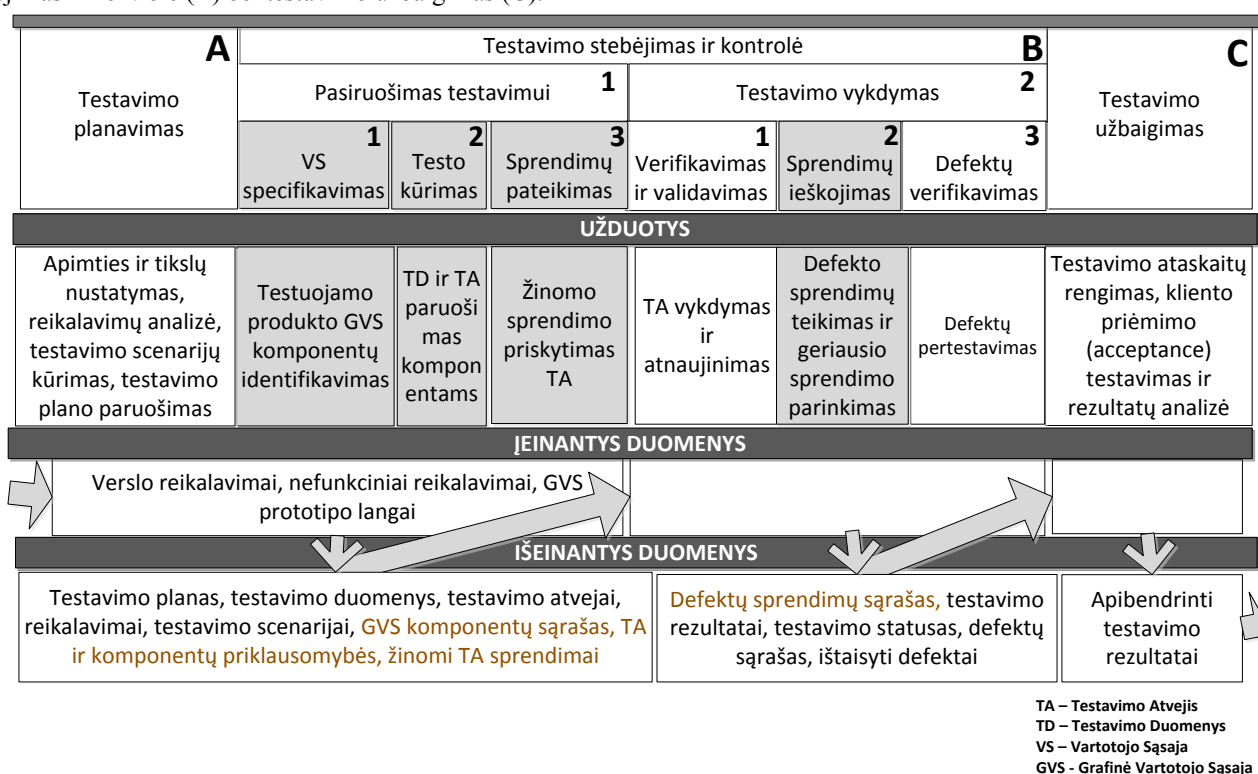
Sudarytas vartotojo sąsajos testavimo metodikos etapų modelis, lyginant su tipiniu programinės įrangos testavimo procesų modeliu, išlaiko nepakitusių testavimo valdymo procesų lygis, t.y. tokį, koks yra apibrėžtas tarptautiniame programinės įrangos testavimo standarte. Testavimo valdymo

procesų lygį sudaro testavimo planavimas (A), testavimo stebėjimas ir kontrolė (B) bei testavimo užbaigimas (C).



Pav. 2. Metodikos taikymo veiklos

Dinaminis testavimo proceso lygis, kurį standartiškai sudaro testo kūrimo (1.2.), testo vykdymo (2.1) ir defektų registravimo etapai (2.3), metodikoje papildytas trimis saityno informacinės sistemos grafinės vartotojo sąsajos testavimui pritaikytais, specializuotais etapais (3 paveikslėlyje pažymėti pilka spalva). Tai vartotojo sąsajos analizė (1.1.), sprendimų pateikimo (1.3.) ir sprendimų paieškos (2.2.) etapai.



Pav. 3. Vartotojo sąsajos testavimo metodikos etapai (pilkiesni elementai – naujai įvesti etapai, pritaikyti vartotojo sąsajos testavimui)

Taigi, siūlomas vartotojo sąsajos testavimo procesas susideda iš keturių etapų:

A. *Testavimo planavimas*

B. *Pasiruošimas testavimui*

- Vartotojo sąsajos analizė.
- Testų kūrimas.
- Sprendimų pateikimas.

C. *Testavimo vykdymas*

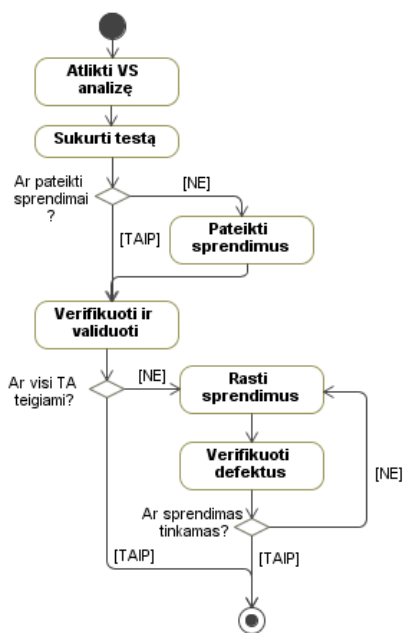
- Validavimas ir verifikavimas.
- Sprendimų ieškojimas.
- Defektų verifikavimas.

D. *Testavimo užbaigimas (schemoje nerodomas)*

Dinaminiai sudarytosios metodikos etapai – tai pasiruošimas testavimui ir testavimo vykdymas. Šie etapai buvo pritaikyti grafinės vartotojo sąsajos testavimui.

Pasiruošimo ir testavimo vykdymo sub-etapų tvarka nėra griežtai apibrėžta. Etapai gali būti persidengiantys, vykdomi kelis kartus pakartotinai pagrindinio etapo ribose arba būtinybei, praleidžiami. Pasiruošimo testavimui sub-etapai gali būti vykdomi lygiagrečiai, tačiau tikslingiausia būtų juos perdengti – pirmiausiai pradėti nuo vartotojo sąsajos analizės, o vykdant testų kūrimo etapą, tuo pat metu pateikti ir žinomus sprendimus.

Taigi, apibendrintas saityno IS vartotojo sąsajos testavimo procesas, nevertinant planavimo ir užbaigimo, susideda iš šešių žingsnių (žr. pav. 4). Kaip matome, identifikuoti trys sprendimo taškai – ar pateikti sprendimai, ar visi TA teigiami ir ar sprendimas tinkamas. Šie sprendimo taškai lemia ar visi grafinės vartotojo sąsajos testavimo metodikos etapai/procesai bus vykdomi ir kiek kartų jie bus vykdomi.

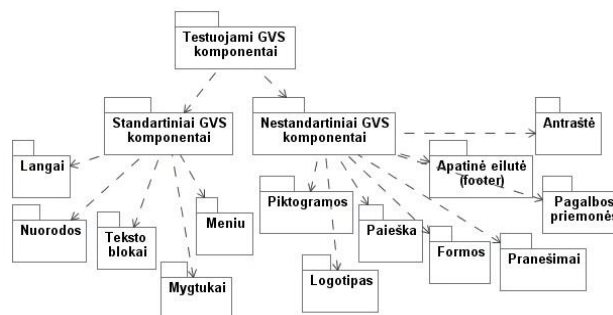


Pav. 4. Bendrinis testavimo metodikos procesas

IV. METODIKOS REALIZAVIMO IR TAIKYMO GALIMYBĖS

Vienas iš pirmųjų ir svarbiausių pasiruošimo etapų, numatytų sudarytoje saityno grafinės vartotojo sąsajos testavimo metodikoje, yra vartotojo sąsajos analizės etapas. Šio etapo metu pagal turimus dizaino šablonus ar prototipus yra sudaroma saityno grafinės vartotojo sąsajos komponentų aibė. Pagal sudarytą komponentų aibę yra nusprendžiama, kokie testavimo atvejai bus reikalingi testuojamos saityno informacinės sistemos grafinėi vartotojo sąsajai.

Saityno informacinės sistemos skiriasi savo struktūra ir turiniu, tačiau pasitaiko standartinių, daugelio saityno IS vartotojo sąsajai būdingų komponentų. Tyrimo metu rekomendacinis universalus standartinių komponentų sąrašas (žr. pav. 5) sudarytas atsižvelgiant į dažniausiai pasitaikančių saityno informacinių sistemų vartotojo sąsajos klaidų grupes, apibrėžtas J. Johnson [7] [8]. Klaidų grupės turi joms būdingą klaidų pasitaikymo vietą saityno informacinėse sistemose, kurios gali būti traktuojamos kaip komponentai.



Pav. 5. Rekomenduojama komponentų aibė

Kita stambi komponentų grupė yra nestandartiniai grafinės vartotojo sąsajos komponentai. Tokie komponentai gali būti būdingi ne visų saityno informacinių sistemų grafinėi vartotojo sąsajai. Siekiant sudaryti tokį specifinių komponentų sąrašą yra tikslinga pasirinkti tikslinį metodikos sprendimo poaibį.

Rekomenduojamas standartinių ir nestandartinių komponentų rinkinys (žr. pav. 5) gali būti panaudotas kaip aibė, su kuria galima dirbti pagal sukurtą metodiką.

Išskirtiems grafinės vartotojo sąsajos komponentams turi būti sudaryti ir priskirti testavimo atvejai (euristikos). Statiniai (universalūs) testavimo atvejai turi būti įtraukti į duomenų bazę kaip tipiniai testavimo atvejai, tinkami visoms sistemoms. Prasidėjus naujam projektui, jeigu reikalinga, testuojamam produktui būdingi dinaminiai testavimo atvejai gali būti numatomi ir įtraukiami testų kūrimo etape.

V. IŠVADOS

Atlikta vartotojo sąsajos testavimo ir vertinimo metodų analizė parodė, kad šiuo metu dominuojantys metodai nėra pakankamai tinkami saityno informacinių sistemų vartotojo sąsajai testuoti. Vis tik žinomos metodikos pasižymi tam tikrais privalumais, kuriais verta pasinaudoti. Šio tyrimo metu buvo sukurta metodika, kurioje remiamasi testavimo standartais, panaudojamas rankinio testavimo principas bei adaptuotas euristinio vertinimo metodas.

Preliminarus metodikos tyrimas parodė, jog sprendimas gali būti realizuotas ir turėtų leisti pagerinti ne tik sąsajos defektų aptikimo efektyvumą, bet ir jų sprendimo greitį, pritaikytų pataisymų kokybę. Vienas svarbiausių sukurtosios metodikos ypatumų – galimybė kaupti žinias ne tik apie galimas sąsajos IS klaidas, bet ir apie šių klaidų taisymo galimybes.

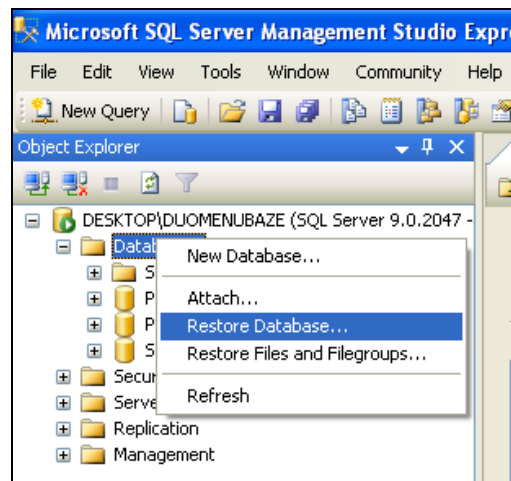
Artimiausiuose planuose numatoma sukurti metodika grindžiamą testavimo sistemos prototipą. Sukurtąjį prototipą bus siekiama panaudoti metodikos tyrimui realiuose projektuose, o gauti rezultatai bus panaudoti tolimesniam metodikos vystymui.

LITERATŪRA

- [1] Leitner, I. Ciupa, B. Meyer, and M. Howard, "Reconciling manual and automated testing: The autotest experience," Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40 2007), pp. 261-261a, 2007.
- [2] M. Memon, "GUI Testing: Pitfalls and Process," Computer, vol. 35, no. 8, pp. 87-88, 2002.
- [3] Lewis, P. Polson, C. Wharton, and J. Rieman. "Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces," in CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM New York, pp. 235-242, 1990.
- [4] G. Mujtaba, T. Mahmood, and Z. Nasir, "A Holistic Approach to Software Defect Analysis and Management," Australian Journal of Basic and Applied Sciences, 5(6), pp. 1632-1640, 2011.
- [5] ISO/IEC JTC1/SC7, ISO/IEC 29119 Software Testing Standard, prieiga per internetą: <http://www.softwaretestingstandard.org/>
- [6] J. Bach, "Heuristic Risk-Based Testing," Software Testing and Quality Engineering Magazine, 11/1999, prieiga per internetą: <http://nilachakra.org/documents/material/L%20-%20RiskAnalysis.pdf>
- [7] J. Johnson, Web Bloopers: 60 Common Web Design Mistakes, and How to Avoid Them. Morgan Kaufman, 2003.
- [8] J. Johnson, GUI Bloopers 2.0, Second Edition: Common User Interface Design Don'ts and Dos. Morgan Kaufman, 2007.
- [9] K. Smilgytė, J. Nenortaitė, "Dirbtiniais neuroniniais tinklais paremtas testavimo tipo pasirinkimo metodas," Proceedings of the 15th Master and PhD conference "Information Society and University studies" (IVUS 2010), pp. 111-116, 2011.
- [10] OTS Solutions, "Manual testing v/s automated testing," prieiga per internetą: <http://www.slideshare.net/Softwarecentral/manual-testing-vs-automated-testing>
- [11] W. C. Hetzel, The Complete Guide to Software Testing, 2nd ed. QED Information Sciences, 1988.
- [12] W. O. Galitz, The Essential Guide to User Interface Design. Third Edition". Wiley Publishing, 2007.

7.2 priedas. Sukurto sprendimo prototipo vartotojo vadovas Sistemos paleidimo instrukcija

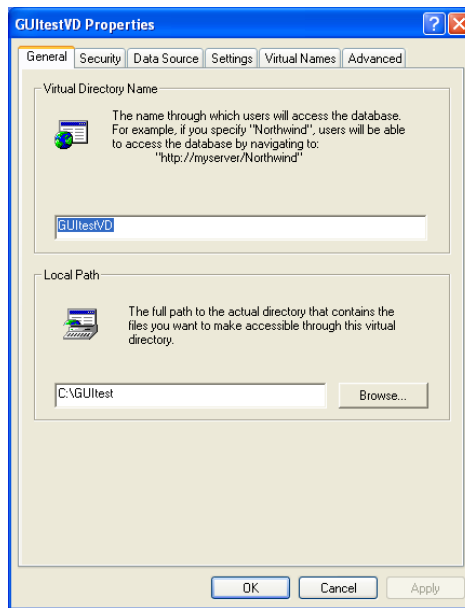
Sistema pateikta dviejuose failuose. Pirmame *.rar formato archyve yra stiliaus bei XML ir XSLT failai. Antrasis failas yra duomenų bazės atsarginė kopija. Pirmiausiai reikia atsikurti duomenų bazę iš jos kopijos. Tai atliekama SQL Server Management Studio dešiniuoju pelės klavišu spaudžiant Object Explorer dalyje ant Databases aplanko ir pasirenkant Restore Database, taip kaip iliustruojama 1 paveikslėlyje.



1 pav. Atkūrimo duomenų bazės iš kopijos pasirinkimas

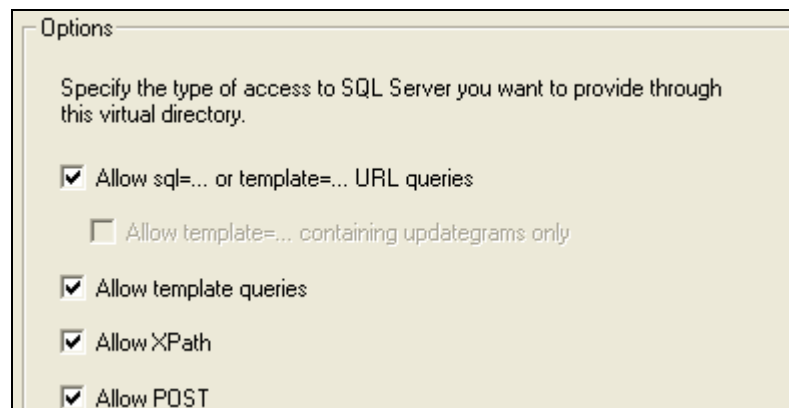
Dialogo lange Restore Database pasirenkame, kad atkurtų iš failo duomenų bazę. Tam reikia pažymėti pasirinkimą From device, bei nurodyti kelią iki kopijos failo. Atsiradusiame dar viename lange paspaudžiame mygtuką Add ir surandame kopijos failą. Paspaudžiame du kartus OK mygtuką. Nepamirštame uždėti Restore varnelės bei parašyti duomenų bazės pavadinimo ir tada paspaudžiame OK. Dialogo langas su užpildyta reikiama informacija parodytas toliau.

Tolesnis žingsnis yra įtraukti puslapį per IIS Virtual Directory Management for SQLXML programą. Joje sukuriame naują direktoriją vardu GUItestVD, o kelias iki jos tai aplankas GUItest. Kaip tai turi atrodyti matoma sekančiam paveiksle.



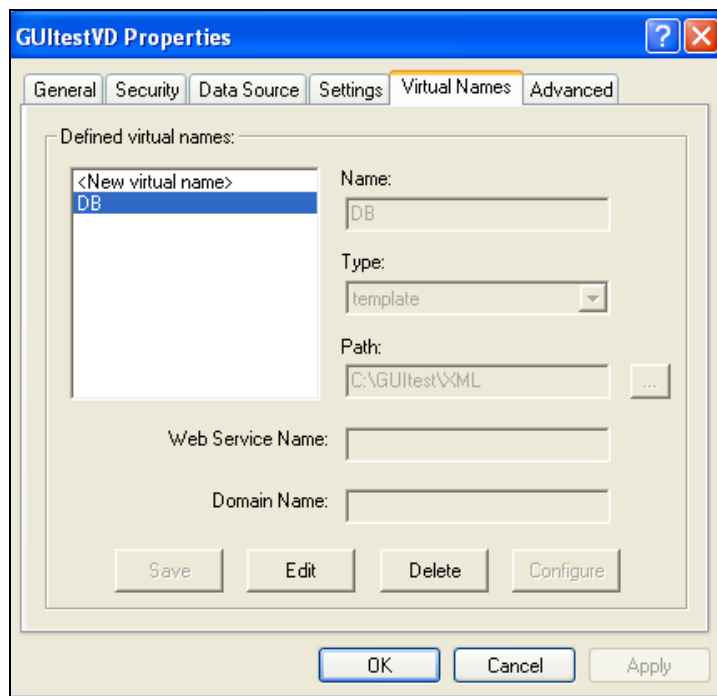
2 pav. Virtualios direktorijos pavadinimo bei kelio iki puslapio nurodymas

Security ir Data Source dalyse reikia nurodyti prisijungimą prie duomenų bazės bei serverio pavadinimą bei pačią duomenų bazę. Skiltyje Settings reikia uždėti tas varneles, kurios matomos žemiau.



3 pav. Ilustracija, kurias varneles reikia pažymėti „Options“ dalyje.

Toliau reikia sukurti virtualių vardų sritį skiltyje Virtual Names, kurios kelias rodo į db aplanką, o pavadinimas yra puslapiai. Kuriuose laukuose ką nurodyti matoma sekančiame paveiksle.

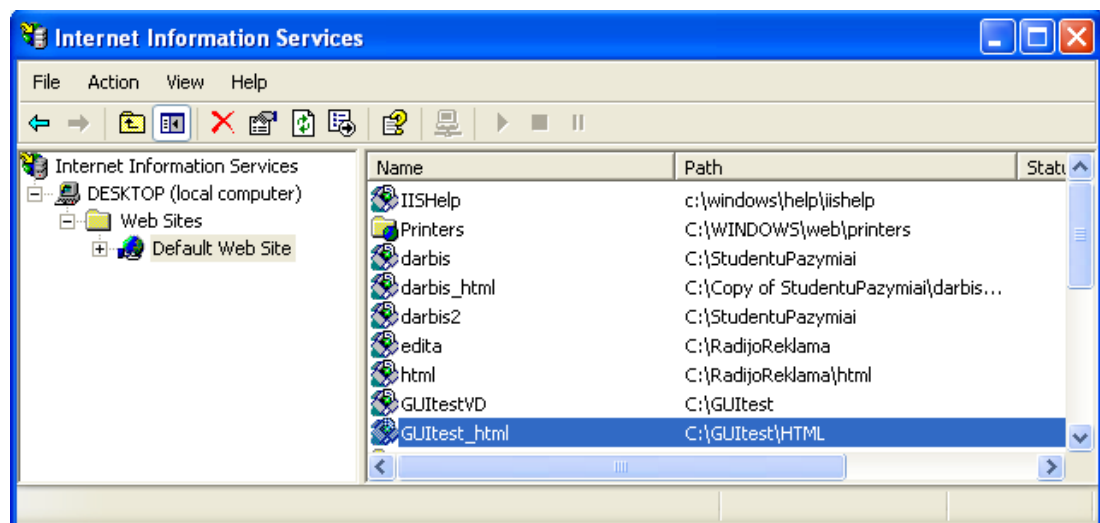


4 pav. Virtualios vardų srities nurodymas

Visa tai atlikus adresas iki prisijungimo formos turi būti panašus į tokį:

<http://localhost/GUItestVD/DB/sistema.xml>

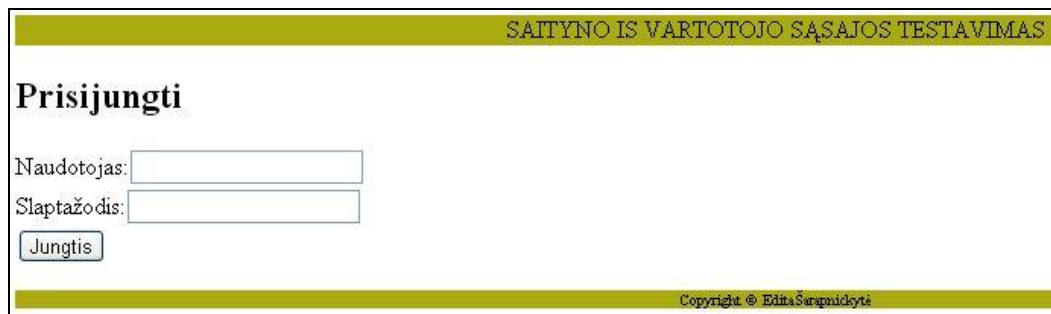
Prieš jungiantis taip pat reikia nurodyti kelią iki CSS aplanko, kuriame yra visi paveikslėliai bei stiliaus failai. Tai atliekama Internet Information Services įrankio pagalba. Jame į Default Web Site reikia įtraukti naują virtualią direktoriją. Pirmiausiai turime įvesti vardą GUItest_html, paspaudus next mygtuką nurodome adresą iki CSS aplanko.



5 pav. Css aplanko įtraukimas į Internet Information Services

Sistemos funkcinis aprašymas

Prisijungus prie sistemos atidaromas sistemos prisijungimo langas (6 pav.). Prisijungimo lange būtina nurodyti sistemos naudotojo prisijungimo vardą ir slaptažodį



6 pav. Sistemos prisijungimo langas

Prisijungimas prie sistemos galimas trimis skirtingais sistemos naudotojų tipais – testuotojas, programuotojas ir analitikas.

Prisijungus prie sistemos testuotoju pateikiamas pagrindinis sistemos langas (7 pav.)



7 pav. Pradinis sistemos langas

Testuotojas gali:

A1 – įvesti naujus VS komponentus, peržiūrėti ir koreguoti esamų sąrašą. Priskirti komponentams testavimo atvejus ir testavimo atvejams priskirti žinomus sprendimus.

A2 – sukurti naują testą, peržiūrėti esamų sąrašą ir jį koreguoti. Sudarius testą galima jį atlikti.

A3 – peržiūrėti užregistruotus defektus pagal jų priklausomybę testams, priskirti arba užregistruoti naują sprendimą defektui.

A1. Pasirinkus meniu punktą Komponentai ir TA yra pateikiamas visų duomenų bazėje saugomų VS komponentų sąrašas (8 pav.).

Komponentai ir TA Testai Defektai ir sprendimai	ID	KOMPONENTAS	veiksmas	
	34	apatinė eilutė		 pridėti TA
33	formos		 pridėti TA	 trinti
38	įvedimo laukai		 pridėti TA	 trinti
25	langai		 pridėti TA	 trinti
29	menu		 pridėti TA	 trinti
28	mygtukai		 pridėti TA	 trinti
43	navigacija		 pridėti TA	 trinti
26	nuorodos		 pridėti TA	 trinti
37	pagalbos priemonės		 pridėti TA	 trinti
32	paieška		 pridėti TA	 trinti
44	paveikslukai		 pridėti TA	 trinti
45	pažymimosios varnelės		 pridėti TA	 trinti
35	piktogramos		 pridėti TA	 trinti
36	pranešimų tekstai		 pridėti TA	 trinti
39	punktavimas		 pridėti TA	 trinti
40	spalvų paletė		 pridėti TA	 trinti
30	svetainės logotipas		 pridėti TA	 trinti
27	teksto blokai		 pridėti TA	 trinti
31	viršutinė eilutė		 pridėti TA	 trinti

Komponento pavadinimas:

8 pav. Meniu „Komponentai ir TA“ langas

Šiame lange galima pridėti naują VS komponentą arba ištrinti pasirinktą. Taip pat galima peržiūrėti jau komponentui priskirtus testavimo atvejus (9 pav.).

Komponentai ir TA Testai Defektai ir sprendimai	ID	EURISTIKA	TIPAS	veiksmas
	80	Visada yra aišku kuris klaidos ar informacinis pranešimas kuriam funkcionalumui priklauso.	U	 pridėti sprendimą
81	Informaciniai pranešimai visada pateikiami zonoje į kurią žiūri vartotojas.	U	 pridėti sprendimą	
82	Pateikiami konkretūs ir vartotojui suprantami pranešimai.	U	 pridėti sprendimą	
83	Klaidų, paaiškinamiesiems ir informaciniams pranešimams taikomi skirtingos teksto spalvos.	U	 pridėti sprendimą	
84	Vartotojui visada yra suprantama, kodėl jis nemato informacijos ar kodėl jo yra apriboti.	U	 pridėti sprendimą	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Pridėti"/>

Copyright © Eita Šešpudytė

9 pav. Testavimo atvejų pridėjimo komponentui langas

Pasirinktam VS komponentui galima pridėti naują TA paspaudus sprendimo pridėjimo nuorodą. Atidaroma naujo sprendimo pridėjimo forma (10 pav.), kurioje būtina nurodyti sprendimo aprašymą. TA išsaugomas duomenų bazėje sukuriant ryšį su jam priskirtu VS komponentu.

Komponentai ir TA Testai Defektai ir sprendimai	ID		SPRENDIMAS		veiksmas
	32	Privaloma laukai turi būti pažymėti specialiu simboliu, kuris būtų paašškintas apie informacijos įvedimo į laukus būtinumą.			<input type="text"/>
Copyright © Edita Širgėdaitė					

10 pav. Testavimo atvejų pridėjimo komponentui langas

Suvedus visus pageidaujamus komponentus, priskyrus jiems testavimo atvejus ir jų sprendimus, galima pradėti testo kūrimą.

A2. Pasirinkus meniu punktą Testai atidaromas duomenų bazėje egzistuojančių testų sąrašas (11 pav).

Komponentai ir TA Testai Defektai ir sprendimai	ID	Pavadinimas	Veiksmas		
	16	Savitarna1_T1	sudaryti	trinti	atlikti
	17	Savitarna1_T2	sudaryti	trinti	atlikti
	18	Savitarna1_T3	sudaryti	trinti	atlikti
	19	Savitarna2_T1	sudaryti	trinti	atlikti
	20	Savitarna2_T2	sudaryti	trinti	atlikti
	21	Savitarna2_T3	sudaryti	trinti	atlikti
	22	Savitarna3_T1	sudaryti	trinti	atlikti
	23	Savitarna3_T2	sudaryti	trinti	atlikti
	24	Savitarna3_T3	sudaryti	trinti	atlikti
	Testo pavadinimas: <input type="text"/>		<input type="button" value="Pridėti"/>		

11 pav. Meniu punkto „Testai“ langas

Šiame lange galima sukurti naują ir ištrinti pasirinktą testą. Testas sudaromas pasirenkant komponentus, kurių testavimo atvejus norima įtraukti į testą, tai atliekama paspaudus nuorodą sudaryti.

Komponentai ir TA Testai Defektai ir sprendimai	Pasirinkite komponentus, kurie yra testuojamame saityne		
	ID	PAVADINIMAS	įtraukti
	34	apatinė eilutė	<input type="checkbox"/>
	33	formos	<input type="checkbox"/>
	38	įvedimo laukai	<input type="checkbox"/>
	25	langai	<input type="checkbox"/>
	29	menu	<input type="checkbox"/>
	28	mygtukai	<input type="checkbox"/>
	43	navigacija	<input type="checkbox"/>
	26	nuorodos	<input type="checkbox"/>
	37	pagalbos priemonės	<input type="checkbox"/>
	32	paieška	<input type="checkbox"/>
	44	paveikslukai	<input type="checkbox"/>
	45	pažymimosios varnelės	<input type="checkbox"/>
	35	piktogramos	<input type="checkbox"/>
	36	pranešimų tekstai	<input type="checkbox"/>
	39	punktavimas	<input type="checkbox"/>
	40	spalvų paletė	<input type="checkbox"/>
	30	svetainės logotipas	<input type="checkbox"/>
	27	teksto blokai	<input type="checkbox"/>

12 pav. Testo sudarymo langas

Norint sudaryti testą pakanka sužymėti varneles prie komponentų, kurių testavimo atvejai turėtų būti įtraukti, ir paspausti mygtuką Pridėti (12 pav.).

Testas galima atlikti paspaudus nuorodą atlikti, kuri pateikiama testų sąrašo lange.

Komponentai ir TA Testai Defektai ir sprendimai	TA Nr.	Euristika	Tipas	Būsena	Defektas	Veiksmas
	47	Tame pačiame lange pateikiami skirtingi funkcionalumai.	U	NG		Saugoti
	48	Langų dizainas visoje sistemoje turi vieno da/vientisą stilių.	U	OK		Saugoti
	49	Lango elementai nėra "sulipę", jie yra atskiriami vieno do dydžio tarpais.	U	OK		Saugoti
	50	Po kiekvieno vartotojo atlikto veiksmo, kuris inicijuoja kažkokio rezultato pateikimą, arba kuris inicijuoja perėjimą veiksmų seka (angl. flow), vartotojui pateikiami laukimo langai.	U	NG		Saugoti
	51	Vartotojui atliekant veiksmų seka turi būti rodoma kuriame žingsnyje jis yra.	U	OK		Saugoti
	52	Nuoroda į išorinį puslapį atidaroma naujame naršyklės lange.	U	OK		Saugoti
	53	Nuoroda į vidinius puslapius atidaroma tame pačiame naršyklės lange.	U	OK		Saugoti
	54	Nuorodoms visoje saityno informacinėje sistemoje taikomas vieno da stilius.	U	NG		Saugoti
	55	Nuorodos turi tris būsenas – normali (angl. normal)/užvestas (angl. hover)/paspaustas (angl. pressed)	U	OK		Saugoti

13 pav. Testo atlikimo langas

Pasirinkus testo atlikimą pateikiami jam priklausantys testavimo atvejai. Galima keisti testavimo atvejų būseną į T arba N (teigiamas arba neigiamas). Testavimo atvejui, kuris įgyja būseną neigiamas, galima pridėti defekto aprašymą.

Komponentai ir TA Testai Defektai ir sprendimai	ID	DEFECTAS	Veiksmas
	35	Ne visai aišku, ką reiškia raudonos žvaigždutės prie įvedimo laukų registracijos formoje.	pridėti sprendimą trinti
	37	Ne visada aišku ar tai neaktyvi nuoroda ar nuorodos stilių turintis tekstas, pvz. "Pamiršote slaptažodį?".	pridėti sprendimą trinti
	38	Informacinis pranešimas informuojantis apie sėkmingą registraciją nesiskiria nuo paprasto teksto.	pridėti sprendimą trinti
	39	Informacinis pranešimas informuojantis apie sėkmingą registraciją pateikiamas zonoje į kurią nežūri vartotojas.	pridėti sprendimą trinti
	40	Informacinis pranešimas informuojantis apie sėkmingą registraciją pateikiamas po antrašte, kuri neatitinka pranešimo (pvz. "Savitarnos svetainėje galėsite Registracija sėkminga").	pridėti sprendimą trinti
	41	Pagrindiniame lange po prisijungimo pateikiamas pašto funkcionalumas du kartus, tampa neaišku ar funkcionalumas skirtingas ar vieno da.	pridėti sprendimą trinti
	42	Ne visada aišku kurioje sistemos vietoje yra vartotojas.	pridėti sprendimą trinti
	43	Aktyvių menu punktų/skirtukų žymėjimas ne visada yra aiškus, nes skirtingais atvejais jis yra taikomas skirtingas.	pridėti sprendimą trinti
	44	Vartotojui inicijavus veiksmą, kuris trunka ilgiau nei puslapio atidarymas, neaišu ką veikai sistema (pvz. paspaudus mygtuką prisijungti ar atsijungti).	pridėti sprendimą trinti

14 pav. Registruotų testo defektų sąrašas

A3. Testo atlikimo metu pastebėtus ir užregistruotus defektus galima peržiūrėti sugrupuotus pagal testus pasirinkus meniu nuorodą „Defektai ir sprendimai“. Pasirinkus norimo testo defektų peržiūrą yra pateikiamas defektų sąrašas (14 pav.). Pasirinkus prie konkretaus defekto esančią nuorodą „ pridėti sprendimą“ galima peržiūrėti siūlomus defekto pašalinimo sprendimo būdus ir pasirinkti tinkamą, arba pridėti naują, defektui šalinti tinkamą sprendimą (15 pav.).

SAITYNO IS VARTOTOJO SAŠAJOS TESTAVIMAS		
Komponentai ir TA Testai Defektai ir sprendimai	ID	PRISKIRTI SPRENDIMAI
	142	Informaciniai ir klaidų pranešimai turėtų būti išskiriami priešingomis spalvomis.
	ID	SIŪLOMI SPRENDIMAI
	140	Informacinis pranešimas gali būti išskiriamas specialia piktograma pranešimo pradžioje.
	141	Informacinis pranešimas turėtų būti prēmintas.
Naujas sprendimas: <input type="text"/> arba pasirinkti sprendimai: <input type="text"/>		<input type="button" value="Pridėti"/>
atsijungti		
Copyright © Edita Šarapinskaitė		

15 pav. Defekto šalinimo sprendimo būdų priskyrimo langas

Siūlomų defekto šalinimo sprendimo būdų sąrašas priklauso nuo dviejų priklausomybių. Gali būti pateikiami testavimo atvejui, kuriam priklauso defektas, priskirti sprendimai, arba pateikiami to paties testavimo atvejo kitiems defektams pasiūlyti sprendimai.