

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Viktorija Šilinskaitė

**AGILE POŽIŪRIO TAIKYMAS INTERNETINIŲ
SVETAINIŲ KŪRIME**

Magistro baigiamasis darbas

Darbo vadovas: doc. Rita Butkienė

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Viktorija Šilinskaitė

**AGILE POŽIŪRIO TAIKYMAS INTERNETINIŲ
SVETAINIŲ KŪRIME**

Magistro baigiamasis darbas

Recenzentas

dr. Tomas Blažauskas

2008-01-

Vadovas

dr. doc. Rita Butkienė

2008-01-

Atliko

**IFM-2/4 gr. stud.
Viktorija Šilinskaitė**

2008-01-14

Kaunas, 2008

The Application of Agile Approach in the Development of Web Sites

Summary

A fast increase in the number of web sites the last years, shows the importance of information existing in web sites giving the possibility for users to have access to it. Many companies create websites, but they face problems related with requirements changing until the developing process completes.

In this work you will find analysis of software development methods. Agile approach allows changing requirements even in last phase of the developing process. Creating a web site does not follow one single path and there is no ideal method creating a web site. In this dissertation will be analyzed a method for web sites development based on Agile approach.

Turinys

Ivadas.....	8
Agile požiūrio taikymo internetinių svetainių kūrime analizė.....	10
0.1 Tyrimo sritis, objektas ir problema.....	10
0.2 Analizės tikslas.....	10
0.3 Analizės metodai.....	10
0.4 Tyrimo objekto analizė.....	11
0.4.1 Internetinės svetainės apibrėžimas.....	11
0.4.2 Internetinės sistemos architektūra.....	12
0.4.3 MVC architektūra.....	14
0.4.4 Internetinių svetainių struktūra.....	15
0.4.5 Internetinių sistemų kūrimui naudojamos technologijos bei diegimo aplinkos.....	16
0.5 Vartotojų analizė.....	16
0.5.1 Vartotojų aibė, tipai ir savybės.....	16
0.5.2 Vartotojų tikslai ir problemos.....	17
0.6 Sprendimo analizė.....	18
0.6.1 Internetinių svetainių kūrimo problematika.....	18
0.6.1.1 Kainos vaidmuo kūrime.....	18
0.6.1.2 Kūrėjo klaidos internetinių svetainių kūrime.....	19
0.6.1.3 Kliento klaidos internetinių svetainių kūrime.....	21
0.6.2 Sistemos kūrimo metodų analizė.....	22
0.6.2.1 Internetinių taikomųjų programų kūrimas panaudojant Agile.....	28
0.6.2.2 Agile metodologija.....	29
0.6.2.3 Agile metodų analizė.....	31
0.7 Analizės išvados.....	39
1 Metodo, skirto internetinių svetainių kūrimui, reikalavimų specifikacija.....	40
1.1 Siūlymo taikymo sritis, sąlygos ir prielaidos.....	40
1.2 Funkciniai ir nefunkciniai reikalavimai.....	40
1.2.1 Funkciniai reikalavimai.....	40
1.2.2 Nefunkciniai reikalavimai.....	40
1.3 Duomenų modelis.....	41
2 Metodo, skirto internetinių svetainių kūrimui, projektas.....	42
2.1 Kada taikyti metodą?.....	42
2.2 Pagrindinė idėja.....	42
2.3 Principai.....	42
2.4 Metodo savybės.....	43
2.5 Komanda.....	43
2.6 Kūrimo procesas.....	44
2.6.1 Reikalavimų surinkimas.....	45
2.6.2 Susirinkimas.....	53
2.6.3 Architektūros nustatymas.....	54
2.6.4 Dizainas ir prototipo kūrimas.....	55
2.6.5 Iteracinis kūrimas.....	56
2.6.6 Kūrėjų testavimas.....	57
2.6.7 Užsakovo testavimas ir naujų reikalavimų pateikimas.....	58
2.6.8 Galutinis produktas.....	58
2.7 Komandos bendradarbiavimas.....	58
2.8 Užsakovo įtraukimas.....	59
3 Eksperimentinis tyrimas.....	60
3.1 Eksperimentinio tyrimo projektas ir realizacija.....	60
3.1.1 Dalykinė sritis.....	60
3.1.2 Vartotojai ir sistemos funkcijos.....	60

3.1.3 Duomenų bazė.....	66
3.1.4 Vartotojo sąsaja. Dizainas.....	67
3.1.5 Vartotojo sąsaja. Sistemos lango struktūra.....	68
3.1.6 Vartotojo sąsaja. Sistemos struktūra.....	68
3.1.7 Nefunkciniai reikalavimai ir apribojimai.....	69
3.1.7.1 Sistemos saugumo priemonės.....	69
3.1.7.2 Sistemos prieinamumas.....	70
3.1.7.3 Sistemos apribojimai.....	70
3.1.7.4 Techninė specifikacija.....	70
3.1.8 Architektūros nustatymas.....	70
3.1.9 Kūrimo procesas.....	71
3.2 Ekspertiniai vertinimai.....	73
3.2.1 Produkto įvertinimas.....	74
3.2.1.1 Projekto trukmė.....	74
3.2.1.2 Vartotojų įvertinimas.....	74
3.2.1.3 Internetinės svetainės „Jadvygos žolės“ kokybės pagal ISO 9126 standartą įvertinimas.....	75
4 Rezultatai ir išvados.....	77
5 Literatūra.....	78
6 Priedai.....	80

Lentelių sąrašas

1 lentelė Vartotojų problemų aprašymo pavyzdys.....	11
2 lentelė Vartotojų problemų aprašymas.....	17
3 lentelė Krioklio modelio analizė.....	22
4 lentelė Pakartotinio panaudojimo modelio analizė.....	23
5 lentelė Evoliucinio modelio analizė.....	23
6 lentelė Formaliojo modelio analizė.....	24
7 lentelė Žingsninio kūrimo analizė.....	25
8 lentelė Spiralinio kūrimo analizė.....	25
9 lentelė Agile metodai.....	26
10 lentelė XP analizė.....	31
11 lentelė DSDM analizė.....	32
12 lentelė FDD analizė.....	33
13 lentelė Scrum analizė.....	35
14 lentelė Agile metodų savybių apibendrinimas.....	35
15 lentelė Agile metodų procesai.....	36
16 lentelė Vaidmenų atsakomybės.....	44
17 lentelė Dalykinės srities sąvokų žodynas.....	47
18 lentelė Vartotojo registracijos pavyzdys.....	47
19 lentelė Užsakymo sudėtiniai elementai.....	49
20 lentelė Testavimo rezultatai.....	58
21 lentelė Dalykinės srities sąvokų žodynas.....	60
22 lentelė Vartotojų registracija.....	63
23 lentelė Užsakymas.....	64
24 lentelė Užsakymo sudėtiniai elementai.....	66
25 lentelė Vartotojų registracijos duomenys.....	66
26 lentelė Testavimo rezultatai.....	73
27 lentelė Svetainės „Jadvygos žolės“ kokybės įvertinimas.....	76

Paveikslų sąrašas

1 pav. Sistemos architektūra.....	13
2 pav. Model-View-Controller architektūrinio šablono schema.....	14
3 pav. Extreme Programming (XP) metodo schema 5.....	31
4 pav. FDD proceso schema 5.....	33
5 pav. FDD proceso ciklas 5.....	33
6 pav. Scrum proceso schema 5.....	34
7 pav. Funkciniai metodo reikalavimai.....	40
8 pav. Projekto schema.....	45
9 pav. Reikalavimų surinkimo metu įeinantys ir išeinantys duomenys.....	45
10 pav. Reikalavimų surinkimo schema.....	46
11 pav. Panaudojimo atvejo „Redaguoti prekių katalogą“ veiklos diagrama.....	48
12 pav. Pavyzdinė sistemos lango struktūros schema.....	51
13 pav. Svetainės struktūra - šoninis meniu.....	52
14 pav. Specifikacijų pateikimo įeinantys ir išeinantys duomenys.....	53
15 pav. Architektūros nustatymo įeinantys ir išeinantys duomenys.....	54
16 pav. MVC architektūros pavyzdys.....	55
17 pav. Dizaino ir prototipo kūrimo įeinantys ir išeinantys duomenys.....	55
18 pav. Dizaino ir prototipo kūrimo schema.....	56
19 pav. Kūrimo proceso iteracija.....	57
20 pav. Panaudos atvejų diagrama.....	61
21 pav. Panaudojimo atvejo „Redaguoti prekių katalogą“ veiklos diagrama.....	65
22 pav. Duomenų bazės schema.....	67
23 pav. Sistemos lango struktūra.....	68
24 pav. Svetainės struktūra - šoninis meniu.....	69
25 pav. MVC architektūros pavyzdys.....	70
26 pav. Projekto trukmės lentelė.....	74
27 pav. Užsakovo atsiliepimai.....	75
28 pav. Vartotojo sąsajos langas.....	80
29 pav. Vartotojo sąsajos langas. Žolelių mišiniai.....	80
30 pav. Vartotojo sąsajos langas. Vartotojų tvarkymas.....	81
31 pav. Vartotojo sąsajos langas. Užregistruoti naują vartotoją.....	81
32 pav. Vartotojo sąsajos langas. Rašyti naują pranešimą.....	82
33 pav. Vartotojo sąsajos langas. Mano žinutės.....	82
34 pav. Vartotojo sąsajos langas. Siųsti naujienas el. paštu.....	83
35 pav. Vartotojo sąsajos langas. Redaguoti prekių katalogą.....	83

Išvadas

D. Šilingo straipsnyje „Programinės įrangos reikalavimų valdymo principai ir praktika“ teigiama: „Statistika teigia, kad apie $\frac{3}{4}$ programinės įrangos projektų nepasie užsibrėžtų tikslų — vėluoja, viršija biudžetą arba visiškai sužlunga ir yra nutraukiami“ 5. Dažniausiai atsitinka taip, kad dauguma problemų atsiranda reikalavimų apibrėžimo ir valdymo srityje. Vartotojas yra nepakankamai įtraukiamas, todėl reikalavimai yra neteisingai suprantami. Dažnai jie būna nevienareikšmiškai, nepilnai apibrėžiami, todėl proceso eigoje keičiasi.

Kiekviena įmonė kurdama sistemą pasirenka jai patraukliausią metodą. Metodo pasirinkimas dažnai priklauso nuo projekto finansavimo, projekto dydžio, užsakovo poreikių. Kokį metodą pasirinkti, įtakos turi ir tai, ar klientas yra pasiruošęs bendrauti su kūrėju. Pavyzdžiui į programavimo komandą, kuri naudojasi *Agile* metodu, įtraukiamas užsakovas, detalizuojantis ir nustatantis darbų prioritetą, taip pat galintis iškart atsakyti į iškilusius klausimus.

Sistemos kūrimo metodas turi būti pasirenkamas priklausomai nuo sistemos apimties, projekto finansavimo pobūdžio ir pan. Visada reikia atsižvelgti į tai, ar užsakovas gali dalyvauti kūrimo procese, ar reikalavimai, realizuojant sistemą, gali kisti. Pagal tradicinius reikalavimų inžinerijos principus, projekto apimtis ir kaštai vertinami pagal reikalavimus 5. Taip pat, priklausomai nuo reikalavimų, yra sudaromas darbų tvarkaraštis 5. Pasikeitus reikalavimas, kaina labai išauga, kai projektas pereina į vėlesnes fazes, nes dažnai tenka taisyti ne tik programinį kodą, bet ir susijusių reikalavimų aprašymus bei projektą modelius.

2007 m. gruodžio mėnesį Netcraft internetinių serverių paieškos sistema (Netcraft Web Server Survey) rado 155 milijonus atskirų internetinių puslapių, iš kurių net apie 50 milijonų atsirado nuo 2006 metų gruodžio mėnesio 5. Tai rodo didelį internetinių sistemų poreikį, o efektyviam svetainių kūrimui reikalinga pritaikyti tinkamą kūrimo metodą.

Yra įvairių sistemų kūrimo metodų, tačiau problema iškyla jų pritaikomume internetinių svetainių kūrimui. Kaip sukurti kokybišką internetinę svetainę, informacijos yra daug, tačiau, kaip turi būti atliekamas ir organizuojamas pats internetinių svetainių kūrimo procesas, kokį metodą taikyti, nėra pakankamai, kadangi yra orientuojamasi apskritai į sistemų kūrimą, o ne į internetinių svetainių kūrimą.

Šio darbo analizės dalyje yra išanalizuotas internetinių sistemų tipinis kūrimo procesas, kūrimo problematika, esamos programų kūrimo metodikos, jų privalumai ir trūkumai. Pagal internetinėms svetainėms būdingas kūrimo savybes išanalizuoti *Agile* metodai. Analizės

tikslas yra suprast svetainių kūrimo procesą, kuo šis procesas skiriasi nuo kitokio tipo sistemų kūrimo, kokie metodai yra taikomi arba galėtų būti taikomi internetinėms svetainėms kurti.

Darbo tikslas – sukurti metodą, skirtą internetinių svetainių kūrimui, ir jį pritaikyti kuriant internetinę svetainę.

Agile požiūrio taikymo internetinių svetainių kūrime analizė

0.1 Tyrimo sritis, objektas ir problema

Internetinė svetainė gali teikti paslaugą, parduoti produktą, pateikti teminę informaciją, skelbimus, informuoti žmones apie asmenį (pvz. asmeninė svetainė), pateikti naujienas, o taip pat gali būti kuriami forumai konkrečioms problemoms spręsti. Pagrindinis svetainės kūrimo tikslas turi būti aiškus, vartotojams lengvai suprantamas. Informacijos, kaip sukurti kokybišką internetinę svetainę, kaip turi būti pateikta informacija, kaip išdėstyti meniu, yra daug, tačiau, kaip turi būti atliekamas ir organizuojamas pats internetinių svetainių kūrimo procesas, kokį metodą taikyti, nėra pakankamai, kadangi yra orientuojamasi apskritai į sistemų kūrimą, o ne į internetinių svetainių kūrimą. Yra reikalingas metodas, pritaikytas internetinių svetainių kūrimui.

Problema. Sistemų kūrimo metodų yra įvairių, tačiau problema iškyla jų pritaikomume internetinių svetainių kūrimui.

Tyrimo objektas – procesas, skirtas internetinių svetainių kūrimui.

Tyrimo sritis – metodo, skirto internetinių svetainių kūrimui, sukūrimas ir pritaikymas.

0.2 Analizės tikslas

Analizės tikslas yra suprast svetainių kūrimo procesą, kuo šis procesas skiriasi nuo kitokio tipo sistemų kūrimo, kokie metodai yra taikomi arba galėtų būti taikomi internetinėms svetainėms kurti.

0.3 Analizės metodai

Tyrimo objekto analizei atlikti bus naudojamas teorinės analizės ir apibendrinimo metodas. Šioje dalyje bus pateiktas internetinės svetainės apibrėžimas, internetinių svetainių ir kitokių sistemų skirtumai, kokios yra išskirtinės jos savybės, bendroji internetinių svetainių struktūra (komponentai), galimos architektūros, naudojamos technologijos, diegimo aplinkos, vartotojai ir pan.

Vartotojų analizė vartotojai bus išanalizuoti šiais aspektais:

- Kokie yra vartotojų tipai,
- Kokia vartotojų atsakomybė,
- Kokia vartotojų kvalifikacija,
- Kiek yra vartotojų,

- Kokia yra rinka.

Taip pat bus išanalizuoti vartotojų tikslai ir problemos. Problemos bus pateikiamos tokia lentele:

1 lentelė Vartotojų problemų aprašymo pavyzdys

Problema	Kaip viskas vyksta dabar

Aprašius problemas bus pateikta, kaip nauja sistema išspręs šias problemas.

Sprendimo analizei bus pritaikytas teorinės analizės ir apibendrinimo metodas. Šioje dalyje bus aptarta:

- 1) Internetinių sistemų tipinis kūrimo procesas ir jo ypatybės.
- 2) Kūrimo problematika.
- 3) Esamos programų kūrimo metodikos. Išanalizuoti metodų privalumai ir trūkumai, pateiktas pagrindimas, kodėl internetinių svetainių kūrimui tinkamesnis yra *Agile* požiūris.
- 4) Konkretūs *Agile* metodai.

Sprendimų analizė parodys naujo sprendimo reikalingumą.

0.4 Tyrimo objekto analizė

Šiame skyriuje pateikiamas internetinės svetainės apibrėžimas, skirtumas nuo kitokių sistemų, naudojamos technologijos, diegimo aplinkos ir pan.

0.4.1 Internetinės svetainės apibrėžimas

Įvairiuose šaltiniuose, priklausomai nuo autoriaus, jo nagrinėjamos srities, yra pateikiama skirtingų internetinės svetainės apibrėžimų.

J. Ruževičiaus ir N. Gusevos straipsnyje „Interneto svetainių kokybės vertinimo ypatumai“ teigiama: „Žodynuose internetinė svetainė apibrėžiama skirtingai. Monash universiteto (Australija) žodyne internetinė svetainė apibrėžiama kaip rinkinys internetinių puslapių ir grafinių elementų, susijusių hipertekstu. Tasmanijos universiteto (Australija) mokslininkai internetinę svetainę apibrėžia kaip organizacijos, asmens ar interesų grupės internetinių puslapių rinkinį, kurio pirmasis puslapis yra pagrindinis. Pagal prancūzų autorių E. Sloim (2001), internetinė svetainė – susijusi nuorodomis puslapių visuma, įdėta tame pačiame serveryje ir prieinama internetu. Galima siūlyti ir tokį apibrėžimą: internetinė svetainė – tai sisteminis produktas, sujungiantis koncepciją, instrumentus, metodus, techniką, žmones, tinklą, interfeisą ir teikiamas paslaugas, susietus internetinių puslapių sisteminiu rinkiniu“ 5.

Kompiuterinės leksikos aiškinamajame žodyne pateikiamas toks svetainės apibrėžimas: „Rinkinys [tinklapių](#), kuriuos sieja bendra tematika, priklausomybė vienai įstaigai ar kitokie bendri dalykai.“ 5

Internetinės svetainės yra skirtos teikti paslaugą, parduoti produktą, pateikti informaciją tam tikra tema, skelbimus, informuoti žmones apie asmenį (pvz. asmeninė svetainė), pateikti naujienas, o taip pat gali būti kuriami forumai konkrečioms problemoms spręsti.

Internetinės svetainės vartotojai yra visi asmenys, kurie naudodamiesi naršykle internetu pasiekia įvairius tinklalapius.

Yra trijų tipų vartotojai:

- nepatvirtinti vartotojai (anoniminiai);
- registruoti vartotojai;
- administratoriai.

Anoniminiai yra tokie vartotojai, kurie tik naršo internetinę svetainę ir niekada neužsiregistruoja arba niekada neprisijungia.

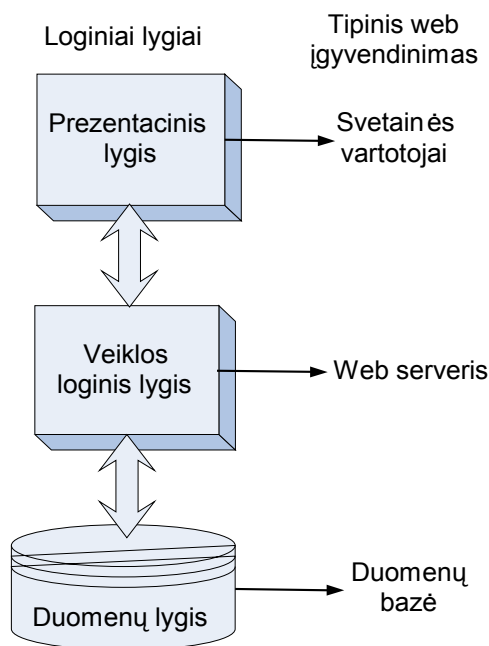
Registruoti vartotojai yra tokie, kurie prisijungia. Tam, kad prisijungtų, jie turi prieš tai būti užsiregistravę. Registruotų vartotojų asmeniniai duomenys yra žinomi, todėl jiems galima suteikti specialius leidimus peržiūrėti ar koreguoti dalykus, kurie kitiems neprieinami.

Nebūtinai internetinėje svetainėje turi būti registruoti vartotojai. Tai yra pasirinktina galimybė, kadangi galima turėti internetinę svetainę, kurios turinys laisvai prieinamas visiems. Tačiau bet kokių atveju, internetinei svetainei visada yra bent vienas registruotas vartotojas, t.y. administratorius. 5

Yra nagrinėjamas atvejis, kai įmonė ne tik pateikia informaciją apie save, veiklą, produkciją, o taip pat yra prieiga prie informacinės sistemos, kuri leidžia įvairias funkcijas. Taigi, tai nėra vien statiniai puslapiai, tai yra internetinė informacinė sistema.

0.4.2 Internetinės sistemos architektūra

Internetinė svetainė (sistema) savo architektūra skiriasi nuo kitų sistemų. Internetinės sistemos architektūra pavaizduota 1 pav. Naudojama reliacinė duomenų bazė.



1 pav. Sistemos architektūra

Sistemos architektūra susideda iš trijų dalių:

- Prezentacinio lygio, kuris generuoja internetinius puslapius, įtraukiant ir dinaminį turinį. Šis lygis interpretuoja kliento patvirtintas internetinių puslapių formas.
- Veiklos loginio lygio, kuris priverčia vykdyti patvirtinimus ir kuris valdo interakcijas su duomenų baze.
- Duomenų lygio, kuris saugo duomenis tarp transakcijų. 5

Prezentacinis lygis

Prezentacinis lygis generuoja tinklalapius ir į tinklalapį įtraukia dinaminį turinį. Dinaminis turinys dažniausiai prasideda nuo duomenų bazės (pvz.: sąrašas visų įvykdytų užsakymų). Kitas svarbus prezentacinio lygio darbas yra dekoduoti tinklalapius, ateinančius iš kliento (pvz.: surasti vartotojo įvestus duomenis ir perteikti šią informaciją į veiklos loginį lygį.).

Prezentacinis lygis yra įgyvendintas Apache Web serveryje. Web serveris gali valdyti kelių taikomųjų programų užklausas taip gerai, kaip ir svetainių statiniams tinklalapiams. Yra paremta pirmine konfigūracija. Web serveris žino, kokį tinklalapį aptarnauti.

Veiklos loginis lygis

Didžioji dalis taikomųjų uždavinių logikos yra įrašyta į veiklos loginį lygį. Veiklos logika apima:

- visų reikalingų skaičiavimų ir patvirtinimų įvykdymą;
- Sekų valdymą (įtraukiant sesijos duomenis);
- Visos prezentacinės pakopos duomenų priėmimo valdymą.

Veiklos loginis lygis yra įgyvendintas taikomųjų uždavinių serveryje. Serveris automatizuoja daug paslaugų, tokių kaip transakcijos, saugumas, žinučių siuntimas ir pan.

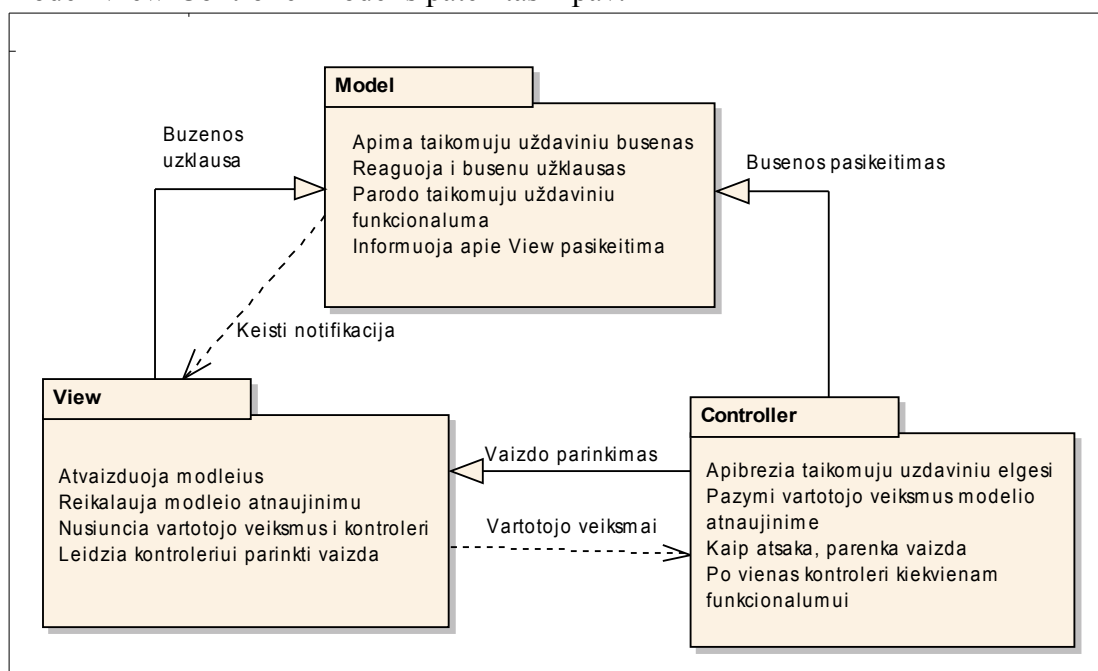
Duomenų lygis

Duomenų lygis yra atsakingas už duomenų valdymą. Duomenų lygis yra reliacinė duomenų bazė. Duomenų lygio paskirtis yra teikti veiklos loginiam lygiui reikalingus duomenis bei saugoti duomenis, kai to yra reikalaujama.

0.4.3 MVC architektūra

Kai norima struktūrizuoti programos kodą taip, kad naudotojo sąsajos kodas, užklausų mechanizmo kodas bei dalykinės srities logikos kodas būtų lengvai keičiamas ir testuojamas atskirai bei būtų galima paskirti atskirus darbus skirtingiems žmonėms. Pvz. jei PHP kodas, atsakingas už verslo transakcijas persipynęs su kodu generuojančiu HTML, prireikus pakeisti HTML kodą, dažnai tai pakankamai sunkus darbas. naudodami MVC (Model-View-Controller) šabloną mes atskiriame kodą atsakingą už sistemos modelį, vaizdą ir kontrolę.

Model-View-Controller modelis pateiktas 2 pav.



2 pav. Model-View-Controller architektūrinio šablono schema

Modelis (Model)

Modelis pristato duomenis ir veiklos taisykles, kurios daro įtaką priėjimui ir tų duomenų atnaujinimui. Modelis yra duomenų, taisyklių, metodų prieiti prie duomenų ir juos keisti rinkinys. Modelis saugo aplikacijos būseną. Atsako į užklausas apie būseną. Pateikia metodus būsenai keisti ir valdyti aplikaciją. Trumpiau būtų galima sakyti, kad modelis yra

viskas, kas susiję su dalykinės srities ar verslo logika. Modelis yra nepriklausomas nuo kontrolierio ir vaizdo komponentų.

Vaizdas (View)

Atsako už modelio pavaizdavimą naudotojui. Naudojasi modeliu. Vaizdo komponentas dažniausiai pateikia rezultatą HTML kodo pavidalu. Remiasi modeliu. Kai modelis keičiasi, jis turi išlaikyti pateikimo nuoseklumą, logiškumą.

Kontrolieris (Controller)

Naudotojo komandas ar užklausas perduoda modeliui, bei parenka reikalingą vaizdą. Apibrėžia aplikacijos veiksmus (veiksmus, kuriuos gali atlikti naudotojas), suriša naudotojo veiksmus su atitinkamais modelio veiksmis, t.y. remiantis vartotojo sąveika ir modelio veiksmų rezultatais, kontrolieris duoda atsaką parinkdamas tinkamą vaizdą. Ši dalis yra atsakinga už visą darbą su interneto protokolu ir gautų užklausų apdorojimą. Priklausomas nuo modelio ir vaizdo komponentų.

0.4.4 Internetinių svetainių struktūra

Visi tinklapiai yra sukurti HTML pagrindu. HTML (HyperText Markup Language) yra hipertekstų žymėjimo kalba".

Tinklapiai skirstomi į statinius ir dinامينius. Statiniai tinklapiai yra tokie, kurių turinys, dizainas ar kiti komponentai nėra priklausomi nuo laiko, kintamųjų reikšmių ar kitų kintamų dydžių ir yra pastovūs. Šių tinklapių turinį ar kitus komponentus rankiniu būdu gali pakeisti tik tinklapio kūrėjas. Dinaminiai - tai tinklapiai, kurių turinys, dizainas ar kiti komponentai gali keistis priklausomai nuo laiko, kintamųjų reikšmių ar kitų kintamų dydžių.

Tinklapis susideda iš:

- Struktūrinių elementų (dokumento antgalvis ir kūnas (*HTML, HEAD, BODY*), skyriai, poskyriai (*H1-H6*), autoriaus aprašytos struktūros (*DIV, SPAN*), lentelės (*TABLE, COL, TR, TH, TD* ir kt.) , skriptai (*SCRIPT, NOSCRIPT*), nuorodos (*LINK, IMG, OBJECT* ir kt.) bei kiti struktūriniai elementai);
- Prezencijos elementų (stilių lentelės (*PRE, CENTER, BR, STYLE*), teksto stiliai ir šriftai (*TT, I, B, BIG, SMALL, FONT*), spalvos, rėmeliai (*FRAMESET, FRAME, IFRAME, NOFRAMES*) ir kt.);

Tinklapiuose vaizdavimas yra atskirtas nuo struktūros, naudojant .css failus. HTML dokumente yra pateikiamas turinys, jo struktūrą aprašantys elementai (<h1>, <p>, <div> ir t.t.), o CSS (Cascading Style Sheet – pakopiniai stilių šablonai) dokumente - aprašyta kaip konkretų HTML dokumentą naršyklė turi pateikti vartotojui, t.y. HTML dokumento turinys atskirtas nuo išvaizdą aprašančių elementų naudojant CSS.

Pagrindinė komanda, nurodanti, kad bylos tekstas yra parašytas HTML kalba yra <HTML>.....</HTML>. Kiti du pagrindiniai tinklapio elementai yra antraštė (head) ir tinklapio pagrindinė dalis (body). Bendra tinklapio struktūra atrodo taip:

```
<HTML>  
<HEAD>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```

Tinklapio antraštinėje dalyje naudojamos komandos yra <META>, <TITLE>...</TITLE>. Komanda <META> nustato pagrindinius tinklapio parametrus: (kalbą, simbolių koduotę, raktinius tinklapio žodžius ir kt.); Tarp komandų <TITLE>....</TITLE> rašomas interneto naršyklės viršuje matomas tekstas; Dauguma komandų turi savo parametrus, kurie aprašomi komandos pradžioje: <KOMANDA parametrai> </KOMANDA>.

0.4.5 Internetinių sistemų kūrimui naudojamos technologijos bei diegimo aplinkos

Internetinėms svetainėms kurti gali būti naudojamos skirtingos technologijos. Sukurti internetinį puslapį pakanka naudoti ir Notepad teksto redaktorių, tačiau galima naudoti ir specialiai tam sukurtas programines įrangas, tokias kaip Microsoft FrontPage, Macromedia Dreamweaver ar kt. Kuriant internetinę svetainę ar sistemą naudojamos technologijos: HTML/XHTML, PHP, JavaScript, MySQL, CSS, ActionScript (Flash), VBScript, .NET (ASP, C#, C++), Java, Perl, Lingo, SQL (MSSQL, MySQL, PostgreSQL, Oracle), VisualBasic, JSP ir kt.

Norint, kad svetainė būtų pasiekiamą vartotojams, reikia užregistruoti domeno pavadinimą, t.y. adresą, kurį interneto vartotojai įrašo į naršyklę, norėdami rasti tinklapį. Tada reikia rasti svetainei pagrindinį kompiuterį, kuriame ji bus patalpinta. Šis kompiuteris yra vadinamas voratinklio serveriu. Dažniausiai serveris turi palaikyti Apache, PHP ir mySql. Žiniatinklio serveris pateikia svetainę vartotojams, kurie jos ieško naudodami naršykles.

0.5 Vartotojų analizė

Vartotojai yra kelių tipų ir turi skirtingus tikslus. Išsamiau yra pateikiama tolimesniuose poskyriuose.

0.5.1 Vartotojų aibė, tipai ir savybės

Vartotojai yra suskirstyti į tokius tipus:

1. Informacijos ieškotojai – tai anoniminiai neregistruoti vartotojai, ieškantys reikiamos informacijos ar šiaip besidomintys ūkio veikla.
2. Įmonės, užsakančios produkciją – tai sistemoje registruotos įmonės, turintys teisę pateikti užsakymus bei juos peržiūrėti.
3. Asmenys, užsakantys produkciją – tai sistemoje registruoti vartotojai, turintys teisę pateikti užsakymus bei juos peržiūrėti.
4. Savininkas – asmuo, turintis administratoriaus teises sistemoje.
5. Darbuotojai. – asmenys, turintys teisę ataskaitas apie užsakymus bei kitą informaciją.

Didžiausią atsakomybę turi savininkas bei darbuotojai, nes jie turi priėjimą prie pagrindinės informacijos sistemoje. Įmonės bei asmenys, užsakantys produkciją, yra atsakingi už teisingos informacijos pateikimą ūkiui. Informacijos ieškotojai yra neprisijungę vartotojai, todėl jiems nėra suteikta atsakomybė.

Vartotojų kvalifikacija – tai paprasti interneto vartotojai. Todėl sistema turi būti „draugiška“ vartotojui: patogi, paprasta, informatyvi.

Vartotojų skaičius yra didelis. Rinka apima ne tik Lietuvos, bet ir užsienio vartotojus. Todėl internetinė svetainė turi būti pritaikyta ne tik Lietuvos, bet ir užsienio rinkai.

0.5.2 Vartotojų tikslai ir problemos

Klientai nori gauti patikimą ir išsamią informaciją apie ūkio „Jadvygos žolės“ vykdomą veiklą, augalų ekspoziciją, gaminamą produkciją, jos įsigijimo vietas. Be to, klientams dažnai iškyla klausimų, kaip augalai gali padėti jų sveikatai, dažnai iškyla konsultacinių klausimų. O tuo tarpu užsakovai nori greitos ir patikimos užsakymo sistemos. Taigi, vartotojų tikslai yra internetinė sistema, kuri pateiktų reikiama informaciją bei suteiktų galimybes užsisakyti produkciją.

Šiuo metu vartotojai susiduria su problemomis, kurios apsunkina informacijos gavimą ir lėtina ūkio veiklą. Pagrindiniai vartotojų problemų aprašymai pateikiami lentelėje:

2 lentelė Vartotojų problemų aprašymas

Problema	Kaip viskas vyksta dabar
Klientai negauna išsamios ir struktūrizuotos informacijos.	Klientai priversti ieškoti informacijos kitose internetinėse svetainėse. Tačiau ten pateikta informacija yra tik minimali ir neišsami.
Produkcijos užsakymas yra nepatogus.	Klientai, norintys užsisakyti ūkio produkcijos, turi tik vienintelį būdą – telefonu.
Lėtas ir nepatogus ataskaitų formavimas	Darbuotojai rankomis turi suvedinėti užsakymus ir taip sudaryti ataskaitas.

Užsakovai neturi tikslų reikalavimų. Juos reikės išsiaiškinti kūrimo procese. Reikalavimai kils priklausomai nuo užsakovo norų. Sistemos prototipas pakartotinai bus naudojamas ir testuojamas kūrėjo bei užsakovo. Atsiradus naujiems poreikiams automatiškai kinta ir reikalavimai.

Šiuo metu klientai, norėdami gauti informacijos apie įmonės produkciją, turi asmeniškai susisiekti su gamintojais. Įmonė siekia, kad vartotojai galėtų gauti išsamią informaciją apie produkciją, jos pardavimo vietas ir pan., o prekiaujančios įmonės bei pavieniai asmenys iš Lietuvos ir užsienio galėtų pateikti užsakymus internetu. Šiam tikslui įgyvendinti bus kuriama internetinė sistema.

0.6 Sprendimo analizė

0.6.1 Internetinių svetainių kūrimo problematika

Internetiniu svetainių kūrimas dažnai gali būti kebli užduotis. Nuorodų sistemos painumas, technologiniai sprendimai, per dideli lūkesčiai – viskas gali turėti neigiamos įtakos internetinių svetainių kūrimui [17].

0.6.1.1 Kainos vaidmuo kūrime

Pasirašant sutartį, yra svarbu apibrėžti kainą. Klientas ir kūrėjas turi sutarti, ar projektas turės fiksuotą kainą, ar priklausys nuo laiko ir kitų resursų. Fiksuotos kainos kontraktas yra toks, kai sutariama dėl maksimalios produkto pristatymo kainos. Pasirašant sutartį, kai kūrimo kaina priklauso nuo laiko ir kitų resursų, klientas ir kūrėjas susitaria dėl fiksuoto valandinio ar dieninio užmokesčio. Gerai apgalvotas projekto planas leidžia numatyti galimas problemas, todėl žymiai lengviau yra susitarti dėl produkto apmokestinimo. Esant fiksuotai kainai klientas rizikuoja tuo, kad sutarta produkto apimtis gerokai kils. Todėl turint galvoje, kad klientas ir kūrėjas dažnai gali pakeisti savo nuomonę ar poreikius projekto eigoje, turėtų būti kažkiek lankstumo pasirašant sutartį. Dažnai gali atsitikti taip, kad yra sutariama dėl fiksuotos kainos, o vėliau kinta produkto apimtis, arba, sutarus mokėti už sugaištą laiką, produkto kaina tampa pernelyg didelė ir nepriimtina klientui. Todėl geriausia būtų rasti tarpinį variantą.

Sudarant sutartį, visada reikia apibrėžti, kokią įtaką turės pasikeitimai, kaip tie pasikeitimai turi būti įgyvendinti ir kaip jie įtakos produktą bei jo kainą. Be to, klientas visada nori atidėti, užlaikyti fiksuotos kainos mokėjimą iki galutinio produkto atidavimo. Priešingai nei klientas, kūrėjas norės gauti iš anksto kuo daugiau užmokesčio, kiek tik įmanoma tam, kad sumažintų riziką.

Pasirašius sutartį, kai kaina kinta priklausomai nuo laiko ir kitų resursų, rizika dažniausiai iškyla dėl kliento kaltės ir kai pagrindinės projekto savybės nėra aiškiai apibrėžtos ir sutartos, tada klientas bus priverstas sutikti su „atverstos čekių knygutės įteikimu kūrėjui“ [16].

0.6.1.2 Kūrėjo klaidos internetinių svetainių kūrime

Dažnai yra įprasta kaltinti ir kritikuoti klientą, kai projektas nepavyksta. Bet dažnai būna, kad bent dalis kaltės yra kūrėjo.

Dizainas turėtų būti orientuotas į internetinės svetainės funkcijas ir tikslus. Deja, dažnai būna atvirkščiai. Didžiausias dėmesys skiriamas vizualiam dizainui. Tokiu atveju dizainas niekada nebus toks geras, koks jis galėtų būti, nes nebus suderintas su svetainės galutine struktūra ir funkcijomis. Vizualus dizainas, kuris paremtas aiškia ir išsamia specifikacija, visada bus geresnis už dizainą, sukurtą prieš pilnai specifikuojant projektą.

Grafikos dizaineriai turi ne tik dekoruoti tinklalapį, bet taip pat jį padaryti kuo patogesni vartotojui. Deja, kai dauguma žmonių galvoja, jie mano, kad tai yra padaryti kažką atrodyti gražiai. Toks mastymas yra tikrai toli nuo pagrindinės dizaino sąvokos ir to, ką dizaineris turi daryti. Emil Roder yra pasakęs: „Sukurti dizainą yra planuoti, sutvarkyti, susieti ir valdyti. O jei trumpai, tai yra priešprieša visoms reikšmėms „netvarka“ ir „atsitiktinumas““. Dizainerių rūpestis yra, kaip pateikti informaciją kuo aiškesnę ir tikslesnę. Kai dizaineris kuria iš aiškios specifikacijos, ne tik gali padaryti savo darbą greičiau ir geriau, bet taip pat jam yra daug lengviau.

Kai yra pateikiamas išankstinis siūlomas dizaino planas, dažnai klientai gali reaguoti neigiamai. Ta baimė dažniausiai kyla iš to, kad klientai mano, jog kūrėjai nepilnai suprato, ką jie turi galvoje ir koks turi būti svetainės funkcionalumas. Yra pakankamai sudėtinga atskirti matomą išdėstymo vaizdą nuo savo įsivaizduojamos svetainės, būtent, ką jinai turi daryti ir kaip turi atrodyti. Kai tikimasi, jog klientas pažiūrės ir nujaus, kaip ta svetainė atrodys, realiai tai priveda prie to, jog klientai pasimeta ir jaučia, kad jie priversti atsisakyti jų pačių lūkesčių. Kūrėjai prašo jais pasitikėti ir pabandyti įsivaizduoti, kaip ta internetinė svetainė veiks, nors iš to, ką klientas mato, nėra pilnai apibrėžiamas galutinis produktas. Realiai tai yra per didelis prašymas. Be to, kūrėjai dar nėra įgavę pasitikėjimo, kadangi dizainas yra vienas iš pirmųjų pristatymų klientui projekte. Taigi, nėra pakankamai laiko ir galimybių įgyti naujo kliento pasitikėjimą. Kiekviena neigiama kliento reakcija tarsi pasėja nepasitikėjimo ir baimės sėklą tolimesniame projekto vykdyme.

Išspręsti tą problemą padėtų skiriant daugiau laiko analizuojant kliento poreikius ir planuojant svetainės strategiją dar prieš pradėdant dizaino procesą. Tik šiuo atveju yra kita

problema – per ilgas laukimas. Strategija ir planavimas gali užimti per daug laiko, todėl klientas turi per ilgai laukti, kol ką nors pamatys.

Taigi, kūrėjas yra priverstas praleisti daugiau laiko planuojant ir rizikuoti, kad užtruks per ilgai kažką parodyti klientui, arba jis turi parodyti klientui preliminarius planus, kurie nėra paremti visišku projekto supratimu.

Kita internetinių svetainių kūrimo problema yra bendravimas. George Bernard teigimu, viena didžiausių problemų bendravime yra iliuzija, kuri viską užvaldė[17]. Internetinių svetainių kūrime svarbiausia turėti yra ne pritrenkianti dizainą, mirgančią animaciją ar nusimanytą apie internetines rinkas. Vienas iš svarbiausių dalykų yra mokėti bendrauti technine informacija ne techniškai. Jei to kūrėjas negali padaryti, niekada neišpildys kliento lūkesčių.

Kai iškyla bendravimo problemos, problema yra ne tai, kad klientas nesupranta, bet tai, ką kūrėjai prašo jį suprasti. Internetinė svetainė yra techninė sistema. Kūrėjas neturi tikėtis, kad klientas viską supras be išsamaus paaiškinimo. Visgi informacijos dizainas reikalauja žinių. Reikalinga suprasti interaktyvumą, dinaminį turinį, hipertekstą, naršymo sistemą, paieškos vykdymą ir pan. Taigi kūrėjo rūpestis yra tinkamai bendrauti su klientu ir viską išsamiai paaiškinti, jog klientui nekiltų abejonių.

Kita internetinių svetainių kūrimo problema yra dokumentavimas. Internetinė svetainė yra programinė įranga, nes ji parašyta kodu ir vykdoma kompiuteryje [17]. Kuo svetainė yra dinamiškesnė, tuo didesnis skirtumas tarp kliento žinių ir produkto, kurį jis perka. Išsami dokumentacija užima nemažai laiko, o klientas techninės dokumentacijos nesupras, kadangi neturi tam reikalingų žinių. Suvokiant poreikį dokumentacijos, parašytos ne techniniais terminais, dažnai pradedama nuo svetainės žemėlapiu, sistemos veiklos diagramų, rašytinių funkcionalumo aprašymų ar istorijų. Tokie dokumentai padeda, bet jie nesuteikia galimybės bendrauti gilesniame informacijos lygyje. Taip klientai yra paliekami nežinomybėsje iki tol, kol pamato jau kuriamą sistemą. Iš tikrųjų, kūrėjai gali sukurti aiškia ir išsamią dokumentaciją, bet klientas gali nesuprasti arba tiesiog pamiršti. Naudinga yra sukurti prototipą, kad kūrėjas galėtų tinkamai išaiškinti klientui subtilybes, išskylančias problemas ir pan.

Ne visada kuriant internetines svetaines klientas turi galimybę patikrinti svetainės veikimą praktiškai. O pristačius galutinį produktą, klientas pastebį tai, kad ne viskas funkcionuoja taip, kaip jis norėjo ir tikėjosi. Tačiau daryti pakeitimus šioje stadijoje yra brangu ir tai reikalauja nemažai ir laiko sąnaudų. Taigi pristatymo data taip pat gali kisti. Taip gali augti kliento išlaidos arba netgi kūrėjas turi nuostolių.

Kai klientai peržiūri prototipą, jie žymiai daugiau dėmesio skiria į detales nei peržiūrint popierinę ar kitos formos dokumentaciją. Dažnai funkcionalumo ir naudingumo aspektų klientas neapgalvoja, nenuspėja iki tol, kol jis nepabando tinklalapio. O prašyti kliento nuspėti visus aspektus būtų neteisinga. Kai klientas praktiškai turi galimybę išbandyti svetainę, jis pagerina savo supratimą, padeda jam įsivaizduoti, kaip iš tikrųjų sistema turėtų veikti. Bet tai reikia padaryti iki tol, kad nesunku būtų padaryti pakeitimus ir neatsirastų nuostolių.

0.6.1.3 Kliento klaidos internetinių svetainių kūrime

Dizaineriai ir kūrėjai prieš kuriant tinklalapį nori žinoti kuo daugiau informacijos apie turinį. Tiksliai žinant, kokio dydžio formos bus, kokios informacijos kategorijos, kokios diagramos, kaip atrodo lentelės, kūrėjams suteikia labai geras detales ir padeda įsivaizduoti, kaip turi atrodyti vizualus, informacijos dizainas, puslapių struktūra. Tačiau klientas turi žinoti, koks turi būti turinys, kokio ilgio, koks turinio tikslas. Jie dažnai mano, jog pateikta informacija, tokia kaip lankstinukas ar keletas informacijos lapų viską sustato į savo vėžes. Tai sukelia daug bendravimo nesusipratimų, kurie turi įtakos pakeitimams ir projekto vėlavimui.

Kuriant turinį gali kilti sunkumų ir pasimetimų, iš tikrųjų dažna problema yra ir turinio pristatymas. Klientai atitinkamai nepakankamai įvertina laiką, kurio reikės turinio kūrimui ir rinkimui. Kai yra prototipas, jis tampa ne tik pagalba turinio kūrime, bet ir priemone derinant ir pristatant turinį.

Kartais klientai gali galvoti, kad internetinė svetainė yra daug lankstesnė nei ji iš tikrųjų yra. Dažniausiai tai nutinka dėl to, kad dauguma tinklalapio elementų ar savybių gali būti tiesiog įkeltos, jei to prireikia. Bet iš tiesų, ne taip lengva gali būti pakeisti svarbias informacijos dizaino problemas, kurios yra varžomos techninių galimybių.

Informacijos dizaino detalaus parengimo svarba prieš pradėdant kurti dizainą ir programuoti svetainę auga proporcingai svetainės gyliui ir pločiui [17]. Kuo didesnis puslapių skaičius svetainėje, tuo atidžiau turi būti apgalvotas informacijos dizainas. Priešingu atveju bus daug sunkiau sutvarkyti informacijos dizaino problemas nei iš pradžių, kai dar buvo apgalvojama struktūra. Informacija turėtų būti struktūrizuojama taip, kad, esant būtinybei, ją būtų galima išplėsti. Kuo geriau apgalvota struktūra, tuo mažiau darbo bus sunaikinant, perkuriant jau egzistuojančius puslapius ir struktūras ateityje.

Klientai gali pageidauti svetainės su skirtingu turiniu, t.y. skirtingi puslapiai turi skirtingus turinius, pavyzdžiui, namų puslapis turi skirtingą turinį. Jei šis reikalavimas yra pateikiamas projekto pabaigoje, tai gali sukelti dideliu pakeitimų.

0.6.2 Sistemos kūrimo metodų analizė

Programinė įranga Lietuvoje dažniausiai kuriama naudojantis metodais, kurie remiasi dokumentacijos kūrimu. Kad ir kaip išsamiai būtų dokumentuojami reikalavimai, kūrimo eigoje beveik visada atsiranda pakeitimų. Kuo pakeitimai įvyksta vėliau, tuo didesne projekto kaina, nes tai labai įtakoja darbų tvarkaraščius bei projekto apimtį.

Yra daug sistemos kūrimo metodikų. Tačiau, kokį modelį naudoti, priklauso nuo sistemos pobūdžio, kadangi skirtingi modeliai nevienodai tinka įvairaus tipo sistemoms kurti. Tinkamos metodikos pasirinkimui įtakos turi: organizacinė aplinka ir taikomųjų uždavinių rūšis 5.

Krioklio modelis

Krioklio modelis – tai klasikinis modelis. Jį tikslinga taikyti, kai:

- vartotojo reikalavimai yra aiškūs,
- uždavinys yra tradicinis,
- aiškus sprendimo algoritmas,
- uždavinys nėra sudėtingas.

3 lentelė Krioklio modelio analizė

Krioklio modelis	
Trūkumai	Privalumai
2. Nelankstus projekto skirstymas į stadijas.	1. Jei nėra daug grįžimų, nedidelės darbo sąnaudos.
3. Yra sunkumų, kai reikia taisyti klaidas arba reaguoti į vartotojo reikalavimų pasikeitimą.	2. Procesas aiškus, lengvai koordinuojamas.
4. Nėra galimybės grįžti į ankstesnes sistemos kūrimo fazes, atlikti pataisymus.	3. Galimybė suskirstyti kolektyvą į skyrius ir atlikti valdymo kontrolę.
5. Negalima vartotojams pateikti dalinio programos varianto.	
6. Eksploatacijos pradžia galima tik realizavus visus panaudojimo atvejus, kas atitinka ilgiausią kelią tinkliniame grafike.	

Krioklio modelis taikomas tik tokiais atvejais, kai reikalavimai ir jų įgyvendinimas yra labai gerai suprasti. Kūrimas prasideda nuo koncepcijos, po to pereinama prie projektavimo,

realizavimo, testavimo, įdiegimo, gedimų pašalinimo ir baigiasi eksploatacija ir palaikymu. Kiekviena kūrimo fazė vyksta pagal griežtą tvarką, jokių persidengimų ar iteracijų.

Pakartotinio panaudojimo modelis

Pakartotinio panaudojimo modelio esmė – jau esamų sistemų, posistemų ar programų panaudojimas iškilusiems uždaviniams spręsti. Tada reikia skirtingų išdirbėjų sukurtas dalis integruoti į vieningą sistemą.

4 lentelė Pakartotinio panaudojimo modelio analizė

Pakartotinio panaudojimo modelis	
Trūkumai	Privalumai
<ol style="list-style-type: none"> 1. Reikalingi specialistai, gebantys aukščiausiu lygiu įsisavinti svetimus produktus, dažnai neturint išėties kodus. 2. Reikia gerai orientuotis produktų rinkoje. 3. Sudėtinga posistemų sąveika, ne visuomet efektyvūs interfeisai tarp posistemų, skirtingi darbo su posistemėmis stiliai. 	<ol style="list-style-type: none"> 1. Išnaudojamas kitų sukurtas geriausias produktas. 2. Mažesnis PS kūrimo darbo imlumas, ji sukuriama greičiau ir pigiau.

Evoliucinis modelis

Evoliuciniame modelyje projektas vystosi nuosekliai, pradedant panaudojimo atvejo poaibio realizavimu ir nuosekliai prijungiant naujus panaudojimo atvejus arba jų grupes. Modelio esmė – darbas su užsakovu nuo pradinės specifikacijos iki galutinės sistemos versijos. PS reikalavimų supratimas auga realizavimo eigoje. Tikslinga pradėti nuo aiškių ir svarbių panaudojimo atvejų. Labai svarbus yra išdirbėjo bendravimas su vartotoju, gera psichologinė aplinka.

Pagrindinė evoliucinio modelio idėja – veikiančių kuriamos sistemų prototipų naudojimas jos dinaminėms savybėms modeliuoti.

5 lentelė Evoliucinio modelio analizė

Evoliucinis modelis	
Trūkumai	Privalumai
<ol style="list-style-type: none"> 1. “Suveltas procesas”. 2. PS gaunasi blogai struktūrizuota. 	<ol style="list-style-type: none"> 1. Atitinka mūsų “suveltą” gyvenimą. 2. Nereikalauja aukštos kvalifikacijos

<p>3. Ne visada galima įkalbėti vartotoją dirbti su nepilnai suprojektuota sistema.</p>	<p>sisteminių analistų – brangiausių specialistų.</p> <p>3. Efektyvu mažoms sistemoms, nors taip kuriamos ir didelės sistemos, kurios yra per sudėtingos, kad būtų galima jas aprėpti.</p> <p>4. Šis būdas realizuojasi natūraliai, einant per sistemos gyvavimo etapus.</p>
---	--

Šis modelis gali būti veiksmingai naudojamas tik turint prototipams konstruoti skirtas instrumentines priemones. Kadangi prototipus modifikuoti tenka daug kartų, daryti tai rankiniu būdu yra per brangu ir reikia labai daug laiko 5.

Formalusis modelis

Sistemos specifikavimas šiame modelyje traktuojamas kaip atitinkamos probleminės srities teorijos formulavimas. Teorija yra formuluojama etapais, kol pereinama prie abstrakčių formaliosios logikos sąvokų ir formuluojama formali dalykinės srities teorija. Ši teorija kartu yra ir formali kuriamosios sistemos reikalavimų specifikacija. Specifikacija nusakoma formalia dalykinės srities teorija. Kuriamoji sistema yra vienas iš tos teorijos modelių, todėl sistemos korektiškumas yra garantuotas automatiškai 5.

6 lentelė Formaliojo modelio analizė

Formalusis modelis	
Trūkumai	Privalumai
<ol style="list-style-type: none"> 1. Procesas labai formalus ir sudėtingas. 2. Sunku formaliai aprašyti sistemas. 3. Reikia aukštos kvalifikacijos specialistų. 4. Nepavyksta visko gauti vien formalių transformavimų kelių, tenka įdėti ir rankinio darbo, ypač – projektuojant “žmogus-sistema” interfeisą 	<ol style="list-style-type: none"> 1. Iš pat pradžių būtina gerai suprasti sistemos reikalavimus ir jos veikimo ypatybes. 2. Vėlesnis procesas nėra reikalaujantis daug darbo, jis greitas. 3. Gana patogi reinžinerija. 5. Tinka nedidelėms sistemoms bei sistemoms, keliančioms aukštus reikalavimus patikimumui.

Sistema yra aprašoma matematiškai. Galutinis jos variantas generuojamas formalių metodų pagalba. Tokiu būdu galima sugeneruoti tik vykdomą programą, tačiau lieka parengti dokumentaciją, “help” failus ir pan.

Žingsninis kūrimas

Sistemos kūrimas yra išskaidomas į kelis žingsnius. Kiekvieną kartą yra realizuojama dalis sistemos funkcionalumo. Prioritetai suteikiami vartotojų reikalavimams. Pirmiausia nagrinėjami aukščiausio prioriteto panaudojimo atvejai. Procesui prasidėjus, reikalavimų stengiamasi nekeisti, nors tai ne visada pavyksta.

7 lentelė Žingsninio kūrimo analizė

Žingsninis kūrimas	
Trūkumai	Privalumai
1. Atskirų dalių integravimas sukelia problemų.	1. Procesą galima koreguoti, mažėja nesėkmės rizika.
2. Po to, kai jau bus realizuota nemaža dalis sistemos funkcionalumo, gali tekti keisti architektūrą.	2. Sistema gali būti pradėta naudoti dar nerealizavus pilno funkcionalumo.
3. Sistema gali gautis neoptimali.	3. Žingsniniame realizavime įgaunamas patyrimas, kuris gali būti panaudotas kituose žingsniuose.

Spiralinis kūrimas

Spiralinis gyvavimo ciklo modelis – tai klasikinio (“krioklio”) modelio modifikacija pildant jį rizikos analize ir prototipų kūrimu 5.

Procesas vystosi kaip spiralė, o ne kaip veiksmų seka su grįžimais. Kas buvo padaryta, nėra perdaroma, tik patobulinama. Nėra fiksuotų fazių, spiralėje tikslai pasirenkami pagal poreikius. Ne visuomet vykdomas specifikuojimas.

Spiralinis kūrimas susideda iš tokių etapų:

1. Tikslų, apribojimų nustatymas (specifikavimas);
2. Rizikos nustatymas ir mažinimas;
3. Kūrimas ir atestavimas ;
4. Kitos stadijos planavimas.

8 lentelė Spiralinio kūrimo analizė

Spiralinis kūrimas	
Trūkumai	Privalumai
1. Reikalauja didelio rizikos įvertinimo. Rizika specialiai vertinama ir stengiamasi ją sumažinti.	1. Galimybė užsakovui numatyti būtinus pakeitimus, panaudoti naujas technologijas.
2. Padarytas funkcionalumas nėra perdaromas, o tik patobulinamas.	2. Galimybė užsakovui ir kūrėjui kiekvienoje kūrimo stadijoje nustatyti

3. Jei rizikos faktorių nerandama, tai tikimybė atsirasti problemos yra didelė.	ir įvertinti riziką. 3. Rizikos įvertinimas ir aptarimas kiekvienoje sistemos kūrimo stadijoje sumažina problemų skaičių;
---	--

Spiralinis gyvavimo ciklo modelis tinka dideliems ir sudėtingiems projektams. Kadangi daug dėmesio skiriama rizikos įvertinimui, šis modelis netinka mažiems ir nesudėtingiems projektams.

Agile metodai

Agile metodai akcentuoja tiesioginę komunikaciją. Reikalavimų nustatymas yra pagrįstas prototipų kūrimu. Atsiradus naujiems pakeitimams, turi būti greitai į juos reaguojama.

Agile metodai taikomi, kai reikalavimai kinta ir programų kūrimas yra sunkiai valdomas.

9 lentelė Agile metodai

Agile metodai	
Trūkumai	Privalumai
<ol style="list-style-type: none"> 1. Sudėtinga įtraukti vartotojus į kūrimo procesą ir aktyviai jame dalyvauti. 2. Sunku prognozuoti produkto kainą ir galutinio produkto pristatymo terminus. 3. Pripratus prie tradicinių metodų, sunku persiorientuoti. 4. Testuotojai reikalingi viso projekto metu, o tai padidina kaštus. 	<ol style="list-style-type: none"> 1. Orientacija į principus, ne į procesą. 2. Tiesioginė komunikacija tarp kūrėjų ir užsakovų. 3. Operatyvus reagavimas į reikalavimų pasikeitimus. 4. Pakeitimai galimi net paskutiniu momentu. 5. Mažos komandos. 6. Lankstumas.

Atsižvelgiant į internetinių svetainių kūrimo problematiką, galime daryti išvadas, jog *Agile* metodai būtų tinkamiausi internetinių svetainių kūrimui, kadangi dažnai vartotojai nėra tiksliai apsisprendę, kokias funkcijas sistema turi atlikti, todėl kūrimo eigoje jie keičia reikalavimus. Todėl reikalingas greitas reagavimas į reikalavimų pasikeitimus, lankstumas kūrimo procese.

Agile metodai siūlo kuo mažiau dėmesio skirti dokumentavimui. Programinė įranga kuriama trumpomis iteracijomis. Užsakovo reikalavimai kūrimo procese dažnai kinta, todėl, naudojantis *Agile* metodu, vartotojo poreikiai išsiaiškinami palaipsniui bendraujant tiesiogiai su užsakovu.

Kuriant sistemą, dažniausiai iš pradžių nustatoma projekto kaina, ir tada skelbiami konkursai programinei įrangai kurti pagal numatytus reikalavimus. *Agile* metodai gali būti labai efektyvūs, bet jie reikalauja visiškai kitokio programinės įrangos kūrimo proceso, jo planavimo, valdymo bei finansavimo. Pavyzdžiui, naudojantis *Agile* metodais, programinės įrangos kūrimas yra apmokamas už trumpo periodo iteracijas pagal dirbtų valandų skaičių.

Kai vartotojui pateikiamos dažnai atnaujinamo produkto versijos, jį sunku įtraukti į kūrimo procesą. Tokiais atvejais, kai galutiniai vartotojai neturėtų dirbti su tarpinėmis neišbaigtomis produkto versijomis, pavyzdžiui, kai kalbama apie kritines arba svarbias verslo valdymo sistemas (bankinės, gamyklų valdymo, apskaitos sistemos), tai ypač keblu. Be to, praktikoje sunku, o kartais ir neįmanoma, įtraukti vartotojus ir verslo atstovus į kasdienes sistemos kūrimo procesus. Tačiau *Agile* metodams yra būtina, kad užsakovas nenutrūkstamai dalyvautų kūrime.

Bendruojant gyvai galima išsiaiškinti daug detalių, tačiau nedokumentuota informacija paprastai yra įvairiai interpretuojama ir kinta ją perduodant. Greitai reaguoti į pakeitimus ir atsisakyti formalaus reikalavimų pakeitimų valdymo galima tik tuomet, kai pakeitimai yra nežymūs. Tačiau stambesni pakeitimai turi būti apsvarstomi, įvertinami, jiems turi būti suteikiami prioritetai. Dideli pakeitimai lemia ne tik vykdomo projekto apimtį, darbų tvarkaraščius, bet ir produkto išleidimo į rinką laiką.

Dėl minėtų priežasčių *Agile* metodai labiau tinkami naujiems produktams kurti bei vykdyti vidinius projektus, kur galutinius vartotojus gali pakeisti kompanijos darbuotojai. Šiuo atveju prioritetų suteikimas ir pakeitimų valdymas tampa mažiau formalus, nes yra lengviau valdyti projekto apimtį, tvarkaraščius ir biudžetą.

Projektų, kuriuose naudojami *Agile* metodai sėkmės faktoriai – vartotojo įtraukimas, efektyvus reikalavimų valdymas, komunikacija. Pagrindinės projektų nesėkmių priežastys – reikalavimų valdymo problemos.

Klientai gali pradžioje gerai nežinoti, ką jų sistema turi daryti. Galima situacija, kai reikia sukurti sistemą, kurios funkcionalumas kinta kas keletą dienų ar kelis mėnesius. T.y. kai kuriose srityse dinamiškai besikeičiantys reikalavimai yra vienintelis pastovus dalykas [14]. Naudojant *Agile* metodus yra propaguojamas reikalavimų aiškinimasis, įtraukiant vartotoją į kūrimo procesą ir pateikiant dažnas prototipo versijas, kurių dėka greitai reaguojama į reikalavimų pakeitimus. Tokios metodų savybės yra labai tinkamos internetinių svetainių kūrimui, kadangi reikalavimai gali keistis, o kaskart pateikti užsakovui peržiūrėti kuriamą produktą nėra sudėtinga. Taigi, nedidelių internetinių sistemų kūrimui, kai užsakovo reikalavimai nėra tiksliai žinomi arba laikui bėgant jie gali kisti, labiausiai tinka *Agile* metodas.

0.6.2.1 Internetinių taikomųjų programų kūrimas panaudojant Agile

Internetas pakeitė programinės įrangos kūrimo pagrindinius prioritetus iš „ką“ į „kada“ (from what to when) [6]. Rinka reikalauja ne tik kuo greičiau pristatyti programinę įrangą, bet dar ir aukštesnės kokybės bei didesnio funkcionalumo. Be to kūrimo metu turi būti greitai įvykdomi pakeitimai. Vikrumas programinės įrangos kūrime reiškia ne tik greitą produktų pristatymą, bet ir greitą prisitaikymą prie besikeičiančių reikalavimų. Kad procesas būtų vikrus (*Agile*), jis turi būti pakankamai lankstus, kad sklandžiai ir laiku prisitaikytų prie reikalavimų pasikeitimo.

Dabartinių internetinių sistemų kūrimą tampa vis sunkiau ir sunkiau suvaldyti. Dauguma kūrėjų galvoja, kad internetinių taikomųjų programų kūrimas yra tiesiog paprastas internetinis puslapis, sukurtas naudojant HTML ar internetinių svetainių kūrimo įrangą, tokią kaip Front Page ar Dreamweaver [7]. Daugumos internetinių programinių įrangų reikalavimai keičiasi. Yra apimamas planavimas, architektūra, sistemos dizainas, kokybę užtikrinantys pasikeitimai, nuolatinis atnaujinimas, sistemos priežiūra kai reikalavimai kinta.

Per paskutinius trisdešimt metų buvo pristatyta daug programinės įrangos procesų ir metodologijų, kurie padeda susitvarkyti su problemomis, susijusiomis su programinės įrangos inžineriniais projektais [2]. Internetinių svetainių kūrimo praktiniai niuansai:

- Pradėti nuo vartotojo sąsajos. Pirmiausia sukurti vartotojo sąsają, o vėliau rūpintis įgyvendinimu, nes kodo architektūra ir duomenų modelis turi palaikyti vartotojo dizainą, o ne atvirkščiai.
- Greitai daryti iteracijas. Tai padeda įgyvendinimo procesui.
- Naudoti, kol dar kuriate. Visi komandos nariai kūrimo metu turi naudoti programinę įrangą. Jie yra geriausi prototipo testuotojai.
- Daryti greitas iteracijas. Leisti vartotojams naudoti programinę įrangą, nors jiniai dar vis yra kuriama. Taip galima pamatyti realius elgesio pavyzdžius ir pataisyti kritines vietas [9].

Yra daug metodologijų, skirtų kurti sistemoms, bet internetinėms sistemoms labiausiai tinka *Agile* metodai, nes *Agile* procesai yra paremtas:

- pakeitimų galimybe;
- žmonėmis (orientuoti į žmones);
- mažomis kūrėjų komandomis;
- trumpais kūrimo ciklais [2].

Galutiniams vartotojams yra svarbus taikomųjų programų dizainas, tačiau šiaip ar taip viena yra sukurti internetinę svetainę, o kita – reikia suprasti, ko reikia galutiniam vartotojui. Internetinė inžinerija yra pakankamai komplikauta pridedant papildomus kuriamus dizaino modelius. Problema atsispindi estetiniuose vartotojo sąsajos aspektuose [2].

Toliau esanti citata iš Manifesto programinės įrangos kūrimui pateikia tikslią santrauką:

„Mes atskleidžiame geresnius programinės įrangos kūrimo būdus tai darydami ir padėdami kitiems tai padaryti. Dirbdami priėjome prie išvadų:

- Individualumas ir sąveika per procesus ir įrankius.
- Veikianti programinė įranga per išsamią dokumentaciją.
- Užsakovų bendradarbiavimas pagal kontrakto sąlygas.
- Atsakas į pasikeitimus vykdant planą.“ [13]

Apskritai inžineriniuose projektuose pagrindinis faktorius, nulemiantis sėkmę ar nesėkmę internetinių taikomųjų programų kūrime yra ir kūrėjai, ir organizacijos. Todėl *Agile* pasirinktas kelias su orientavimusi į žmones bei individualumu į sąveika per procesus ir įrankius stipriai priartina metodą prie internetinio kūrimo.

0.6.2.2 Agile metodologija

Agile metodologija yra metodų visuma, skirta minimizuotų pasikeitimų kainą, ypatingai ten, kur svarbūs faktai paaiškėja tik baigiantis projektui, arba kai reikia prisitaikyti prie svarbių nekontroliuojamų veiksnių [11].

2001 m. paskelbtas „Agile Software Development“ manifestas. Jo metu buvo suformuota dvylika *Agile* programinės įrangos kūrimo principų:

- Didžiausias prioritetas yra patenkinti užsakovo reikalavimus per ankstyvą ir vėlyvesnius programinės įrangos pristatymus.
- Priimtini besikeičiantys reikalavimai netgi vėlyvajam kūrime. *Agile* procesas kompetentingiems užsakovo norams suteikia pirmenybę.
- Dažnai pristatyti veikiančią programinę įrangą, nuo dviejų savaitių iki dviejų mėnesių, pirmenybė teikiant trumpesniems laiko intervalams.
- Verslo žmonės ir kūrėjai turi dirbti kartu vykstant projektui ištiesai kiekvieną dieną.
- Projektas kuriamas motyvuojant individualiai. Reikia suteikti aplinką ir palaikymą, kurio jiems reikia, ir pasitikėti, kad darbas bus įvykdytas.

- Veiksmingiausias ir efektyviausias informacijos perteikimo metodas kūrimo komandai ir tarp komandos narių yra pokalbis akis į akį.
- Veikianti programinė įranga yra pagrindinis progreso matas.
- *Agile* metodai skatina nepertraukiamą kūrimą. Finansuotojai, kūrėjai ir vartotojai privalo kaskart stebėti nuolatinius žingsnius.
- Nepertraukiamas dėmesys techniniam meistriškumui ir geram dizainui pagerina judrumą.
- Paprastumas – menas maksimaliai išnaudoti nepadarytą darbą – yra svarbus.
- Geriausia architektūra, reikalavimai ir dizainas atsiranda iš autonominių komandų.
- Reguliariuose intervaluose komandos apgalvoja, kaip padaryti dar efektyviau, tada darnumas ir prisitaikymas funkcionuoja atitinkamai.[12]“

Agile metodai:

- [Extreme Programming](#) (XP)
- [Scrum](#)
- [Agile Modeling](#)
- [Adaptive Software Development](#) (ASD)
- [Crystal Clear](#) and Other Crystal Methodologies
- [Dynamic Systems Development Method](#) (DSDM)
- [Feature Driven Development](#) (FDD)
- Lean Software Development
- [Agile Unified Process](#) (AUP)
- Agile Database Techniques (AD)
- Test-Driven Design (TDD)
- XBreed

Agile metodai vertinami kaip adaptyvūs, o ne kaip prognozuojantys, orientuoti į žmones, o ne į procesą [10].

Metodas yra *Agile*, jei sistemos kūrimas yra :

- laipsniškas (nedideli PĮ leidiniai, trumpi kūrimo ciklai);
- kooperatyvus (užsakovas ir kūrėjas pastoviai dirba kartu ir glaudžiai bendrauja);
- nesudėtingas (metodą lengva išmokti, keisti; gerai dokumentuotas);
- adaptyvus (pakeitimai galimi net paskutiniu momentu) [10].

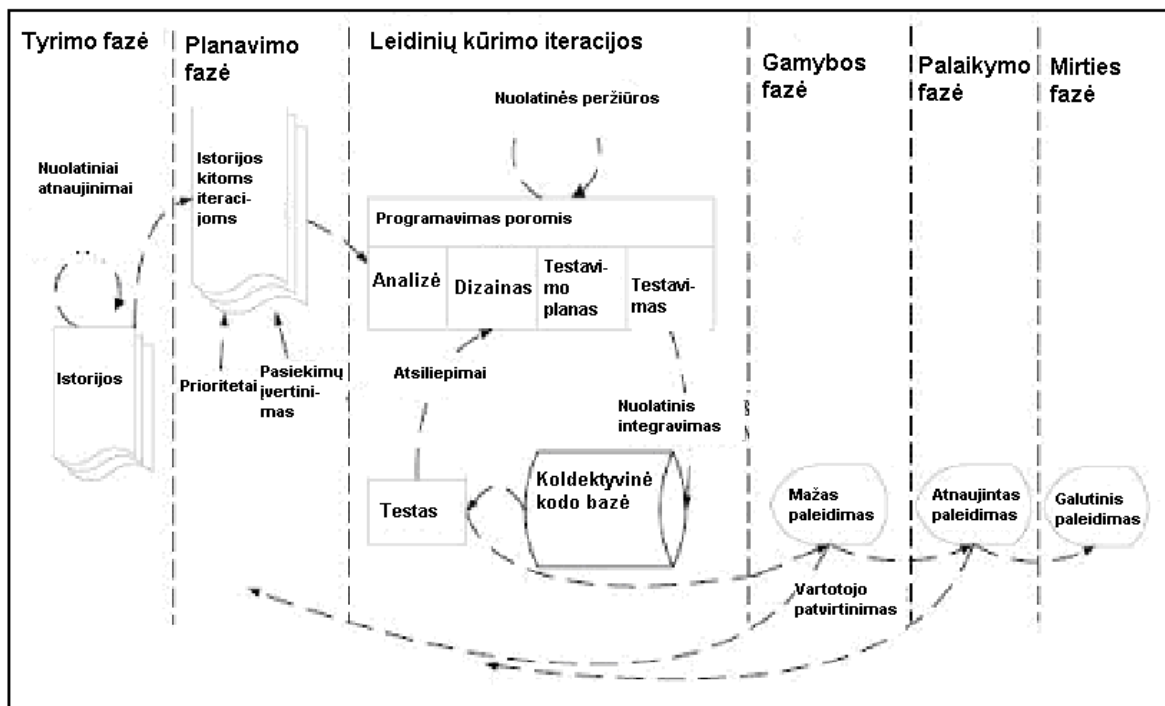
0.6.2.3 Agile metodų analizė

Extreme Programming (XP) metodas

Autorius Kent Beck, 1999 m.

XP komandos narių vaidmenys: programuotojas, užsakovas, testuotojas, seklys, treneris, konsultantas, vadovas.

Šis metodas nėra universalus.



3 pav. Extreme Programming (XP) metodo schema 5

10 lentelė XP analizė

XP savybė	Tinkamumas internetinėms svetainėms kurti
Labiausiai tinka mažoms komandoms.	Tiktų svetainių kūrimui.
Reikalingas betarpiškas bendradarbiavimas tarp komandos narių.	Internetinių svetainių kūrime nereikalinga visiems komandos nariams išmanyti kito darbą.
Naudojantis XP metodu, labai svarbus yra testų rašymas. Pirmiausia realizuojamas testas, o po to kodas. Testuojama yra nuolatos.	Ši savybė visiškai netinkama internetinių svetainių kūrime.
XP vienas iš praktikuojamų dalykų yra programavimas poromis, kai du žmonės vienu metu rašo kodą prie vieno kompiuterio.	Tai nereikalinga kuriant internetines sistemas.
XP propaguoja tai, kad visi komandos nariai galėtų daryti pakeitimus, t.y. bet kuris	Internetinių svetainių kūrimui labiausiai tinka pasiskirstymas darbais, kur

komandos narys bet kada gali pakeisti bet kurią kodo dalį.	kiekvienas komandos narys turi savo užduotis ir jas vykdo.
Sistemos funkcionalumas apibrėžiamas iš pasakojimų, kaip sistema turi veikti. Vėliau tai naudojama užsakovo ir programuotojo.	Tinkama savybė.

Dynamic System Development Method (DSDM)

Pasiūlytas 1994 m. kaip greitojo taikomųjų programų kūrimo karkasas (angl. Framework).

11 lentelė DSDM analizė

DSDM savybė	Tinkamumas internetinėms svetainėms kurti
Pagrindinė DSDM idėja yra ta, jog vietoje to, kad pirmiau fiksuoti produkto funkcionalumo apimtį ir po to skaičiuoti kiek tam funkcionalumui realizuoti reikės laiko ir kitų išteklių, verčiau pirmiau fiksuoti laiką bei kitus išteklius ir pagal juos priderinti sistemos funkcionalumo apimtį. [8]	Visiškai netinkama internetinių svetainių kūrimui, nes kuriant svetainę svarbiausia yra, kad ji atitiktų reikiama funkcionalumą. Ir pagal tai turėtų būti skaičiuojami laiko ir kiti ištekliai.

Savybių vairuojamas kūrimo procesas (angl. Feature Driven Development)

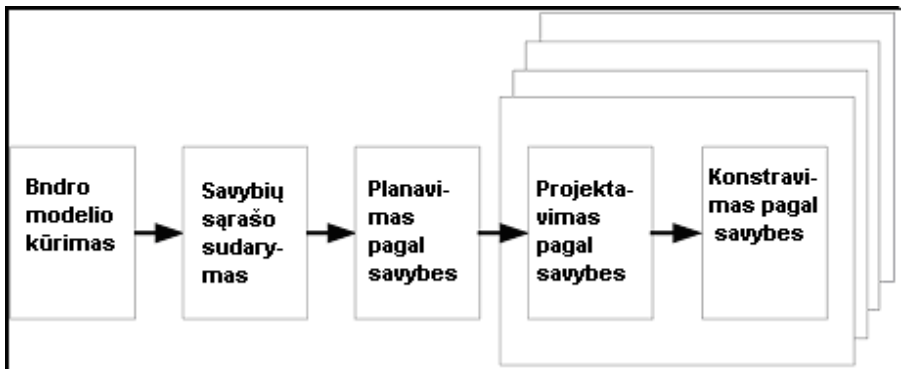
Tai yra pakankamai naujas metodas, sukurtas apie 2000 m. Jo autorius yra Jeff De Luca.

FDD proceso žingsniai: bendro modelio kūrimas, savybių sąrašo sudarymas, planavimas pagal savybes, projektavimas ir konstravimas pagal savybes.

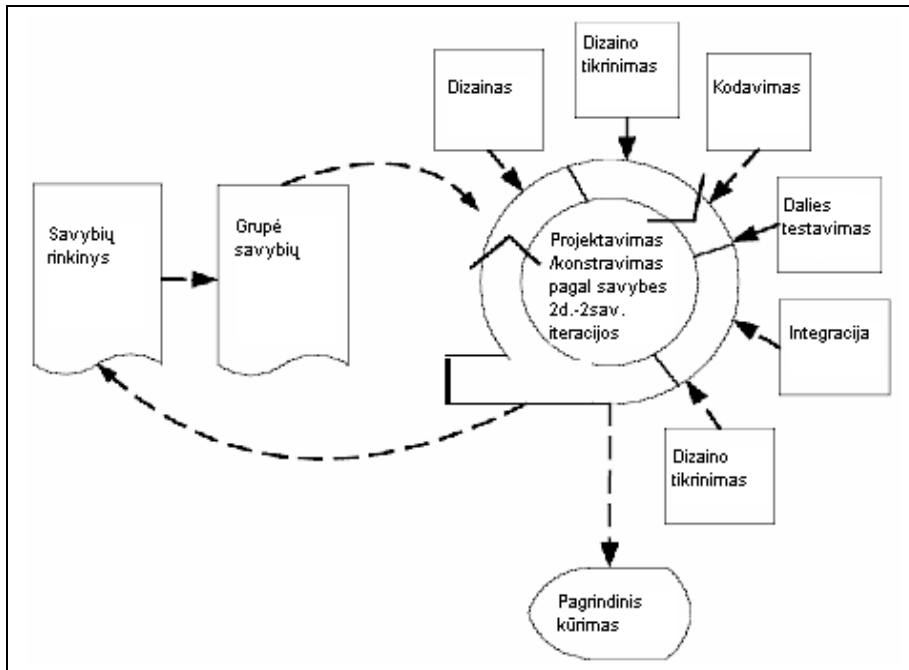
FDD labiausiai tinkamas veiklai kritinėms programų sistemoms kurti (jei reikia jas sukurti laiku). [10]

Metodas iš esmės apima tik sistemos projektavimo ir konstravimo etapus.

FDD proceso etapai: bendro modelio kūrimas, savybių sąrašo sudarymas, planavimas pagal savybes, projektavimas ir konstravimas pagal savybes.



4 pav. FDD proceso schema 5



5 pav. FDD proceso ciklas 5

12 lentelė FDD analizė

FDD savybė	Tinkamumas internetinėms svetainėms kurti
Iš pradžių kuriamas visos sistemos modelis, kad srities ekspertas suprastų sistemos apimtį, kontekstą, reikalavimus. Modelis yra suskirstomas į sritis ir komanda grupelėmis kuria objektų modelius.	Tinkama svetainėms kurti.
Sudaromas visapusiškas savybių sąrašas, kuris yra vis tikslinamas.	Ne visada reikalinga aprašyti visas savybes išsamiai.
Už klasę atsakingas tik vienas asmuo.	Daugumoje atvejų tinka.
Tinkamas dideliems projektams.	Ne visada tinkamas metodas kuriant

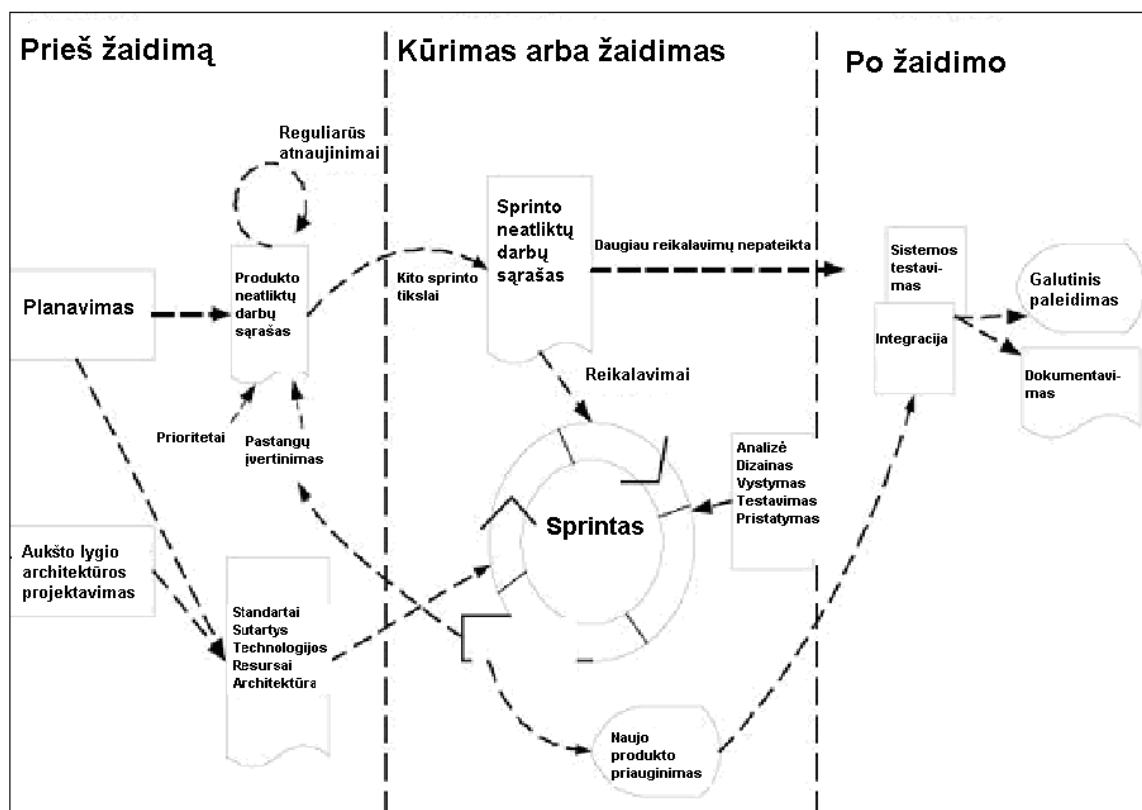
	mažesnes svetaines.
FDD neapima reikalavimų surinkimo, vartotojo sąsajos dizaino, testavimo ir išdėstymo.	Vartotojo sąsajos dizainas, testavimas ir išdėstymas yra svarbios internetinių svetainių kūrimo proceso dalys, kurios turėtų būti aprašytos metode, kuris taikomas internetinių svetainių kūrimo.

Scrum metodas

Scrum metodo idėja paskelbta 1987 m., pats metodas aprašytas 2002 m.

Sistemos realizavimo priemonėms nekeliama apribojimai. Pagrindinis uždavinys – veiklos organizavimas, kad besikeičiančių reikalavimų aplinkoje būtų lanksčiai kuriama sistema.

Scrum pagrindinė idėja – įtraukiant naujas greito valdymo veiklas, kurių paskirtis – pastoviai identifikuoti trūkumus ir kliūtis kūrimo procese ir naudojamose priemonėse. Kūrimo metu gali kisti reikalavimai, todėl tuo pat metu gali pasikeisti ir programos kūrimo laikas, ištekliai, naudojamos technologijos.



6 pav. Scrum proceso schema 5

Schemoje pateikti 3 *Scrum* proceso etapai: prieš žaidimą, kūrimas arba žaidimas, po žaidimo.

13 lentelė *Scrum* analizė

Scrum savybė	Tinkamumas internetinėms svetainėms kurti
Vykiant <i>Scrum</i> metodą, yra akcentuojami neatliktų darbų sąrašai, pastangų įvertinimai, kasdieniai susirinkimai.	Tai nėra reikalinga kuriant internetines svetaines, kadangi gaištamas laikas.
Lankstumo, adaptyvumo ir produktyvumo idėja.	Tinkama savybė.
Kintantys reikalavimai yra priimtini.	Tinka.
Nuolatinis trūkumų ir kliūčių kūrimo procese identifikavimas.	Reikalinga. Tai turėtų būti atliekama kiekvienos iteracijos metu pasitariant su užsakovu.
Kasdieniniai susirinkimai	Netinka. Kuriant internetines svetaines nėra reikalingi dažni susitikimai.
Neatliktų darbų sąrašai	Netinka.

Agile metodų savybių apibendrinimas pateiktas 14 lentelėje.

14 lentelė *Agile* metodų savybių apibendrinimas

Metodas	Išskirtinės savybės	Tinkamumas
XP	Programavimas poromis.	–
	Pirmiausia realizuojamas testas, o po to kodas.	–
	Kolektyvinė kodo nuosavybė.	+ –
	Kasdieniai produktai	–
DSDM (Dynamic System Development Method)	Pirmiausia fiksuojamas laikas bei kiti ištekliai, ir pagal juos priderinama sistemos funkcionalumo apimtis.	–
	Prototipų kūrimas	+
FDD (Feature Driven Development)	Pirmiausia sukuriamas visos sistemos modelis	+
	Sudaromas visapusiškas savybių sąrašas	+ –
	FDD neapima reikalavimų surinkimo, vartotojo sąsajos dizaino, testavimo ir išdėstymo.	–
	Labai trumpos iteracijos (nuo 1 val. iki 2 sav.)	+

<i>Scrum</i>	Pastoviai identifikuoja trūkumus ir kliūtis kūrimo procese ir naudojamose priemonėse.	+
	Kasdieniai susirinkimai	–
	Neatliktų darbų sąrašai	+–
	Susirinkimas sprintui suplanuoti	+
	Iteracijos trukmė yra 2-4 sav.	–

Agile metodų procesų etapų lentelė pateikta 15 lentelė *Agile* metodų procesai

15 lentelė *Agile* metodų procesai

Metodas	Proceso etapų skaičius	Proceso etapai	Komentariai
XP	6	Tyrimas, planavimas, leidinių kūrimo iteracijos, gamyba, palaikymas, mirtis.	Prieinamas, tačiau detalus metodo aprašas prieinamas tik konsorciumo nariams.
DSDM (Dynamic System Development Method)	7	Prieš-projektinis etapas, galimybių studija, verslo (veiklos) studija, funkcinio modelio iteracija, projektavimo ir konstravimo iteracija, realizavimas, po-projektinė studija.	Praktikuojami dalykai tinka daugelyje situacijų, tačiau bendram vaizdai ir valdymui skirta mažai dėmesio.
ASD (Adaptive Software Development)	3	Spėlioti, bendradarbiauti, mokytis.	Kalba daugiau apie sąvokas ir kultūrą, nei apie PĮ kūrimo praktiką
FDD (Feature Driven Development)	5	Bendro modelio kūrimas, savybių sąrašo sudarymas, planavimas pagal savybes, projektavimas ir konstravimas pagal savybes.	Metodas yra paprastas, sistema projektuojama ir realizuojama pagal savybes. Skirtas tik projektavimui ir realizavimui. Reikalingi kiti palaikantys metodai (požiūriai).
<i>Scrum</i>	3	Prieš žaidimą, kūrimas arba žaidimas, po žaidimo.	Detalizuoja kaip valdyti 30-dienų <i>Scrum</i> ciklą, bet nedetalizuoja integravimo

			ir priėmimo testų.
--	--	--	--------------------

Internetinių svetainių kūrimui labiausiai tiktų FDD ir *Scrum* metodai. Tačiau efektyviam internetinių svetainių kūrimui reikėtų vadovautis kitais prioritetais, tokiais kaip greitas kūrimas, skiriant kuo mažiau dėmesio į nuveiktų darbų ataskaitas, nereikalingi kasdieniai susitikimai. Bendravimas tarp komandos narių yra svarbus, o jo įgyvendinimo būdai turėtų būti išplėsti. Internetinių svetainių kūrime yra labai svarbus dizainas, taigi tam taip pat turėtų būti skiriama daugiau dėmesio.

Iš *Agile* metodų savybių apibendrinimo matome, kad FDD neapima reikalavimų surinkimo, vartotojo sąsajos dizaino, testavimo ir išdėstymo. Metodas apima tik projektavimo ir realizavimo etapus. Norint taikyti šį metodą, reikalinga remtis ir kitais požiūriais. Šis metodas labiausiai tinkamas kritinėms sistemoms kurti, kai reikalinga jas sukurti laiku. Tam turi būti išsamiai aprašomos visos sistemos savybės. Kuriant internetines svetaines dažniausiai nėra būtina išsamiai aprašyti visas savybes.

Scrum metodas propaguoja kasdienes susirinkimus, neatliktų darbų sąrašus. Internetinių svetainių kūrimas turi būti greitas, o per dažnas susirinkimų skaičius, neatliktų darbų sąrašai lėtina procesą. *Scrum* iteracijos trukmė yra 2-4 savaitės. Internetinių svetainių kūrime ilgesnė turėtų būti pirmoji iteracija, o kitos iteracijos trumpos (iki 1 savaitės).

Galime daryti išvadą, kad internetinių svetainių kūrimui naudinga sukurti atskirą *Agile* metodologiją, paremtą kai kuriomis FDD ir *Scrum* savybėmis.

FDD savybės, tinkamos internetinių svetainių kūrimui:

- Pirmiausia sukuriamas visos sistemos modelis;
- Sudaromas visapusiškas savybių sąrašas – panaudojama tik dalinai;
- Labai trumpos iteracijos (nuo 1 val. iki 2 sav.);

Scrum savybės, tinkamos internetinių svetainių kūrimui:

- Pastoviai identifikuoja trūkumus ir kliūtis kūrimo procese ir naudojamose priemonėse;
- Neatliktų darbų sąrašai – panaudojama tik dalinai;
- Susirinkimas iteracijai suplanuoti;
- Savarankiška komanda;
- Lankstumo, adaptyvumo ir produktyvumo idėja;
- Nuolatinis trūkumų ir kliūčių kūrimo procese identifikavimas;

- Reikalavimai, dizainas, kodavimas, testavimas – visada daroma visko po truputį;

Metodas, skirtas internetinių svetainių kūrimui, turi pasižymėti greitu kūrimu, kadangi vartotojai visada nori kuo greičiau pamatyti užbaigtą produktą. Esant reikalavimų pasikeitimams, į juos reikia reaguoti efektyviai. Turi būti organizuoti komandos darbas, taip, kad visi atlikti pakeitimai būtų įgyvendinti tinkamai. Komanda turi būti savarankiška ir bendradarbiauti tarpusavyje, iškilus nesklandumams – kuo greičiau spręsti susidariusias problemas, padėti komandos nariams.

Kuriant pagal *Agile* metodą, užsakovas visada turi noriai ir nuolat bendradarbiauti su komanda. Iškilus bet kokiems neaiškumams, užsakovas turi atsakyti į iškilusius klausimus.

Kuriant internetinę svetainę, dėmesys turi būti skiriamas ir dizainui, ir svetainės funkcionalumui. Vartotojui yra svarbu, kad dizainas būtų patogus, įdomus ir aiškus, tačiau jei svetainė bus ideali dizaino atžvilgiu, bet neatliks reikiamų funkcijų, vartotojams ji taps neįdomi. Vadinasi internetinė svetainė turi pasižymėti ir gera išvaizda, ir funkcionalumu.

0.7 Analizės išvados

1. Išanalizavus sistemų kūrimo modelius ir metodus nustatyta, kad *Agile* požiūris yra vienas iš tinkamiausių efektyviam internetinių svetainių kūrimui.
2. Atlikta analizė parodė, kad internetinių svetainių kūrime FDD ir *Scrum* metodai nėra visiškai tinkami naudoti tokie, kokie jie yra, todėl reikalinga sukurti metodą, paremtą FDD ir *Scrum* metodų savybėmis, kuris pagerintų internetinių svetainių kūrimo procesą.

Metodas turi pasižymėti tokiomis savybėmis:

- Greitas kūrimas;
- Lankstumas;
- Efektyvus reagavimas į reikalavimų pasikeitimus;
- Nuolatinis bendradarbiavimas su užsakovu;
- Efektyvus bendravimas tarp komandos narių;
- Dėmesys vartotojo sąsajai;
- Dėmesys funkcionalumui;
- MVC architektūros naudojimas;
- Trumpos iteracijos;

1 Metodo, skirto internetinių svetainių kūrimui, reikalavimų specifikacija

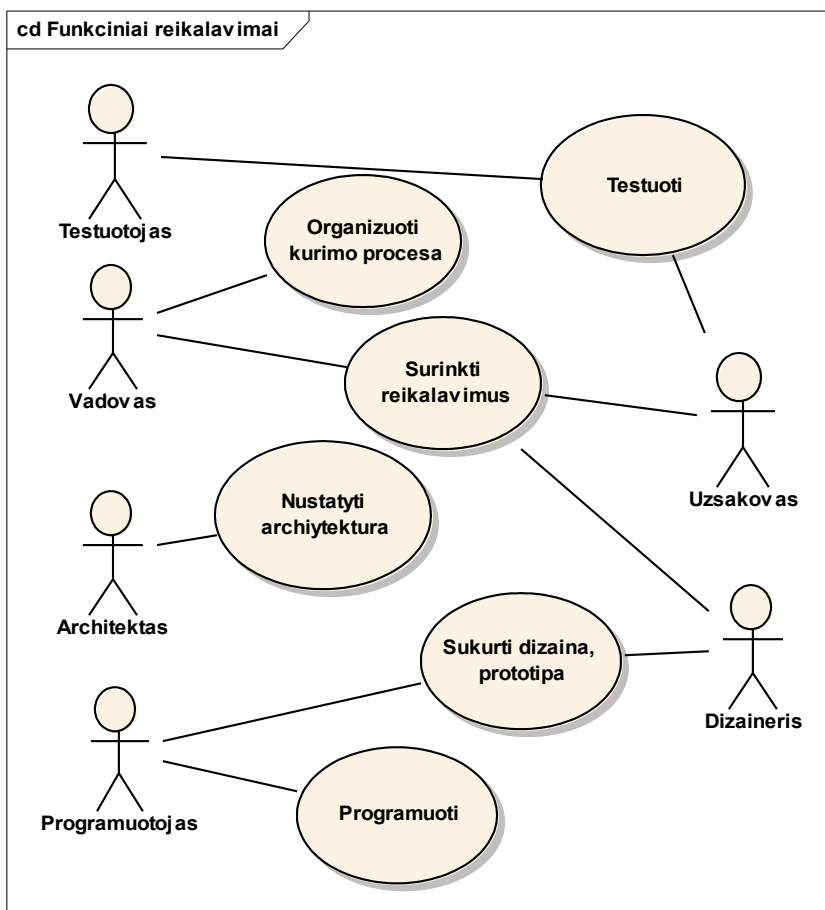
1.1 Siūlymo taikymo sritis, sąlygos ir prielaidos

Kuriant internetines sistemas dažnai iškyla problemų, kurios atsiranda dėl netinkamos kūrimo metodikos pasirinkimo. Dažniausiai užsakovai kūrimo eigoje nori keisti reikalavimus, todėl tokiu atveju tinkamiausias būtų *Agile* požiūris. Siūlomas metodas yra skirtas internetinių svetainių kūrimui. Metodas efektyviai leidžia reaguoti į reikalavimų pasikeitimus, bet tam reikalingas nepertraukiamas užsakovo bendradarbiavimas.

1.2 Funkciniai ir nefunkciniai reikalavimai

1.2.1 Funkciniai reikalavimai

Funkciniai metodo reikalavimai pateikti 7 pav.



7 pav. Funkciniai metodo reikalavimai

1.2.2 Nefunkciniai reikalavimai

Norint naudoti metodą, skirtą internetinių svetainių kūrimui, užsakovas turi būti visada „šalia“. Pavyzdžiui, prireikus patikslinti reikalavimus, užsakovas turi nedelsiant tai padaryti,

prireikus informacijos – ją suteikti ir pan. Pagrindinis reikalavimas užsakovui – nuolat teikti informaciją, reikalingą svetainės kūrimui.

Kūrėjų kvalifikacijai ypatingų reikalavimų nėra keliama. Jei internetinės svetainės kūrimą atlieka vienas asmuo, jis turi išmanyti ir architektūrą, ir dizainą, ir programavimą, ir testavimą bei turėti vadovaujančių savybių, kad galėtų tinkamai organizuoti ir vykdyti kūrimo procesą. Komandos narių skaičius neturėtų būti didelis – nuo vieno iki keleto narių.

1.3 Duomenų modelis

Funkciniams reikalavimams aprašyti yra naudojama panaudojimo atvejų diagrama, veiklos diagrama. Kiekvienam panaudojimo atvejui išsamesni reikalavimai gali būti pateikti tekstu, lentelėmis ir pan. Duomenų bazės schema gali būti pavaizduota klasių diagrama.

Nefunkciniai reikalavimai aprašomi tekstu.

Vartotojo sąsajos specifikacijai apibrėžti turi būti pateikta sistemos lango struktūra. Kiti reikalavimai pateikiami tekstu.

Testavimui turi būti pateikiama lentelė, kur surašomos rastos klaidos. Lentelės pavyzdys pateiktas 20 lentelė Testavimo rezultatai.

2 Metodo, skirto internetinių svetainių kūrimui, projektas

Projekto tikslas – parengti ir pademonstruoti metodiką internetinių svetainių kūrimui panaudojant *Agile* metodologiją.

Internetinių svetainių kūrimui patogiu taikyti *Agile* metodologiją, kadangi užsakovo reikalavimai gali kisti – kai yra nerami, neaiški aplinka.

2.1 Kada taikyti metodą?

Metodas skirtas internetinių svetainių kūrimui, kai užsakovo reikalavimai gali kisti. Svarbų vaidmenį kūrimo procese atlieka tiek kūrėjas tiek užsakovas, todėl ypač svarbus nenutrūkstamas bendradarbiavimas. Tai turi didelę įtaką produkto sėkmei. Užsakovas nesunkiai gali būti įtraukiamas į internetinės svetainės kūrimą, kadangi kiekvienos iteracijos rezultatus jis gali matyti internete.

Metodo taikymui būtinos sąlygos:

- Vartotojo reikalavimai gali kisti.
- Reikalingas nuolatinis vartotojo dalyvavimas kūrimo procese.
- Maža kūrimo komanda (nuo vieno iki keleto narių).
- Techninė architektūra yra parinkta ir aiški.

2.2 Pagrindinė idėja

Pirmiausia yra nustatoma svetainės funkcionalumo apimtis, o tada atitinkamai skaičiuojamas sugaištamas laikas ir kiti išteklių. Svarbiausia yra tai, kad proceso metu gali pasikeisti reikalavimai, tačiau atitinkamai keisis ir išteklių. Dėl šios priežasties kūrimo procesas tampa neprognozuotinas ir sudėtingas, reikalaujantis lankstumo, kad būtų galima tinkamai reaguoti į pasikeitimus ir sukurti naudingą sistemą. Yra naudojamos veiklos, kad pastoviai būtų galima identifikuoti trūkumus ir kliūtis kūrimo procese ir naudojamos priemonės.

2.3 Principai

- Būtinai aktyvus vartotojo įtraukimas;
- Komanda turi būti įgalinta priimti sprendimus;
- Pirmiausia sukuriama pradinis produktas, o vėliau daromi pataisymai, pridedamos naujos savybės;
- Didesnis dėmesys dizainui;
- Iteratyvus ir priauginantis kūrimo procesas yra būtinas siekiant tikslaus sprendimo;

- Testavimas integruotas kiekvienoje iteracijoje;
- Bendravimas ir bendradarbiavimas visų projekto dalyvių yra esminis.

2.4 Metodo savybės

- Priauginantys kūrimo ciklai (iteracijos);
- Pirmasis kūrimo ciklas užtrunka ilgiau, kol įvykdomi pradiniai užsakovo reikalavimai;
- Išryškintas bendravimas ir bendradarbiavimas tarp komandos narių ir užsakovo;
- Tiesioginis užsakovo įtraukimas;
- Neriboja kūrimo priemonių, įrankių, įpročių;
- Svarbiausia sukurtas produktas, o ne dokumentacija;
- Taikoma lankstumo, adaptyvumo ir produktyvumo idėja;

2.5 Komanda

Komandą gali sudaryti nuo vieno iki keleto narių. Narių skaičius priklauso nuo internetinės sistemos dydžio: kuo didesnė sistema – tuo daugiau narių. Išskiriami vaidmenys, tokie kaip programuotojas, dizaineris, architektas, testuotojas. Komanda pasidalina darbais sistemą išskirstydami į posistemas. Kiekvienam nariui tenka viena arba kelios posistemos.

Testavimas atliekamas komandos narių. Į internetinės sistemos testavimą iš dalies yra įtraukiamas ir užsakovas, kuris testuoja svetainės funkcionalumą, gali pateikti papildomų reikalavimų. Užsakovas arba jo patikėtas asmuo, kuris žino, ko reikia sistemai, testuoja svetainę kaip vartotojas. Komandos nariai glaudžiai bendradarbiauja su užsakovu ir vykdo jo reikalavimus sistemai.

Komandos nariai ir jų vaidmenys:

- Užsakovas
 - dalyvauja užduotyse susijusiose su produkto kūrimo darbais: testuoja svetainę, praneša apie testavimo rezultatus komandai, nusprendžia kada reikalavimai yra patenkinti, suteikia reikalavimams prioritetus;
- Komanda
 - atsakinga už sprendimus dėl būtinų veiksmų ir organizavimosi siekiant tikslų. Testuoja, kuria kuo paprastesnę bei aiškesnę programos kodą, svetainės dizainą. Komandos nariai turi bendrauti tarpusavyje.
 - Komandą sudaro:
 - Dizaineris;
 - Architektas;

- Programuotojas;
- Testuotojas.
- Vadovas
 - priima sprendimus, atsakingas už projektą, valdymą, neatliktų darbų sąrašo kontrolę ir pateikimą. Bendrauja su projekto komanda, dalyvauja nustatant tikslus ir reikalavimus;

Nebūtinai visiems vaidmenims reikia skirti atskirą žmogų. Esant nedidelei sistemai, vienas asmuo gali turėti keletą vaidmenų.

Komandos nariai tarpusavyje turi nuolatos bendrauti (jei komandą sudaro daugiau nei vienas narys). Bendravimas gali vykti internetu, telefonu, tačiau bent kartą per savaitę turi būti suorganizuotas bendras susitikimas.

16 lentelė Vaidmenų atsakomybės

Vaidmuo	Atsakomybė
Užsakovas	Pateikia svetainės reikalavimus, dalyvauja užduotyse susijusiose su produkto kūrimo darbais: testuoja svetainę, praneša apie testavimo rezultatus vadovui, nusprendžia kada reikalavimai yra patenkinti, suteikia reikalavimams prioritetus.
Dizaineris	Atsakingas už svetainės dizainą. Bendradarbiavimas su užsakovu, sukuriama vartotojo sąsaja.
Architektas	Apsprendžia tai, kaip komponentai bendrauja tarpusavyje, kaip architektūros lygiai pasiskirsto techniniuose įrenginiuose. Jis priima sprendimus, susijusius su svetainės architektūra, struktūra ir pan.
Programuotojas	Kuria kuo paprastesnį bei aiškesnį programos kodą.
Testuotojas	Testuoja svetainę prieš pateikiant svetainės versiją užsakovo testavimui.
Vadovas	Priima sprendimus, atsakingas už projektą, valdymą, neatliktų darbų sąrašo kontrolę ir pateikimą. Bendrauja su projekto komanda, dalyvauja nustatant tikslus ir reikalavimus.

2.6 Kūrimo procesas

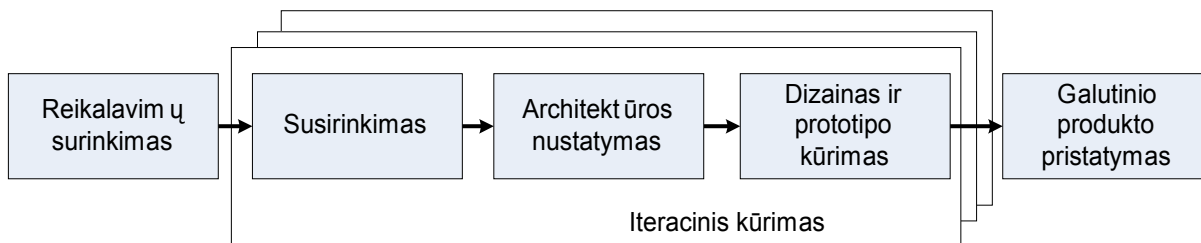
Kūrimo procesas prasideda nuo užsakovo reikalavimų surinkimo ir apibendrinimo. Reikalavimai nėra išsamiai dokumentuojami. Pagrindinis tikslas – aiškiai apibrėžti pagrindinius reikalavimus.

Projekto vykdymo pradžioje yra paskirstomi darbai. Sistema išskirstomai į posistemas. Kiekvienam komandos nariui tenka viena arba kelios posistemas. Jei komandoje yra tik vienas asmuo, visi darbai atitenka jam. Aptariami architektūriniai sprendimai

Pirmiausia yra realizuojami pirminiai užsakovo reikalavimai, t.y. sukuriamas prototipas, kuris leistų užsakovui įvertinti svetainės funkcionalumą. Daugiausia laiko

reikalinga pirmajai iteracijai. Svetainė yra ištestuojama kūrėjų ir tik tada sistemą testuoja užsakovas ir pateikia tolimesnius reikalavimus sistemai. Vėlesnės iteracijos yra trumpesnės; Vis dažniau užsakovas yra įtraukiamas į svetainės kūrimą. Iteracijos atliekamos tol, kol užsakovas daugiau reikalavimų nebepateikia. Galutinis produktas yra pristatomas užsakovui.

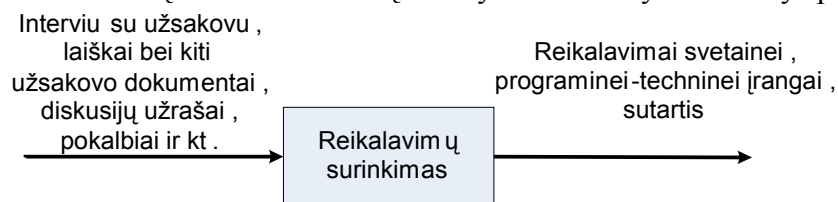
Projekto vykdymo schema pateikta 8 pav. Projekto schema.



8 pav. Projekto schema

2.6.1 Reikalavimų surinkimas

Reikalavimų surinkimo metu įeinantys ir išeinantys duomenys pateikti 9 pav.



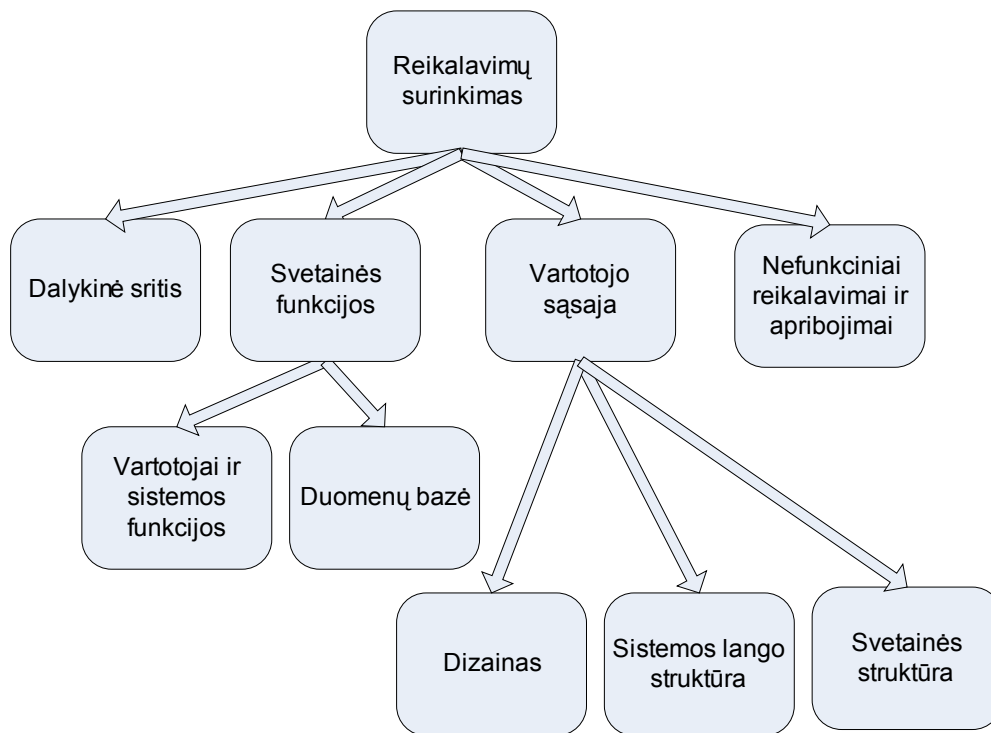
9 pav. Reikalavimų surinkimo metu įeinantys ir išeinantys duomenys

Reikalavimų surinkime dalyvauja užsakovas, vadovas ir dizaineris (jei pageidautinas didelis dėmesys dizainui). Organizuojamas interviu. Dizaineris dalyvauja interviu tam, kad susidarytų bendrą vaizdą apie sistemos pobūdį, pats užduoda užsakovui klausimus, kad išsiaiškintų, kokio dizaino pageidaujama.

Interviu metu surinkti reikalavimai yra susisteminti ir pateikiami kūrėjų komandai.

Pagal reikalavimų apimtį yra nustatoma pradinė projekto trukmė ir išteklių, tačiau vėliau, jei reikalavimai pasikeičia, jie gali kisti.

Sistemos funkcionalumui išsiaiškinti naudojama 10 pav. Reikalavimų surinkimo schema.



10 pav. Reikalavimų surinkimo schema

Dalykinė sritis

Šioje dalyje reikalinga išsiaiškinti tik pagrindinius svetainės kūrimo tikslus, apie ką bus kuriama svetainė, nesigilinant į detalius reikalavimus.

Norint išsiaiškinti internetinės svetainės dalykinę sritį, pateikiami tokie klausimai:

- Kokie svetainės kūrimo tikslai, koks svetainės tipas (teikti informaciją, e-komercija ir pan.)?
- Kokio turinio informaciją ketinate pateikti?
- Į kokio tipo auditoriją (vartotojus) orientuojamasi?

Gauta informacija užrašoma tekstu.

Dažnai reikalavimai yra keičiami dėl to, kad jie buvo aprašyti nevienareikšmiškai ir skirtingai suprasti skirtingų žmonių, dalyvaujančių projekte. Tam, kad būtų išvengta reikalavimų daugiareikšmiškumo reikalinga sudaryti dalykinės srities sąvokų žodyną. Dalykinės srities sąvokų žodyne reikia paaiškinti visus sistemai specifinius žodžius bei jais apibūdinamų sąvokų svarbiausias savybes. Dalykinės srities sąvokų žodynas pateiktas pavyzdyje .

17 lentelė Dalykinės srities sąvokų žodynas

Terminas	Paaiškinimas
Užsakovas	Asmuo, kuris administratoriaus yra užregistruotas sistemoje, galintis vykdyti užsakymus.
Išankstinis užsakovas	Asmuo, galintis pats užsiregistruoti sistemoje, pateikti užsakymą, tačiau užsakymas vykdomas, kai įmonei iš anksto yra pervedami pinigai.
Darbuotojas	Įmonės patikėtas darbuotojas, turintis teisę prieiti prie tam tikrų sistemos duomenų.

Svetainės funkcijos

Vartotojai ir sistemos funkcijos

- Kokios vartotojų kategorijos (anoniminis vartotojas, registruotas vartotojas, administratorius)?
- Kokios vartotojų funkcijos?
- Kokia informacija pasiekama atskiroms vartotojų grupėms?
- Kokios papildomos sistemos funkcijos?

Aprašomi sistemos funkciniai reikalavimai, t.y. kokias funkcijas turi atlikti kuriama sistema, ir kaip šios funkcijos turi būti atliekamos.

Vartotojų funkcijos pateikiamos *panaudojimo atvejų diagrama*. Kiekvienas panaudojimo atvejis yra aprašomas išsamiau, reikalavimus pateikiant tekstu, lentelėmis ar *veiklos diagramomis*.

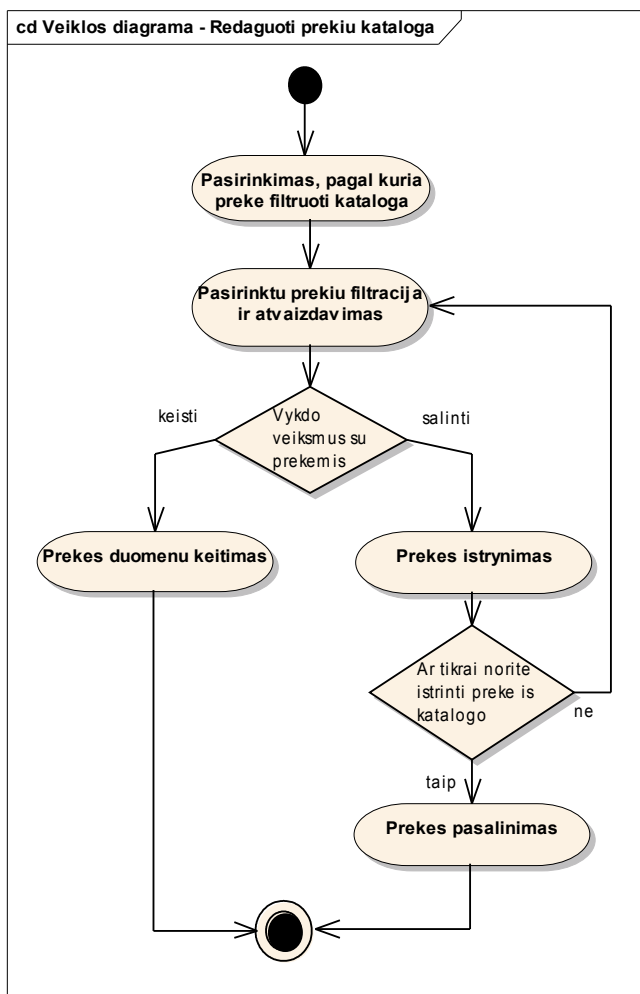
Jei panaudojimo atvejis yra aiškus, nebūtinai reikia braižyti veiklos diagramą, užtenka pateikti trumpą aprašymą, kaip sistema turi funkcionuoti, jei reikia – pateikti lenteles. Sudėtingesniais atvejais, reikalinga pateikti veiklos diagramas.

Nagrinėjame panaudos atvejį „Užregistruoti naują vartotoją“. Šiuo atveju vartotojo registraciją patogiausia pateikti lentele, pavaizduota 18 lentelė Vartotojo registracijos pavyzdys.

18 lentelė Vartotojo registracijos pavyzdys

Prisijungimo duomenys	
El.paštas:	
Slaptažodis:	
Pakartokite slaptažodį:	
Duomenys apie vartotoją	
Vardas:	
Pavardė:	
Miestas:	
Šalis	

Veiklos diagramos pavyzdys pateiktas 11 pav.



11 pav. Panaudojimo atvejo „Redaguoti prekių katalogą“ veiklos diagrama

Vartotojui gali būti pasiūlytos svetainei būdingos **tipinės sistemos funkcijos**:

- ✓ Kalbų pasirinkimo galimybė;
 - Ar reikalinga kalbų pasirinkimo galimybė?
 - Ar skirtingų kalbų meniu sutampa, ar skirtingomis kalbomis pateikiama skirtinga informacija (skirtingas meniu)?

- Kelios kalbos reikalingos?
- ✓ Vartotojų prisijungimas;
- ✓ Slaptažodžio priminimas;
- ✓ Paieškos funkcijos pateikimas;
- ✓ Kontaktų pateikimas;
- ✓ Kt.

Gauta informacija užrašoma tekstu.

Duomenų bazė

Vartotojui dažniausiai nerekomenduojama pateikti duomenų bazės schemos, jei jis neturi tam reikalingų žinių. Pavyzdžiui, norime sudaryti užsakymo duomenų bazės lenteles. Išėitis – pateikti lentelę, kuri atvaizduoja visus elementus, susijusius su užsakymu. Vartotojui turi būti parodoma, ką jis matys užsakymų lange. Viskas surašoma į užsakymo lentelę, o tada kūrėjai gali spręsti, kaip atrodys duomenų bazės schema.

Kūrėjai nusprendžia:

- Kokia yra duomenų bazės schema?
- Kokie ryšiai?

Duomenų bazės schemai atvaizduoti gali būti panaudojama *klasių diagrama* arba duomenų bazės schema gali būti nubraižoma naudojant tam skirtas priemones, pvz.: Microsoft Access.

Lentelė, sudaroma dalyvaujant užsakovui, pateikiama lentelėje 19 lentelė Užsakymo sudėtiniai elementai.

19 lentelė Užsakymo sudėtiniai elementai

Kur naudojama	Elementai
Prekės	Arbata
	Lotyniškas pavadinimas
	Sudėtis
	Aprašymas
	Kaina
	Komentarai
	Neto svoris
	Bruto svoris
	Kategorija
	Užsakymas
Telefono numeris	
Užsakymą sudarančio asmens vardas	
Pristatymo vieta	Gatvė
	Miestas
	Apskritis
	Pašto kodas

Vartotojo sąsaja

Dizainas

Reikalavimai, susiję su dizainu, aptariami su dizaineriu.

Priklausomai nuo užsakovo pageidavimų, svetainės pobūdžio, yra aptariamoms dizaino savybės.

Užsakovui pateikiami klausimai:

- Kokia spalvinė gama?
- Ar reikalinga flash animacija? (Su užsakovu turi būti aptarta ir flash animacijos įtaka svetainei.)
- Galbūt patinka kai kurių lankytų svetainių dizainas?
- Ar nori specialių formų, kontūrų (angl. shapes) mygtukų, dizaino?
- Koks raidžių stilius (pavadinimo, paprasto teksto ir pan.)?
- Kokio tipo grafiniai elementai (paveikslėliai, lentelės ir pan.)?

Užsakovo norai užrašomi tekstu. Taip pat turi būti pateikiami langų eskizai - piešiniai.

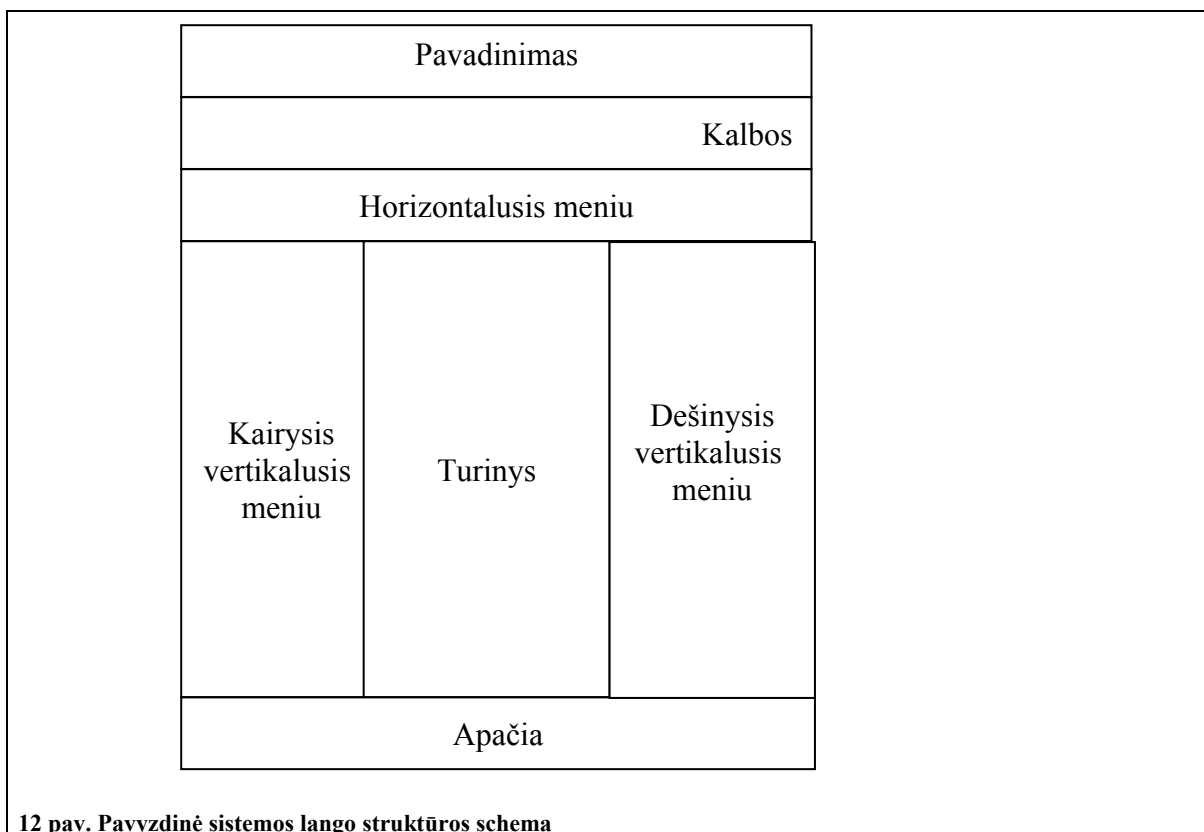
Dizaineris piešia ir pateikia užsakovui eskizus bei klausia, ar tinka.

Sistemos lango struktūra

- Kurioje vietoje užrašytas svetainės pavadinimas?
- Jei yra kalbų pasirinkimas, kurioje vietoje?
- Kaip išdėstyta svetainės informacija? Vertikalusis meniu? Horizontalusis meniu? Abu?

- Kokia informacija pateikta meniu?
- Ar reikalingas žemesnio lygio meniu?
- Kokia informacija pateikta puslapio apačioje?

Sistemos lango struktūra pateikiama *schema*, kurios pavyzdys pateiktas 12 pav. Pavyzdinė sistemos lango struktūros schema.

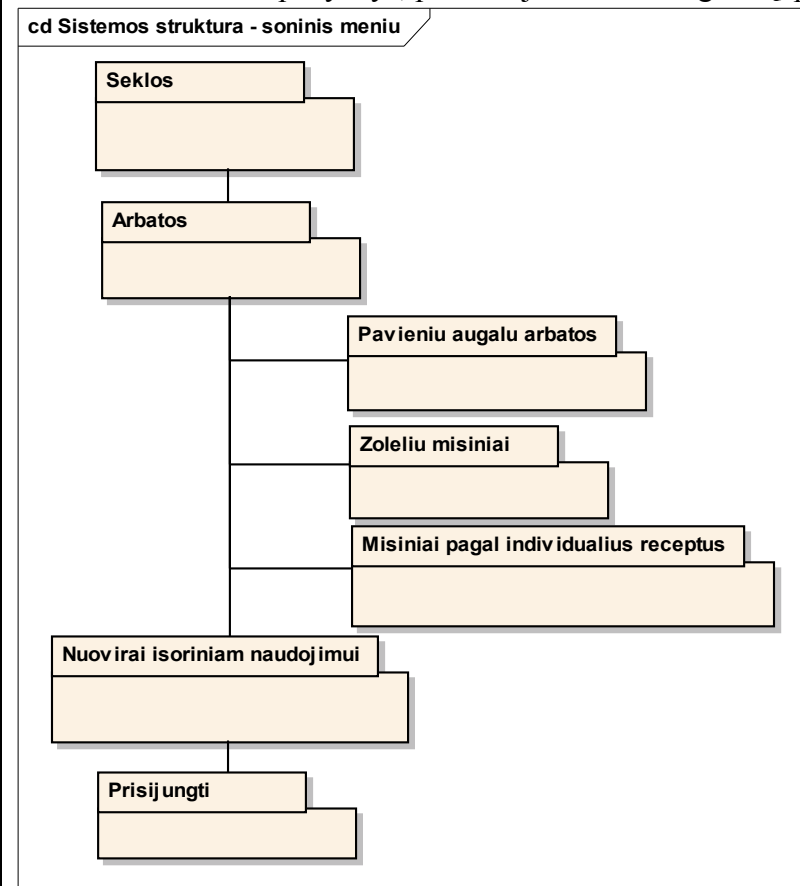


Svetainės struktūra

Apibrėžiama svetainės struktūra: pateikiama svetainės navigacija, t.y. kaip pasiekiami tam tikri duomenys, funkcijos.

Svetainės struktūros pateikimo notacija turi būti suprantama vartotojui. Tam gali būti naudojamos *UML diagramos* arba pateikiamos reikalavimų surinkimo metu ant popieriaus ranka braižytos schemas.

Svetainės struktūros pavyzdys, panaudojant UML diagramą, pateikiamas 13 pav.



13 pav. Svetainės struktūra - šoninis meniu

Nefunkciniai reikalavimai ir apribojimai

Nefunkciniai reikalavimai:

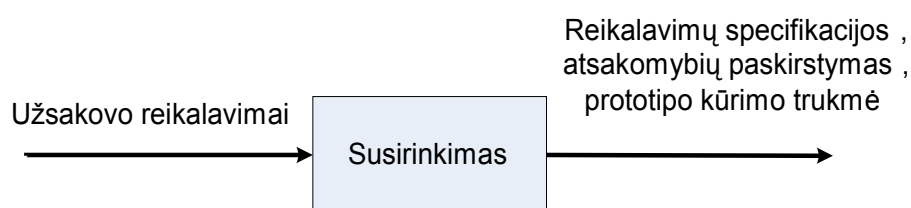
- Sistemos kokybės reikalavimai:
 - ✓ Patogumas (patogus vartotojo interfeisas);
 - ✓ Patikimumas (sistemos darbo korektiškumas);
 - ✓ Palaikomumas;
 - ✓ Saugumas (slaptažodis, duomenų kontrolė, vartotojų grupių prisijungimo apsauga ir pan.);
 - ✓ Greitis (atsakymo laikas);
 - ✓ Reikalavimai atminčiai;

- ✓ Mobilumas (perkeliamumas – sistema turi būti perkeliama į visas operacines sistemas, turinčias reikalingas programas bei atitinkančias apribojimus).
- Juridiniai bei kontrolės reikalavimai;
- Suderinamumas (su egzistuojančiomis priemonėmis, pvz.: programavimo kalbų ir operacinių sistemų);
- Sistemos apribojimai (Kokią programinę įrangą turi palaikyti serveris, kad sistema tinkamai funkcionuotų? Sistemos vartotojų apribojimai. Įvedamų duomenų apribojimai ir pan.);
- Techninė specifikacija:
 - ✓ Reikalavimai serveriui (Operacinė sistema, programinė įranga, duomenų bazė. Pasirenkant serverį, reikia atsižvelgti ir į planuojamą svetainės lankomumą.);
 - ✓ Sistemos prieinamumas (Ar keliami reikalavimai naršyklei? Kokios techninės ar programinės įrangos reikia, kad sistema būtų pasiekiamo?);
- Apribojimai (įvedimo/ išvedimo įrenginio galimybės).

Nefunkciniai reikalavimai užrašomi tekstu.

2.6.2 Susirinkimas

Specifikacijų pateikimo įeinantys ir išeinantys duomenys pavaizduoti 14 pav.14 pav.



14 pav. Specifikacijų pateikimo įeinantys ir išeinantys duomenys

Šio etapo metu yra:

- Paskirstomi darbai;
- Pateikiami užsakovo reikalavimai;
- Nustatoma iteracijos trukmė.

Susirinkimo trukmė priklauso nuo kūrimo progreso bei reikalavimų kiekio. Pirmasis susirinkimas reikalauja daugiausia laiko. Yra paskirstomos atsakomybės komandai, pateikiami reikalavimai, nustatoma, kiek užtruks iteracija. Susirinkimas gali trukti nuo 1 iki 5

valandų, priklausomai nuo projekto dydžio. Vėlesnių susirinkimų trukmė mažėja. Susirinkimas gali sutrumpėti net iki 15 minučių.

Susirinkimas gali būti organizuojamas keliais būdais:

- Susirenkant į vieną patalpą;
- Internetinė konferencija;
- Mišrus variantas (jei kuris nors komandos narys dirba nuotoliniu būdu).

Jei organizuoti susirinkimo netikslinga, reikalinga informacija gali būti pateikiama kiekvienam komandos nariui atskirai (tiesiogiai ar el. paštu).

Susirinkimo metu dalyvauja vadovas ir visi komandos nariai. Susirinkimą ir jo eigą organizuoja vadovas.

Komandos susitikimai organizuojami bent kartą vienos iteracijos metu. Dažniausiai organizuojamas vienas susirinkimas prieš pradėdant iteraciją. Jo metu kiekvienas komandos narys parodo, ką atliko prieš tai buvusios iteracijos metu. Pristatomos naujos sistemos savybės, aptariami kūrimo niuansai. Pačiam pristatymui nėra reikalinga sukurti prezentacijas, pakanka žodžiu paaiškinti, kas buvo atlikta iteracijos metu, pademonstruoti tai sistemoje. Susirinkimo metu turi būti pateikta kitiems komandos nariams reikalinga informacija.

Be atliktų darbų pristatymo, taip pat yra aptariami ir dar neįvykdyti darbai. Susirinkimo metu yra išsiaiškinami kylantys klausimai dėl sistemos funkcionalumo, įgyvendinimo priemonių bei galimybių. Kiekvieno susitikimo metu yra aptariami iteracijos darbai, svarbiausios savybės. Svetainės funkcionalumo aprašymus gauna kiekvienas komandos narys.

2.6.3 Architektūros nustatymas.

Surinkus reikalavimus, architektas apsprendžia, kaip komponentai bendrauja tarpusavyje, kaip architektūros lygiai pasiskirsto techniniuose įrenginiuose. Šiame etape nustatoma, kaip bus realizuoti posistemiai, logika, kaip organizuojamas svetainės turinys. Architektas priima architektūrinius sprendimus, kurie apima techninius, estetinius ir funkcinius kriterijus.

Architektūros nustatymo įeinantys ir išeinantys duomenys pavaizduoti 15 pav. .

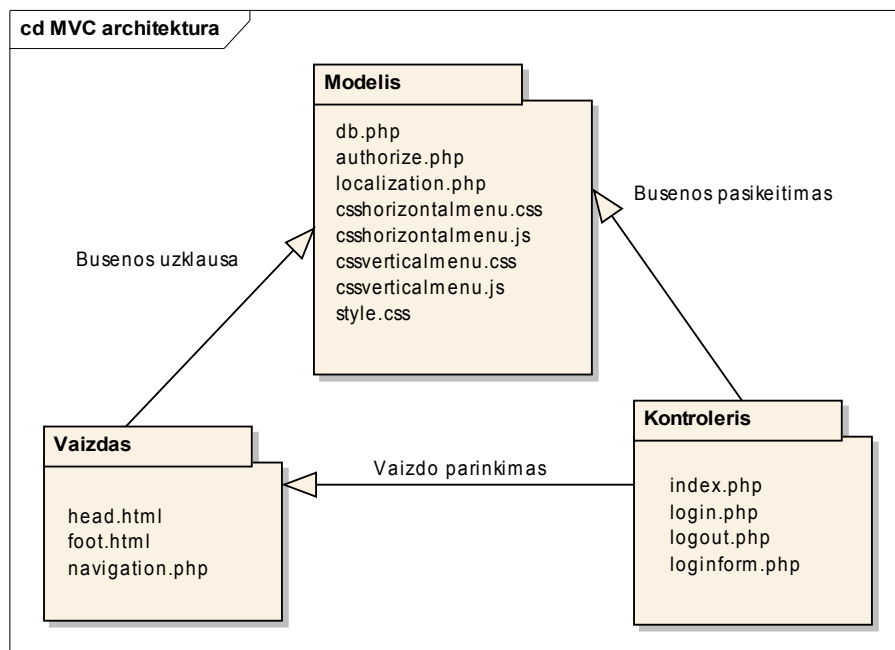


15 pav. Architektūros nustatymo įeinantys ir išeinantys duomenys

Pagal nustatytus apribojimus ir taisykles bus kuriama internetinė svetainė.

Pasaulinė praktika rodo, kad norint sukurti svetainę, kurią būtų lengva palaikyti, plėsti, geriausia naudoti MVC architektūrą.

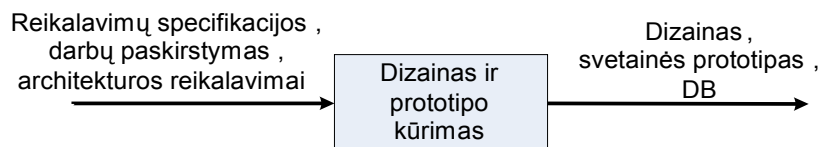
MVC architektūros pavyzdys pateiktas 16 pav., kur pateikta tipinių komponentų schema.



16 pav. MVC architektūros pavyzdys

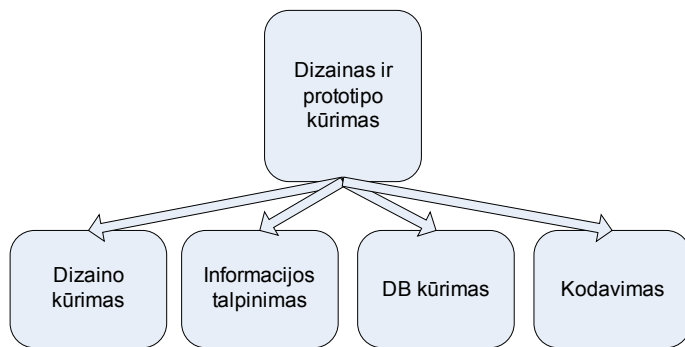
2.6.4 Dizainas ir prototipo kūrimas

Dizaino ir prototipo kūrimo įeinantys ir išeinantys duomenys pavaizduoti 17 pav.



17 pav. Dizaino ir prototipo kūrimo įeinantys ir išeinantys duomenys

Šio etapo metu yra kuriamas dizainas, talpinama informacija į svetainę, kuriama duomenų bazė, atliekami programavimo darbai. Jei komandos narių yra keletas, darbai gali būti atliekami lygiagrečiai. Dizaino ir prototipo kūrimo schema pavaizduota 18 pav.



18 pav. Dizaino ir prototipo kūrimo schema

2.6.5 Iteracinis kūrimas

Pirmosios iteracijos trukmė gali būti 1-3 savaitės, priklausomai nuo kuriamos svetainės apimties. Pirmosios iteracijos metu yra kuriamas pradinis prototipas, kurio savybės vėliau bus išplečiamos tolimesnių iteracijų metu.

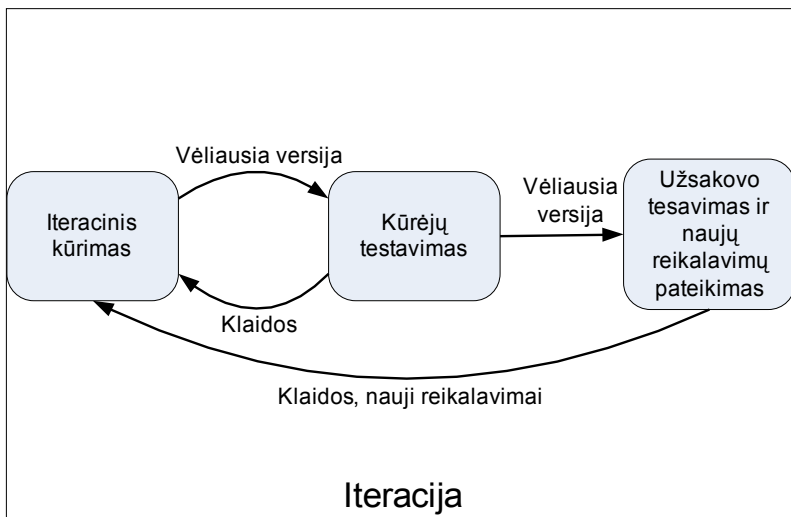
Iteracinio kūrimo metu:

- Vykdomi susirinkimai, kurių metu aptariami būsimi darbai.
- Architektūros parinkimas;
- Kuriama/keičiama svetainės struktūra;
- Rašomas programos kodas;
- Kuriamas/keičiamas dizainas;
- Atliekamas svetainės informacijos talpinimas;
- Duomenų bazės kūrimas.

Visi darbai atliekami lygiagrečiai.

Iteracijų trukmę nustato darbo vadovas susirinkimo metu, kuris įvyksta kiekvienos iteracijos pradžioje. Priklausomai nuo dar likusio padaryti darbo apimties tolimesnių iteracijų trukmė gali svyruoti nuo 1 dienos iki 1 savaitės.

Kūrimo proceso iteracija pavaizduota 19 pav.



19 pav. Kūrimo proceso iteracija

Kiekvienos iteracijos pradžioje yra organizuojami susirinkimai. Jo metu susisteminti užsakovo reikalavimai yra pateikiami komandai. Komandai yra paskirstomi darbai. Kiekvienas komandos narys turi savo užduotis. Tolesnių iteracijų metu užduotis stengiamasi paskirstyti taip, kad sietųsi su prieš tai turėta užduotimi.

Architektūros nustatymas paprastai vykdomas tik pirmosios iteracijos metu. Kitų iteracijų metu šis etapas yra neliečiamas, nebent pasikeitus užsakovo reikalavimams reikalinga pakeisti architektūrą.

Ilgiausia yra pirmoji iteracija, ji gali užtrukti 1-3 savaites, priklausomai nuo svetainės apimties, kadangi pirmos iteracijos metu stengiamasi įvykdyti kuo daugiau pradinių reikalavimų. Pirmosios iteracijos metu turėtų būti sukurtas pradinis dizaino variantas, sukurta navigacijos sistema, sukurta pradinė užsakovo pateikta svetainės informacija, sukurta duomenų bazė, vykdomi programavimo darbai. Iteracijos metu reikalinga įvykdyti tiek darbų, kad užsakovas turėtų galimybę įvertinti kuriamą svetainę, susidaryti išpūdį apie svetainės funkcijas ir galimybes. Tai nebūtinai turi būti galutinis išbaigtas variantas. Vėlesnių iteracijų trukmės trumpėja priklausomai nuo to, kiek reikalavimų dar liko neįvykdyta, ar užsakovas pateikia papildomų reikalavimų. Iteracijų kiekis priklauso nuo svetainės apimties. Kuo didesne apimtis ir daugiau reikalavimų, tuo daugiau iteracijų prireiks.

Daugiau iteracijų nebevykdoma, kai užsakovas nebepateikia daugiau jokių papildomų reikalavimų. Pristatomas galutinis produktas.

2.6.6 Kūrėjų testavimas

Kūrėjų testavimas neturi būti labai formalus, nes priešingu atveju kūrimo laikas labai užsitęstų. Prieš pateikiant svetainės dalis testuoti užsakovui, testavimą pirmiausia atlieka testuotojas. Jei svetainė yra kuriama komandos iš keleto narių, kur testuotojo ir programuotojo pareigas atlieka tas pats asmuo, testavimas turi būti organizuojamas taip, kad

komandos nariai būtų įtraukti ir į savo kolegų kuriamų dalių testavimą, kadangi taip lengviau pastebėti klaidas. Pirmiausia ištestuojama savoji dalis, ištaisomos surastos klaidos, o tik vėliau perduodama kolegoms testavimui.

Kadangi yra vengiama formalus testų rašymo, svetainė testuojama remiantis pateiktais užsakovo reikalavimais. Rastos klaidos yra surašomos į lentelę, pateiktą **20 lentelėje**.

20 lentelė Testavimo rezultatai

Kas testavo?	Kokia klaida?	Kur ta klaida rasta?	Kaip ištaisyti klaidą?	Kas ištaisė klaidą?

2.6.7 Užsakovo testavimas ir naujų reikalavimų pateikimas

Užsakovas atlieka visos svetainės testavimą ir užpildo 20 lentelę Testavimo rezultatai pirmus tris lentelės stulpelius: Kas testavo? Kokia klaida? Kur ta klaida rasta? Jei užsakovas nori padaryti nežymius pakeitimus reikalavimuose, tada užpildo ir ketvirtąjį stulpelį „Kaip ištaisyti klaidą?“. Yra įrašoma, kokias funkcijas turi sistema atlikti ar kokią informaciją teikti.

Naujų reikalavimų pateikimas gali būti organizuojamas vien tik su vadovu arba kartu su vadovu ir komandos nariais. Komandos nariai dalyvauja naujų reikalavimų pateikimo susitikime su užsakovu tokiu atveju, jei pakeitimai tiesiogiai susiję su jų kuriama dalimi ir tai yra būtina norint tinkamai išsiaiškinti reikalavimus. Reikalavimai pateikiami tokia pačia forma kaip ir pradiniai reikalavimai.

2.6.8 Galutinis produktas

Kai užsakovas nebeturi papildomų reikalavimų ir sistema veikia nepriekaištingai, svetainė yra užregistruojama paieškos sistemose ir galutinis produktas yra pateikiamas užsakovui.

2.7 Komandos bendradarbiavimas

Labai svarbus komandos narių bendradarbiavimas kiekvienos iteracijos metu. Tai reikiamos informacijos pasidalinimas su kitais komandos nariais. Jei komanda dirba vienoje patalpoje, sunkumų bendraujant nekyla. Tačiau galima darbą organizuoti ir taip, jog komandos nariai dirba ir atskirose patalpose, skirtinguose miestuose ar net valstybėse. Bendravimas gali vykti telefonu, internetu arba tiesioginiu pokalbiu su reikiamu komandos nariu. Komandos nariai nesunkiai gali matyti kolegų darbą, kadangi svetainė pasiekiamą internetu.

2.8 Užsakovo įtraukimas

Kiekvienos iteracijos pabaigoje užsakovui pateikiama vėliausia produkto versija. Jis peržiūri sukurtą produktą, testuoja, ar sistema atitinka visus jo keliamus reikalavimus. Pataisymai bei nauji reikalavimai yra pateikiami vadovui ir komandai. Detalesni reikalavimai gali būti aptariami ir tiesiogiai su konkrečią sistemos dalį atliekančiais komandos nariais.

Iškilus su reikalavimais susijusiais neaiškumams kūrimo procese, užsakovas turi būti visada pasirengęs pakonsultuoti ir patikslinti reikalavimus.

Bendravimas su užsakovu gali būti atliekamas susitinkant ir bendraujant tiesiogiai, telefonu, el. paštu ar kitokiu būdu. Iškilus didesnėms problemoms, su užsakovu bendrauja vadovas. Jei iškyla nedideli neaiškumai, susisiekti su užsakovu gali ir komandos nariai.

3 Eksperimentinis tyrimas

Lygiagrečiai kuriamam metodui yra daromas bandomasis projektas, kurio metu kuriama internetinė svetainė įmonei „Jadvygos žolės“. Šios svetainės kūrimo metu bandoma nustatyti pritaikyto metodo efektyvumą internetinių svetainių kūrime.

3.1 Eksperimentinio tyrimo projektas ir realizacija

Šiame skyriuje pateikta kuriamos svetainės „Jadvygos žolės“ reikalavimų surinkimo medžiaga, pritaikant kuriamą metodą.

3.1.1 Dalykinė sritis

„Jadvygos Žolės“ svetainė apima informacinę sistemą ir užsakymo sistemą. Vartotojams yra pateikiama įvairi informacija apie ūkį, ūkio gaminamą produkciją, įvairias arbatžoles ir pan. Užsakovams yra suteikiama galimybė užsakymus vykdyti internetu. Sistemoje kaupiami ir sisteminami registruotų vartotojų duomenys.

Dalykinės srities sąvokų žodynas pateiktas 21 lentelėje.

21 lentelė Dalykinės srities sąvokų žodynas

Terminas	Paiškinimas
Užsakovas	Asmuo, kuris administratoriaus yra užregistruotas sistemoje, galintis vykdyti užsakymus.
Išankstinis užsakovas	Asmuo, galintis pats užsiregistruoti sistemoje, pateikti užsakymą, tačiau užsakymas vykdomas, kai įmonei iš anksto yra pervedami pinigai.
Darbuotojas	Įmonės patikėtas darbuotojas, turintis teisę prieiti prie tam tikrų sistemos duomenų.

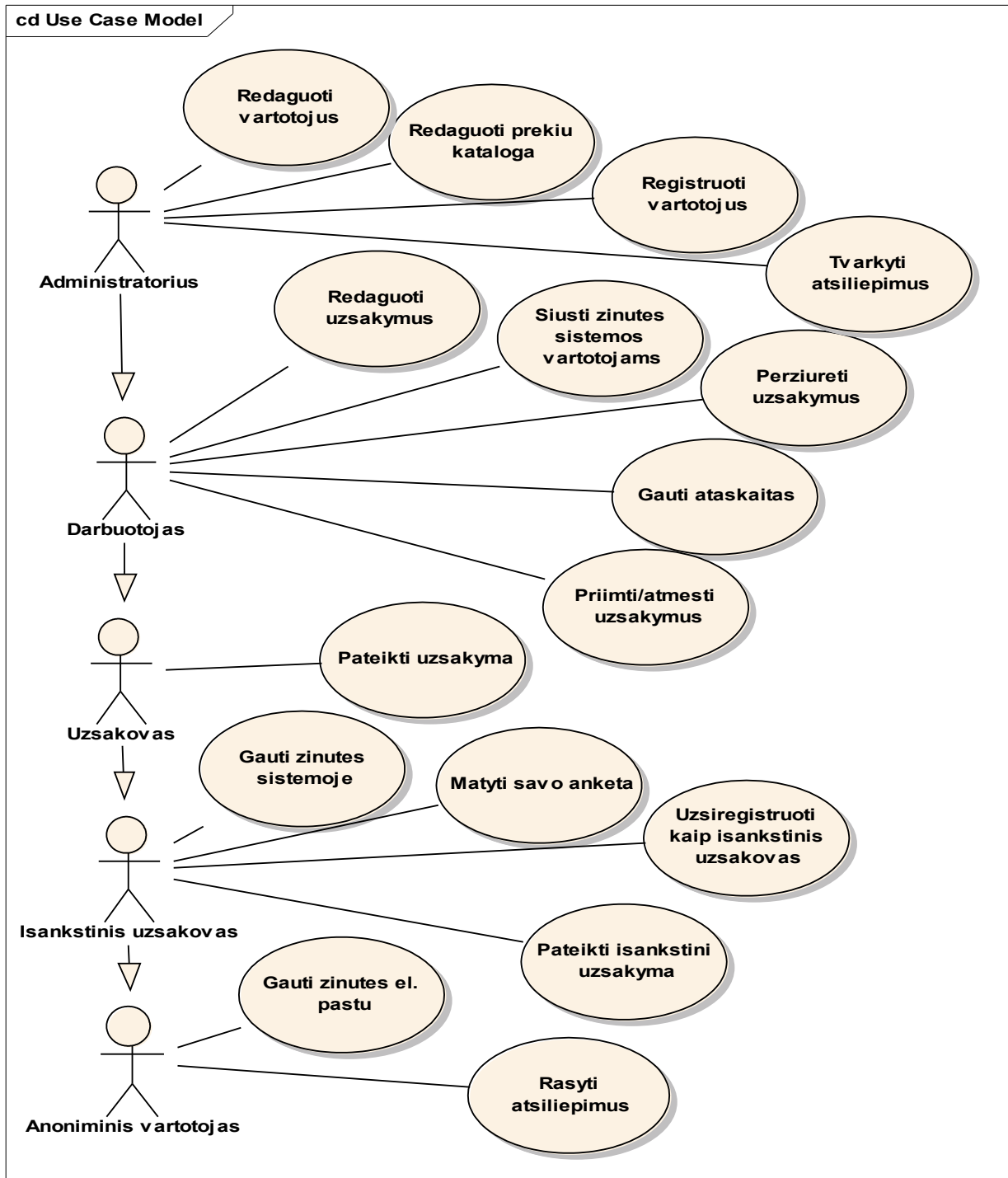
3.1.2 Vartotojai ir sistemos funkcijos

Sistemoje išskiriamos penkios vartotojų kategorijos:

- anoniminis vartotojas;
- užsakovas;
- išankstinis užsakovas;
- darbuotojas;
- administratorius;

Kiekviena vartotojų grupė turi skirtingas teises ir apribojimus.

Sistemos panaudos atvejų modelis pavaizduotas 20 pav. Panaudos atvejų diagrama.



20 pav. Panaudos atvejų diagrama

Projekto vartotojai – *administratoriai*: įmonės savininkas bei jo patikėti asmenys; *Darbuotojai* – ūkio darbuotojai, turintys teisę prieiti prie tam tikrų sistemos duomenų. *užsakovai* – kitų įmonių atstovai, sudarę sutartis su įmone; *Išankstiniai užsakovai* – asmenys, kurie gali išskirtinėmis teisėmis užsisakyti produkciją. Jei užsakymas patvirtinamas, jie turi iš anksto atsiskaityti už prekes; *anoniminiai vartotojai* – visi interneto vartotojai turintys suderinamas naršykles.

Anoniminio vartotojo galimybės

- Peržiūrėti sistemoje esančią informaciją
- Gauti žinutes elektroniniu paštu
- Rašyti atsiliepimus
- Užsiregistruoti kaip išankstinis užsakovas

Užsakovo galimybės

- Peržiūrėti sistemoje esančią informaciją
- Pateikti užsakymą
- Matyti savo anketą
- Gauti ataskaitas
- Gauti žinutes elektroniniu paštu
- Gauti žinutes sistemoje
- Rašyti atsiliepimus

Išankstinio užsakovo galimybės

- Peržiūrėti sistemoje esančią informaciją
- Pateikti išankstinį užsakymą
- Redaguoti savo anketą
- Gauti ataskaitas
- Gauti žinutes elektroniniu paštu
- Gauti žinutes sistemoje
- Rašyti atsiliepimus

Darbuotojo galimybės

- Peržiūrėti sistemoje esančią informaciją
- Redaguoti užsakymus
- Peržiūrėti užsakymus
- Priimti/atmesti užsakymus
- Matyti savo anketą
- Gauti ataskaitas
- Gauti žinutes elektroniniu paštu
- Gauti žinutes sistemoje
- Rašyti atsiliepimus

Administratoriaus galimybės

Administratorius turi visas teises kaip ir užsakovas.

Papildomos administratoriaus galimybės:

- Registruoti vartotojus
- Redaguoti duomenis apie vartotojus
- Suteikti teises
- Redaguoti savo anketą
- Ištrinti vartotojus ir jų duomenis iš duomenų bazės
- Siųsi žinutes elektroninių paštu
- Siųsti žinutes sistemos vartotojams
- Rašyti atsiliepimus
- Tvarkyti atsiliepimus
- Redaguoti užsakymus
- Peržiūrėti užsakymus
- Priimti/atmesti užsakymus
- Gauti ataskaitas
- Redaguoti prekių katalogą

Vartotojų registracija

Užregistruoti vartotojus turi teisę tik administratorius. Registruojant vartotoją prie sistemos reikia užpildyti registracijos formą, patvirtinti duomenis ir, jei viskas užpildyta korektiškai, t.y. prisijungimo vardas (el. paštas) nesidubliuoja ir atitinka el.pašto formos kriterijus, taip pat įvestas korektiškas slaptažodis, naujas vartotojas įtraukiamas į DB.

Vartotojų prisijungimui naudojamas registracijos formoje įvestas prisijungimo vardas (el. paštas). Slaptažodžiui taikomas tik simbolių skaičiaus apribojimas (mažiausiai turi būti 5 simboliai).

Registracijos formoje reikia užpildyti visus būtinus laukus, priešingu atveju registracija neįvyks. Registruojantis išankstiniam užsakovui pateikiama speciali registracijos forma, kur užpildyti reikia tik būtiniausius laukus.

Vartotojų registracijai atlikti reikalinga užpildyti formą, pateiktą 22 lentelėje.

22 lentelė Vartotojų registracija

Prisijungimo duomenys	
El.paštas:	
Slaptažodis:	
Pakartokite slaptažodį:	
Duomenys apie vartotoją	
Vardas:	
Pavardė:	
Gatvė:	

Apskritis:	
Miestas:	
Įmonė:	
Įmonės kodas:	
Pašto kodas:	
PVM kodas:	
Sąskaitos numeris	
Šalis	

Slaptažodžio priminimas

Užmiršęs slaptažodį vartotojas turi galimybę gauti naują slaptažodį, t.y. jis turi įvesti savo vartotojo vardą ir el. pašto adresą, kuris buvo įvestas registracijos formoje. Tuo adresu yra išsiunčiamas naujas slaptažodis. Slaptažodis yra koduojamas naudojant MD5 algoritimą.

Žinučių siuntimas sistemos vartotojams

Administratorius ir darbuotojas gali išsiųsti norimą pranešimą visiems registruotiems sistemos vartotojams, tam tikrai registruotai vartotojų kategorijai (administratoriams, užsakovams ir t.t.) arba pavieniams vartotojams.

Žinučių siuntimas elektroniniu paštu

Administratorius gali išsiųsti informaciją visiems klientams, kurie nori gauti naujienas el. paštu (pateikė savo el. pašto adresą).

Pateikti užsakymą

Užsakovas gali pateikti užsakymą. Jis turi užpildyti formos laukus, pateiktus 23 lentelėje.

23 lentelė Užsakymas

Produkto ID	Pavadinimas	Neto	Kaina	Kiekis	Suma	Komentarai
				Iš viso:		

Užsakovui, įvedus produkto ID, automatiškai yra parenkamas produkto pavadinimas, o pasirinkus produkto pavadinimą – produkto ID. Užsakovas taip pat turi nurodyti, kiek pakelių norimo produkto jis užsako. Taip pat yra suteikta galimybė parašyti komentarą prie užsakomo produkto.

Parametrai: Neto, Kaina ir Suma yra parenkami/apskaičiuojami automatiškai.

Administratoriui ar darbuotojui peržiūrėjus užsakymą, jis gali būti koreguojamas arba paliekamas esamas.

Pateikti išankstinį užsakymą

Išankstinis užsakovas turi galimybę pateikti išankstinį užsakymą. Užsakymas turi būti patvirtinamas administratoriaus arba darbuotojo. Už prekes turi būti atsiskaityta iš anksto. Po prekių apmokėjimo, prekės yra išsiunčiamos išankstiniam užsakovui.

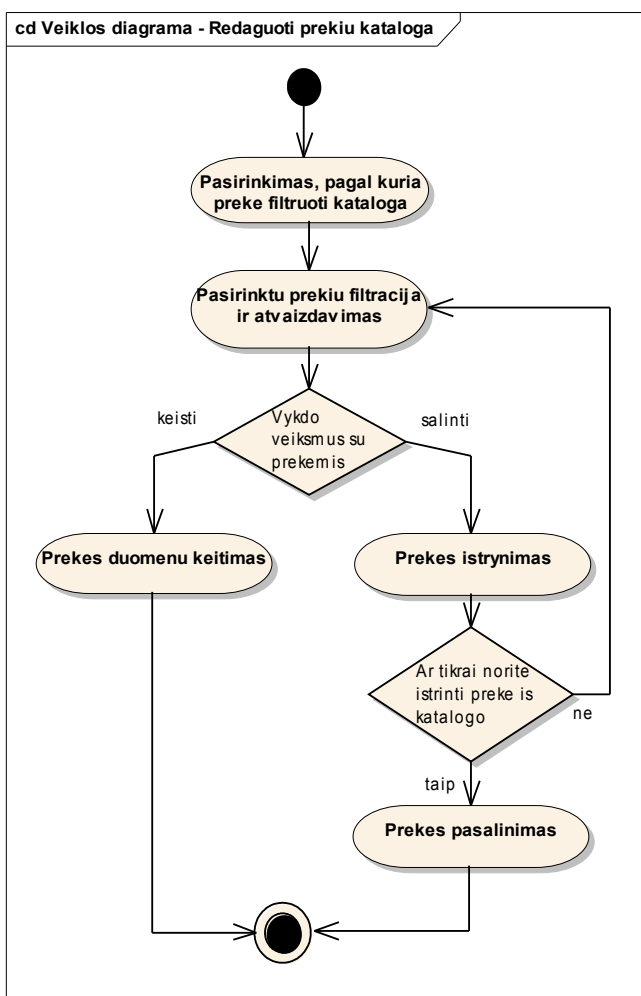
Užsakymo forma yra tokia pati kaip ir užsakovo. Ji pateikta 23 lentelėje.

Peržiūrėti bei redaguoti užsakymus

Administratorius bei darbuotojas gali peržiūrėti bei redaguoti užsakymus. Tai yra daroma, kai užsakovas paprašo pakeisti jo užsakymą.

Redaguoti prekių katalogą

Prekių katalogo redagavimo veiklos diagrama pateikta 21 pav.



21 pav. Panaudojimo atvejo „Redaguoti prekių katalogą“ veiklos diagrama

Priimti/atmesti užsakymus

Administratorius bei darbuotojas priima arba atmeta užsakymus. Jei užsakymas yra atmetamas, susisiekiama su užsakovu arba su išankstiniu užsakovu, ir užsakymas yra atmetamas arba redaguojamas .

Tipinės sistemos funkcijos

Vartotojo sąsajos kalba

Vartotojas turi galimybę pasirinkti, sąsajos kalbą iš lietuvių į anglų. Vokiečių, rusų ir prancūzų kalbomis vartotojui pateikiama tik sutrumpinta informacija apie įmonės teikiamą produkciją. Yra galimybė nesunkiai plėsti kalbų pasirinkimą.

Vartotojo prisijungimas ir slaptažodžio priminimas

Vartotojo prisijungimui naudoti el. pašto adresą ir slaptažodį.

Prisijungimo lange turi būti galimybė atsisiųsti naują slaptažodį į elektroninį paštą, kuriuo vartotojas buvo užregistruotas.

3.1.3 Duomenų bazė

Lentelė, sudaroma dalyvaujant užsakovui, pateikiama lentelėje 19 lentelė Užsakymo sudėtiniai elementai.

24 lentelė Užsakymo sudėtiniai elementai

Kur naudojama	Elementai
Prekės	Arbata
	Lotyniškas pavadinimas
	Sudėtis
	Aprašymas
	Kaina
	Komentarai
	Neto svoris
	Bruto svoris
	Kategorija
Užsakymas	Data
	Telefono numeris
	Užsakymą sudarančio asmens vardas
Pristatymo vieta	Gatvė
	Miestas
	Apskritis
	Pašto kodas

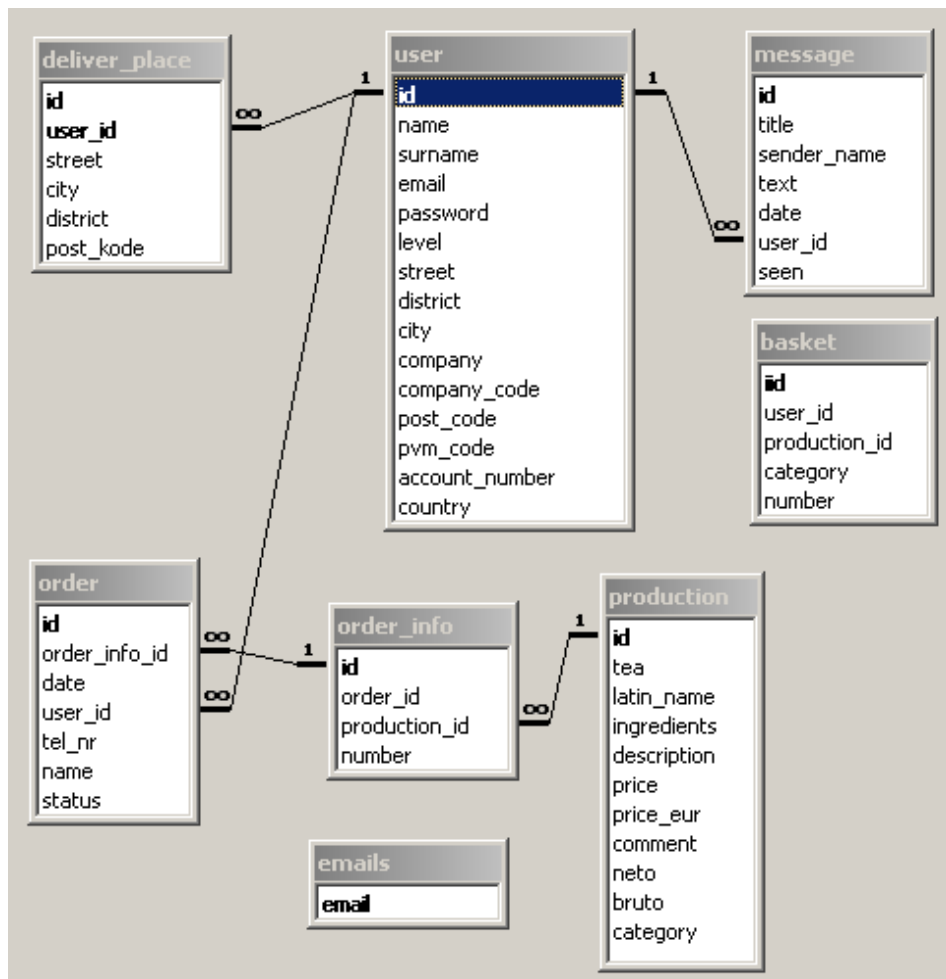
Informacija apie vartotojus pateikta 25 lentelėje.

25 lentelė Vartotojų registracijos duomenys

Kur naudojama	Elementai
----------------------	------------------

Vartotojų registracija	vardas
	pavardė
	el. paštas
	slaptažodis
	gatvė
	apskritis
	miestas
	įmonė
	įmonės kodas
	pašto kodas
	pvm kodas
	sąskaitos numeris
	šalis

Duomenų bazės schema pateikta 22 pav. Į duomenų bazę duomenys turi būti įvesti teisingai, tikrinamas el. pašto ir slaptažodžio teisingumas.



22 pav. Duomenų bazės schema

3.1.4 Vartotojo sąsaja. Dizainas

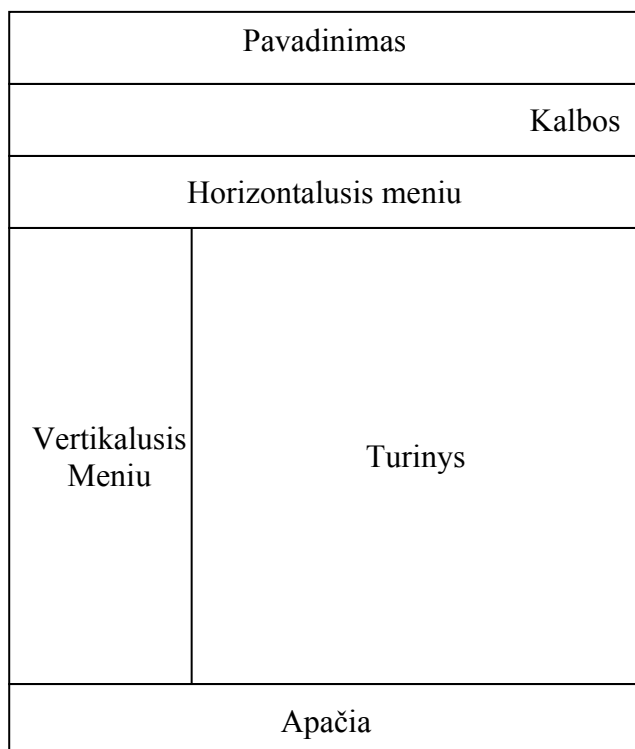
Spalvinė gama – žalia.

Papildomų reikalavimų nepateikta.

3.1.5 Vartotojo sąsaja. Sistemos lango struktūra

Sistemos išvaizdai nėra keliami ypatingų reikalavimų, tiesiog vartotojo sąsaja turi būti paprasta naudoti, aiški. Ji realizuota dviem kalbomis: lietuvių ir anglų kalba. Pageidaujama galimybės prireikus nesunkiai padidinti kalbų skaičių.

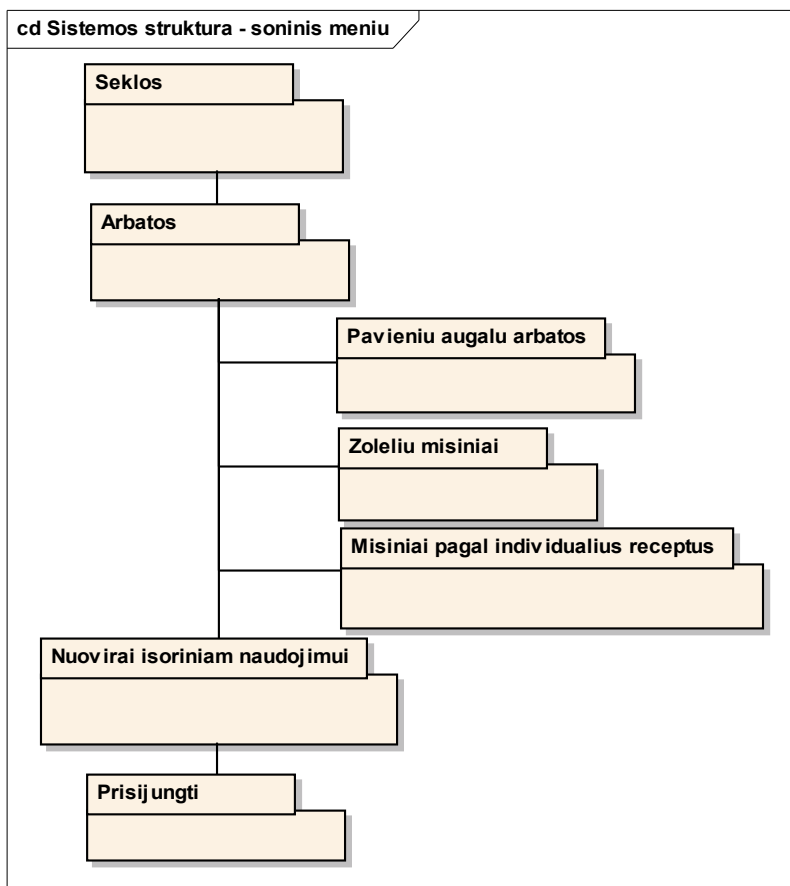
Sistemos lango struktūra pavaizduota 23 pav.



23 pav. Sistemos lango struktūra

3.1.6 Vartotojo sąsaja. Sistemos struktūra

Svetainės šoninis meniu yra pateikiamas 24 pav.



24 pav. Svetainės struktūra - šoninis meniu

3.1.7 Nefunkciniai reikalavimai ir apribojimai

3.1.7.1 Sistemos saugumo priemonės

Pagrindinės saugumo priemonės yra slaptažodis ir duomenų kontrolė. Kiekvienas registruotas vartotojas turi prisijungti prie sistemos. Slaptažodis įvedamas nematoma forma, t.y. simbolius pakeičiant žvaigždutėmis. Pačioje sistemoje Slaptažodis saugomas naudojant MD5 algoritmą taigi net tinklo administratoriai jo sužinoti negali – taip išsaugoma slaptažodžio privatumo politika.

Registracijos metu ir vėliau, keičiant asmeninius duomenis, taikoma jų kontrolė, t.y. pranešama apie klaidingai įvestus duomenis. Slaptažodis įvedamas du kartus, norint užtikrinti, kad nebuvo padaryta klaida.

Kadangi vartotojai išskiriami į penkias vartotojų grupes (neregistruotas vartotojas, užsakovas, išankstinis užsakovas, darbuotojas ir administratorius) yra apsaugota, kad kiekviena grupė galėtų prieiti tik jos teises atitinkančią sritį, t.y. užsakovas adreso lauke įvedęs puslapio skirto siųsti žinutes adresą, jo nepamatys.

3.1.7.2 Sistemos prieinamumas

Sistema pasiekama iš bet kurios vietos, kur yra interneto ryšys ir reikalinga techninė bei programinė įranga.

3.1.7.3 Sistemos apribojimai

Serveris, kuriame bus patalpina sistema, turi palaikyti PHP.

Visi sistemos vartotojai (anoniminis vartotojas, užsakovas ir administratorius) turi skirtingus apribojimus.

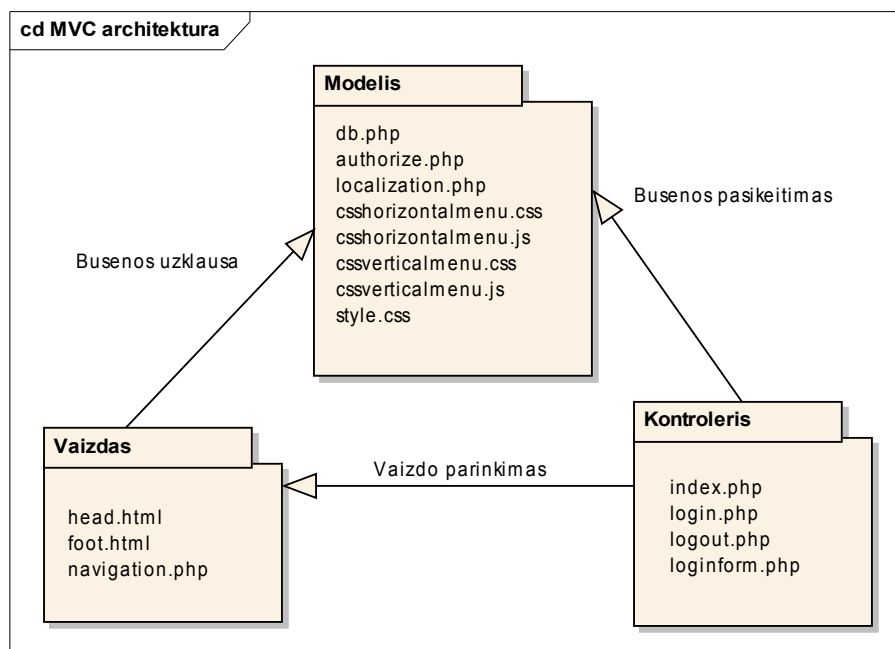
Pildant registracijos formą, tikrinama, ar viskas užpildyta korektiškai, t.y. ar prisijungimo vardas (el. paštas) nesidubliuoja ir atitinka el.pašto formos kriterijus, taip pat ar įvestas korektiškas slaptažodis. Slaptažodis turi būti ne trumpesnis nei 5 simboliai. Būtina užpildyti visus reikalingus registracijos formos laukus.

3.1.7.4 Techninė specifikacija

Programinė įranga kuriama naudojant WWW technologijas: Apache web server, Linux platforma, PHP programavimo kalba ir MySQL duomenų bazė. Programa paleidžiama naudojant bet kurią interneto naršyklę.

3.1.8 Architektūros nustatymas

MVC architektūros pavyzdys pateiktas 16 pav., kur pateikta tipinių komponentų schema.



25 pav. MVC architektūros pavyzdys

3.1.9 Kūrimo procesas

Užsakovo reikalavimų surinkimas:

- Pirmiausia buvo aptartas svetainės pobūdis, kokia informacija bus talpinama svetainėje.
- Vartotojo sąsajos dizainas ir sistemos lango struktūra. Meniu ir juo pasiekiamą informaciją. Visa pateikta informacija buvo užrašoma lapuose.
- Aptartos vartotojų kategorijos. Proceso eigoje šie reikalavimai pasikeičia: užsakovui pasiūlytas tinkamiausias variantas.
- Pageidautina svetainės su kalbų pasirinkimo galimybė.
- Kompiuterizuojamų funkcijų sąrašas iš pradžių buvo surašomas į atskiras dalis atitinkamai kiekvienai vartotojų kategorijai. Vėliau pateikta panaudojimo atvejų diagrama.
- Aptarta duomenų bazės schema. Jos struktūra keitėsi kelis kartus kūrimo procese vis pasikeičiant ar papildant užsakovo reikalavimus.
- Pabaigoje aptartos svetainės patalpavimo serveryje galimybės. Pasiūlytas optimaliausias variantas.

Susirinkimas

Ekspertimentiniu atveju susirinkimai nebuvo organizuojami, kadangi kūrimo procese dalyvauja tik vienas asmuo.

Architektūros nustatymas.

Internetinės svetainės kūrimui naudojama MVC architektūra.

Vartotojo sąsajos kūrimui pasirinkti tokie įrankiai:

- Macromedia Dreamweaver
- Paint.NET

Projektavimui naudojami įrankiai:

- Enterprise Architect

Dizainas ir prototipo kūrimas

Atsižvelgiant į užsakovo norus (kuo skubiau patalpinti pradinę informaciją), buvo sukurta svetainės struktūra ir pradinis minimalus dizainas, kuris vėlesnių iteracijų metu buvo pakeistas. Dizaino ir prototipo kūrimas vyko lygiagrečiai.

Iteracinis kūrimas

Iteracijos trukmė nepastovi, ypač pradinių iteracijų. Užsakovas negali pateikti visos kūrimui reikalingos informacijos. Tam tikros svetainės kūrimo dalys laikinai buvo pristabdytos. Reikalinga informacija užsakovo buvo pateikiama palaipsniui. Pagrindinė

problema – užsakovas tiksliai nežino, ką sistema turėtų daryti ir kokia informacija turi būti pateikiama internetinėje svetainėje.

Iteracinio kūrimo metu nuolat buvo atnaujinama pateikta informacija. Jei kūrėjų komandą sudarytų ne vienas asmuo, kai iteracijos darbai jau pradėti, gali tekti surengti susirinkimą ir įpusėjus iteracijai.

Iteracijų trukmės pateiktos 26 pav. Projekto trukmės lentelė.

Pirmosios iteracijos metu atlikti darbai:

- Parinkta architektūra;
- Sukurta svetainės struktūra;
- Sukurtas pradinis dizaino šablonas (vėlesnių iteracijų metu jis buvo pakeistas);
- Patalpinta pradinė svetainės informacija;
- Sukurta pradinė duomenų bazė (vėlesnių iteracijų metu ji buvo papildyta);
- Sukurta vartotojų prisijungimo galimybė;

Antrosios iteracijos metu atlikti darbai:

- Pakeisti navigacijos meniu (vartotojui pasiūlytas optimaliausias svetainės naršymo būdas);
- Keičiamas dizainas;
- Talpinama svetainės informacija;
- Papildoma duomenų bazė;
- Sukuriama vartotojų registracija;
- Sukuriama vartotojų valdymo galimybė (vartotojų duomenų redagavimas, vartotojų šalinimas, vartotojo teisių suteikimas).

Trečiosios iteracijos metu atlikti darbai:

- Koreguojamas dizainas;
- Koreguojamas meniu;
- Talpinama svetainės informacija;
- Sukuriama informacijos siuntimo el. paštu galimybė;
- Sukuriama vidinė svetainės žinučių siuntimo sistema registruotiems vartotojams.

Ketvirtosios iteracijos metu atlikti darbai:

- Talpinama svetainės informacija;
- Prekių katalogo papildymo galimybė;
- Prekių katalogo redagavimo galimybė;
- Produktų užsakymo galimybės kūrimas.

Penktosios iteracijos metu atlikti darbai:

- Produktų užsakymo galimybės kūrimas (pasikeitus reikalavimams ir pasiūlius patogesnę vartotojui galimybę, užsakymo sistema kuriama iš naujo);
- Papildoma duomenų bazė;
- Talpinama svetainės informacija;
- Dizaino kregavimas;
- Užsakymų peržiūros kūrimas.

Testavimas

Sistemos testavimo rezultatai pateikti 26 lentelė Testavimo rezultatai.

26 lentelė Testavimo rezultatai

Kas testavo?	Kokia klaida?	Kur ta klaida rasta?	Kaip ištaisyti klaidą?	Kas ištaisė klaidą?
Eglė	Viršutinis meniu rodomas „iškreiptai“		Problemos su .css ir .js failais	Viktorija
Viktorija	„Jadvygos Balvočiūtės ekologinis vaistažolių ūkis žinomas žolelių arbatų gamintojas.“ Trūksta brūkšnio prieš „žinomas“.	lt_apie_uki.php	„Jadvygos Balvočiūtės ekologinis vaistažolių ūkis – žinomas žolelių arbatų gamintojas.“	Viktorija
Viktorija	Per didelis žemėlapių paveikslas.	lt_kontaktai.php	Sumažinti paveiksluką.	Viktorija
Eglė	Yra nereikalingos informacijos	ru_tekstas.php fr_tekstas.php	Pateiktas dokumentas su pakeista informacija.	Viktorija
Viktorija	Išsiunčiant naują slaptažodį, el. laiške nerašomas siuntėjas.		Įrašyti į slaptažodžio siuntimo funkciją „from“.	Viktorija
Viktorija	Vartotojų tvarkyme negalima suteikti išankstinio užsakovo teisių.	user_mng.php	Įvesti pasirinkimą „Išankstinis užsakovas“.	Viktorija
Viktorija	Nėra įvedamų duomenų kontrolės	papildyti_prekiu_kataloga.php	Įvesti duomenų kontrolę	Viktorija

Kūrėjų testavimas

Atliktų dalių testavimas atliekamas kiekvieną kartą prieš pateikiant testuoti užsakovui.

Užsakovo testavimas ir naujų reikalavimų surinkimas

Kadangi reikalavimai svetainei buvo nuolat papildomi, ne visada laikantis sistemingumo, galima teigti, kad testavimas taip pat vyko lygiagrečiai.

3.2 Ekspertiniai vertinimai

Eksperimentinių atveju kūrėjas yra vienas asmuo, kuris atlieka visus vaidmenis: ir komandos (dizaineris, architektas, programuotojas, testuotojas), ir vadovo. Šiuo atveju neįvertinamas metodo tinkamumas, kūrėjų komandai esant didesnei.

3.2.1 Produkto įvertinimas

3.2.1.1 Projekto trukmė

Projektų etapų trukmės pateiktos 26 pav. Projekto trukmės lentelė.

Projekto etapai	Trukmės (valandomis)						
		1 iteracija	2 iteracija	3 iteracija	4 iteracija	5 iteracija	
Reikalavimų surinkimas	6						
Susirinkimas							
Architektūros nustatymas		5					
Iteracinis kūrimas			56	48	32	32	40
Galutinio produkto pristatymas							

26 pav. Projekto trukmės lentelė

Projekto trukmė – 219 valandų (~28 dienos). Kadangi galutinis produktas nepristatytas, dar numatomos maždaug dvi iteracijos, kurių trukmės apytiksliai po 24 valandas (po 3 dienas). Taigi, bendra projekto trukmė numatoma 34 dienos.

Kuriama svetainė yra vidutinio dydžio: puslapių skaičius yra daugiau nei 35 psl., užsakymo sistema bei kitos funkcijos, kur atliekami veiksmai su duomenų baze (žiūrėti 20 pav. Panaudos atvejų diagrama). Galima teigti, jog svetainės kūrimas, panaudojant kuriamą *Agile* metodą, yra greitas.

3.2.1.2 Vartotojų įvertinimas

Kokybiškas vartotojo poreikius tenkinantis produktas parodo taikyto metodo gerumą. Vartotojo teikiama nuomonė – produkto įvertinimas. Užsakovas pateikė oficialų raštą-atsiliepimą apie svetainę „Jadvygos žolės“, kuris pateiktas 27 pav. Užsakovo atsiliepimai.

Ūkininkė Jadvyga Balvočiūtė
Gyvolių k.
LT-89483 Mažeikių r.

2008-01-11
Gyvoliai

Kauno technologijos universiteto
Informatikos fakulteto
Informacijos sistemų katedrai

Dėl Viktorijos Šilinskaitės kurtos svetainės www.jadvyga.lt

Dėkoju Viktorijai Šilinskaitei už jos atliktą darbą kuriant svetainę mano ūkio produkcijai pristatyti. Viktorija Šilinskaitė atsižvelgė į visus pageidavimus, kūrybiškai teikė siūlymus, kad svetainė būtų funkcionali, taip pat ji padėjo atrinkti fotografijas ir pasiūlė kokybišką dizainą.

Mūsų bendravimą su kitomis įmonėmis nuo šiol labai palengvina vidinė žinučių siuntimo sistema, galimybė bendrauti elektroniniu paštu su kitais mūsų produkcija besidominčiais vartotojais, įskaitant vartotojus iš kitų ES šalių.

Dar iki svetainės baigimo, be specialios reklamos ją rado ne vienas žmogus, ieškantis informacijos apie galimybę susitikti su manimi konsultacijai, apsilankyti ūkyje ar užsisakyti žolelių arbatų, tapo lengviau bendrauti su klientais. Dėl gana didelio žolelių arbatų asortimento, arbatų sudėtingumo yra labai svarbūs glaudūs kontaktai su žmonėmis, atsiliepimai, kad galėtume tobulinti produktus, parinktume geriausius iš galimų variantų. Todėl džiaugiamės, kad vartotojai galės lengvai rašyti atsiliepimus.

Yra labai gerai, kad nuo šiol galime turėti visiems prieinamą prekių katalogą ir jį galime nesunkiai koreguoti. Taip pat galime geriau informuoti apie būsimas muges, kurių metu mus gali rasti klientai tuose rajonuose, kuriuose nėra mūsų produkcija prekiaujančių parduotuvių. Mugių kalendorių nesunku redaguoti.

Buvo lengva su Viktorija Šilinskaite derinti darbus ir reikalingus pataisymus ir tai buvo ypač aktualu, kadangi tai pirmoji svetainė mūsų ūkiui ir neturėjome paruošę iš karto visos medžiagos, kurią reikia pateikti. Svetainės kūrimas buvo sklandus ir greitas, esame patenkinti, kad Viktorija Šilinskaitė pasisiūlė mums padėti dėl svetainės kūrimo.

Ūkininkė, farmacininkė-farmakognostė

Jadvyga Balvočiūtė



27 pav. Užsakovo atsiliepimai

3.2.1.3 Internetinės svetainės „Jadvygos žolės“ kokybės pagal ISO 9126 standartą įvertinimas

ISO 9126 – tai tarptautiniai programinės įrangos kokybės vertinimo standartai.

Svetainės „Jadvygos žolės“ kokybė pagal ISO 9126 standartus vertinama penkių balų sistemoje, atsižvelgiant į tai, kad produktas dar nėra iki galo užbaigtas.

27 lentelė Svetainės „Jadvygos žolės“ kokybės įvertinimas

Charakteristikos	Savybės	Balas
Funkcionalumas (<i>Functionality</i>)	Tinkamumas (<i>Suitability</i>)	5
	Tikslumas (<i>Accuracy</i>)	5
	Vidinis veikimas (<i>Interoperability</i>)	5
	Reikalavimų atitikimas (<i>Compliance</i>)	5
	Saugumas (<i>Security</i>)	4
Patikimumas (<i>Reliability</i>)	Užbaigtumas (<i>Maturity</i>)	4
	Atstatytinumas (<i>Recoverability</i>)	5
	Klaidų toleravimas (<i>Fault Tolerance</i>)	4
Panaudojamumas (<i>Usability</i>)	Įsisavinimas (<i>Learnability</i>)	5
	Suprantamumas (<i>Understandability</i>)	5
	Veiksmingumas (<i>Operability</i>)	5
Efektyvumas (<i>Efficiency</i>)	Laiko režimas (<i>Time Behaviour</i>)	5
	Resursų režimas (<i>Resource Behaviour</i>)	5
Palaikomumas (<i>Maintainability</i>)	Stabilumas (<i>Stability</i>)	5
	Analizuojamumas (<i>Analyzability</i>)	5
	Kintamumas (<i>Changeability</i>)	5
	Testuojamumas (<i>Testability</i>)	5
Perkeliamumas (<i>Portability</i>)	Instaliuojamumas (<i>Installability</i>)	5
	Pakeičiamumas (<i>Replaceability</i>)	5
	Pritaikomumas (<i>Adaptability</i>)	5

4 Rezultatai ir išvados

1. Išanalizavus sistemų kūrimo modelius ir metodus nustatyta, kad *Agile* požiūris yra vienas iš tinkamiausių efektyviam internetinių svetainių kūrimui.
2. Atlikta analizė parodė, kad internetinių svetainių kūrime FDD ir *Scrum* metodai nėra visiškai tinkami naudoti tokie, kokie jie yra, todėl reikalinga sukurti metodą, paremtą FDD ir *Scrum* metodų savybėmis, kuris pagerintų internetinių svetainių kūrimo procesą.
3. Pritaikant žinomas teorijas, autoriaus patirtį ir praktiką kuriant svetainę, pagal *Agile* metodologiją sukurtas metodas, paremtas FDD ir *Scrum* metodais, skirtas internetinių svetainių kūrimui.
4. Internetinės svetainės įmonei „Jadvygos Žolės“ kūrimas darbe pateiktu metodu parodė, kad metodas pasižymi tokiomis savybėmis:
 - Greitas kūrimas;
 - Lankstumas;
 - Efektyvus reagavimas į reikalavimų pasikeitimus;
 - Nuolatinis bendradarbiavimas su užsakovu;
 - Efektyvus bendravimas tarp komandos narių;
 - Dėmesys vartotojo sąsajai;
 - Dėmesys funkcionalumui;
 - Trumpos iteracijos;
5. Kadangi svetainė buvo kuriama vieno asmens (kūrėjas atliko visus vaidmenis: dizaineris, architektas, programuotojas, testuotojas, vadovas), nėra galimybės pasakyti, kokie galimi rezultatai, jei kurtų komanda.

5 Literatūra

- [1] G. Preston, S. R. Pichler. Agile Risks, Agile Rewards. [žiūrėta 2007-10-20]. Prieiga per Internetą: <http://www.ddj.com/dept/architect/184415308>
- [2] A. McDonald, R. Welland. Agile Web Engineering (AWE) Process. [žiūrėta 2007-10-20]. Prieiga per Internetą: <http://www.dcs.gla.ac.uk/publications/PAPERS/7087/TR-2001-98%5B1%5D.pdf>
- [3] D. Šilingas. Programinės įrangos reikalavimų valdymo principai ir praktika. [žiūrėta 2006-10-20]. Prieiga per Internetą: <http://www.bpi.lt/text.php?lang=1&item=161&arg=119>
- [4] A. Abran, J. W. Moore. Software Requirements. [žiūrėta 2007-10-25]. Prieiga per Internetą: <http://www.swebok.org/ch2.html>
- [5] S. Nuseibeh, Requirements Engineering: A Roadmap. [žiūrėta 2007-10-25]. Prieiga per Internetą: http://www.doc.ic.ac.uk/~ban/pubs/sotar_re.pdf
- [6] M. Aoyama. Web-Based Agile Software Development. [žiūrėta 2007-10-27]. Prieiga per Internetą: <http://ieeexplore.ieee.org/iel4/52/15788/00730844.pdf?arnumber=730844>
- [7] S. Murugesan, A. Ginige. Web Engineering: Introduction and Perspectives, 2005 [žiūrėta 2007-11-06]. Prieiga per Internetą: <http://www.idea-group.com/downloads/excerpts/01%20Suh.pdf>
- [8] [May 2007 Web Server Survey](http://news.netcraft.com/archives/web_server_survey.html). [žiūrėta 2007-11-06]. Prieiga per Internetą: http://news.netcraft.com/archives/web_server_survey.html
- [9] [S. Allen](http://www.ultrasaurus.com/sarablog/archives/000160.html). Agile web site development. [žiūrėta 2007-11-09]. Prieiga per Internetą: <http://www.ultrasaurus.com/sarablog/archives/000160.html>
- [10] R. Butkienė. “Vikrūs” programų sistemų kūrimo metodai. [žiūrėta 2007-11-20]. Prieiga per Internetą: ftp://isd.ktu.lt/isd/Butkiene/T120M051/2_Agile_Metodologijos.ppt
- [11] J. Bach. [Defining Agile Methodology](http://www.satisfice.com/blog/archives/45). [žiūrėta 2007-11-20]. Prieiga per Internetą: <http://www.satisfice.com/blog/archives/45>
- [12] Agile Alliance. [žiūrėta 2006-11-21]. Prieiga per Internetą: <http://agilemanifesto.org/principles.html>
- [13] Agile Alliance. Manifesto for Agile Software Development. [žiūrėta 2007-06-20]. Prieiga per Internetą: <http://www.agilemovement.it/index.php?newlang=eng>
- [14] [Scott W. Ambler](http://www.agilemodeling.com). Agile Modeling and eXtreme Programming (XP). Agile modeling. March 6, 2006 [žiūrėta 2007-06-21]. Prieiga per Internetą: <http://www.agilemodeling.com>
- [15] [James Woodger](http://www.woodger.ca/archweb.htm#enterprise). General Web Architecture. [žiūrėta 2007-06-21]. Prieiga per Internetą: <http://www.woodger.ca/archweb.htm#enterprise>
- [16] [A. Izaz](http://www.lawdit.co.uk/reading_room/room/view_article.asp?name=../articles/Website%20development2.htm). Website development- common problem areas. 2006-05-30. [žiūrėta 2007-09-15]. Prieiga per Internetą: http://www.lawdit.co.uk/reading_room/room/view_article.asp?name=../articles/Website%20development2.htm
- [17] E. Holter. Web Development Fallacies. 2006, Liepa. [žiūrėta 2007-09-15]. Prieiga per Internetą: http://www.newfangled.com/website_development_difficulties
- [18] J. Ruževičius, N. Guseva. Interneto svetainių kokybės vertinimo ypatumai. ISSN 1392–1258. EKONOMIKA 2006. [žiūrėta 2007-11-03]. 77- 90 psl. Prieiga per Internetą: http://www.leidykla.eu/fileadmin/Ekonomika/75/Juozas_Ruzevicius_Natalija_Guseva.pdf
- [19] [DotNetNuke](http://www.dotnetnuke.com/guidedtour/WebsiteManageUsers.htm). Manage Your Website Users. 2007. [žiūrėta 2007-12-06]. Prieiga per Internetą: <http://www.dotnetnuke.com/guidedtour/WebsiteManageUsers.htm>
- [20] R. Butkienė. IS gyvavimo ciklo modeliai ir kūrimo metodikos. [žiūrėta 2007-12-14]. Prieiga per Internetą: ftp://isd.ktu.lt/isd/Butkiene/T120M051/1_Metodikos.ppt
- [21] Lietuvių kalba informacinėse technologijose. Kompiuterinės leksikos aiškinamasis žodynas. [žiūrėta 2007-12-16]. Prieiga per Internetą: <http://www.likit.lt/term/z2odynas.html>

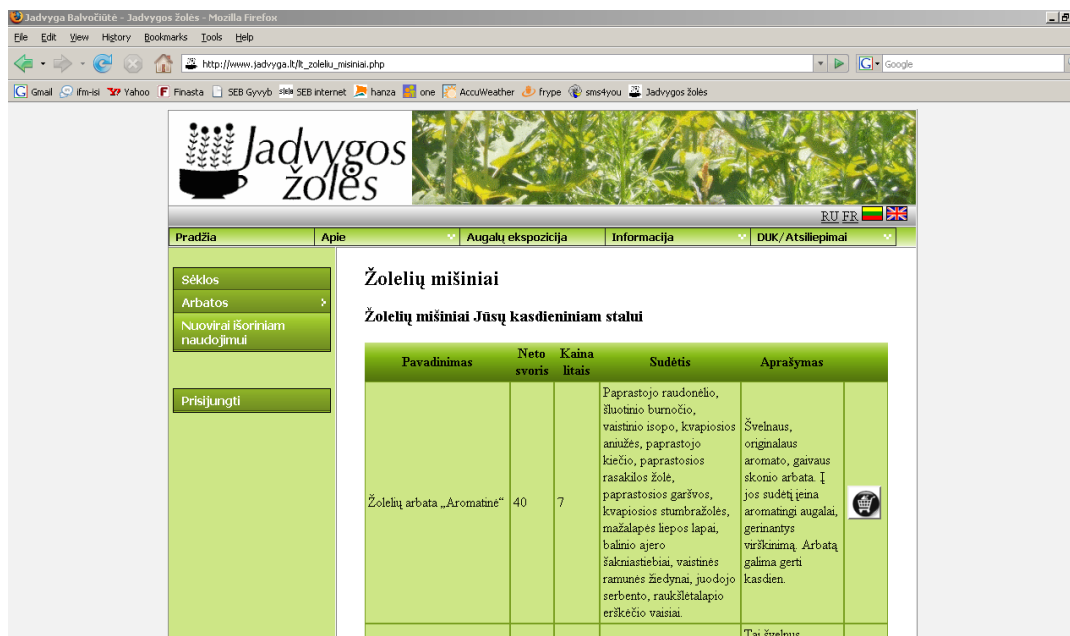
6 Priedai

Priedas 1 Internetinės svetainės pavyzdžiai

Vartotojo sąsajos langų pavyzdžiai pateikti 28 pav. - 35 pav.



28 pav. Vartotojo sąsajos langas



29 pav. Vartotojo sąsajos langas. Žolelių mišiniai



30 pav. Vartotojo sąsajos langas. Vartotojų tvarkymas



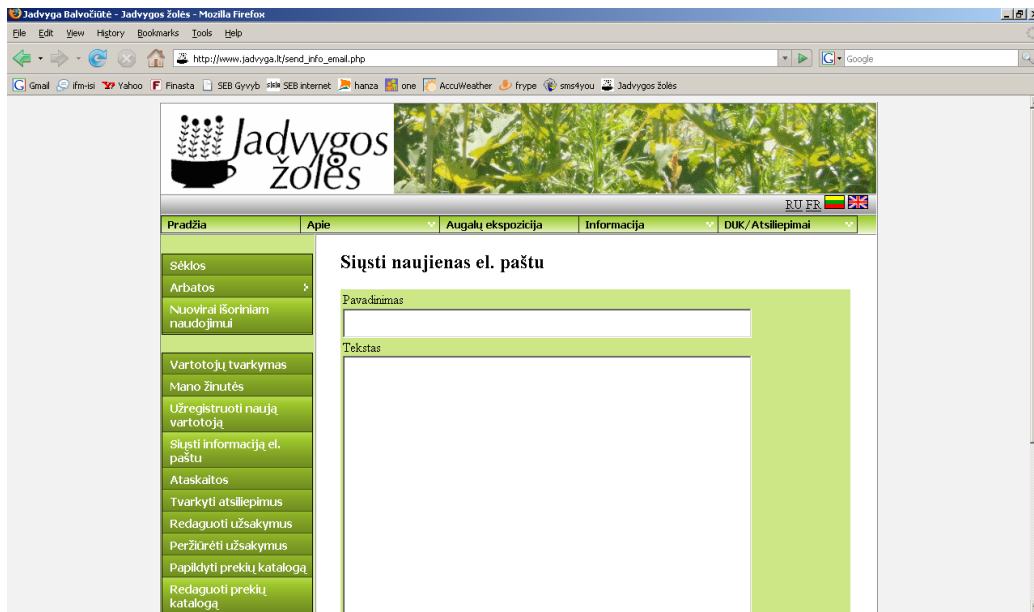
31 pav. Vartotojo sąsajos langas. Užregistruoti naują vartotoją



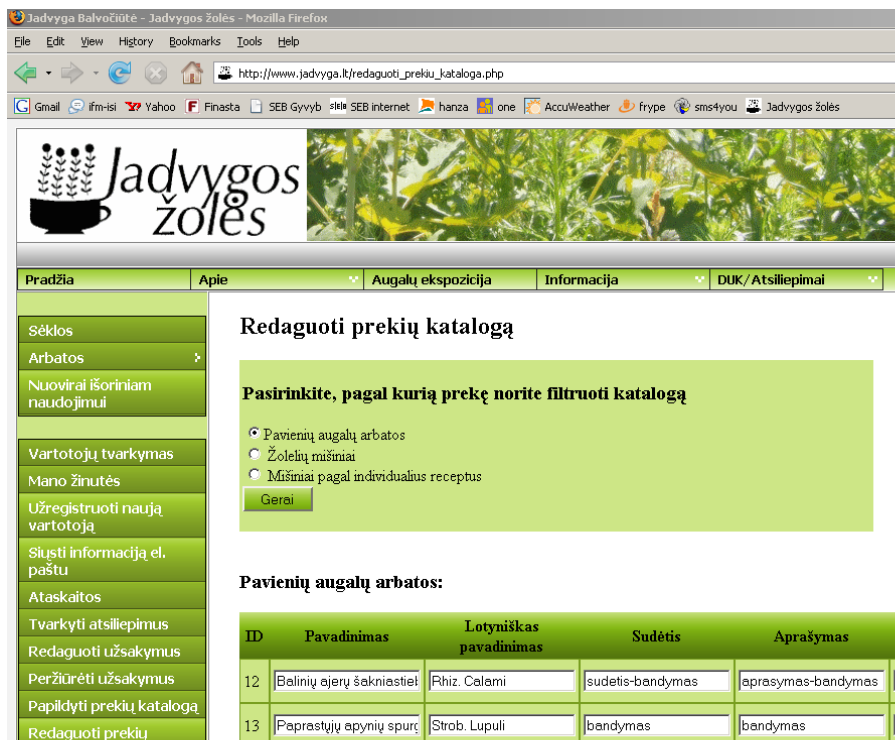
32 pav. Vartotojo sąsajos langas. Rašyti naują pranešimą



33 pav. Vartotojo sąsajos langas. Mano žinutės



34 pav. Vartotojo sąsajos langas. Siųsti naujienas el. paštu



35 pav. Vartotojo sąsajos langas. Redaguoti prekių katalogą