

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

JUSTAS TAMOŠAITIS

BENDRUOMENINIŲ PROJEKTŲ KŪRIMO PROCESŲ
TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas
doc. dr. R. Damaševičius

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

JUSTAS TAMOŠAITIS

BENDRUOMENINIŲ PROJEKTŲ KŪRIMO PROCESŲ
TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas
doc. dr. R. Damaševičius
2013-05-

Recenzentas
lekt. dr. Š. Packevičius
2013-05-

Atliko:
IFME-1/2 gr. studentas
Justas Tamošaitis
2013-05-

KAUNAS, 2013

AUTENTIŠKUMO PATVIRTINIMAS

Patvirtinu, kad įteikiamas baigiamasis darbas „Bendruomeninių projektų kūrimo procesų tyrimas“:

1. Autoriaus atliktas savarankiškai, jame nėra pateikta kitų autorių medžiagos kaip savos, nenurodant tikrojo šaltinio.
2. Nebuvo to paties autoriaus pristatytas ir gintas kitoje mokymo įstaigoje Lietuvoje ar užsienyje.
3. Nepateikia nuorodų į kitus darbus, jeigu jų medžiaga nėra naudota darbe.
4. Pateikia visą naudotos literatūros sąrašą.

Justas Tamošaitis
(studento vardas, pavardė)

2013-05-
(data)

(parašas)

SANTRAUKA

Bendruomeninių paslaugų apibrėžimas dar nėra galutinai susiformavęs ir naudojami įvairūs jo variantai priklausomai nuo konteksto. Tačiau matoma galimybė panaudoti bendruomeninius resursus įvairiose srityse, įskaitant ir programinės įrangos projektų kūrimą. Viena iš svarbių projektų kūrimo sričių yra projektų valdymas.

Siekiant apjungti projektų valdymą su bendruomeninėmis paslaugomis, darbe nagrinėjamas jų integracijos variantas. Aptariami naudojami programų kūrimo modeliai, apžvelgiami susiję sprendimai, jų teikiamas funkcionalumas.

Darbe pateikiamas įgyvendintos projektų valdymo sistemos su integruotomis bendruomeninėmis paslaugomis aprašas, jos esminės savybės. Atliekamas eksperimentinis bendruomeninių projektų kūrimo savybių tyrimas, kurio metu tiriamas bendruomenės narių indėlis projekto maste, bei narių prisijungimo prie projekto tendencijos.

SUMMARY

The definition of crowdsourcing is not yet fully formed and is used in various versions, depending on the context. However, there is the possibility crowdsourcing resources in various fields, including software development projects. One of the most important parts of projects is the project management.

In order to combine project management with crowdsourcing, integration options are analyzed in the paper. Programming models, a review of decisions, they provided functionality is discussed in it.

The paper provides implementation of the project management system with integrated crowdsourcing services and a description of its essential features. The experimental community project development process analysis is conducted to investigate the contribution size of community members and the trend of new members joining the projects.

TURINYS

Lentelių sąrašas	9
Paveikslų sąrašas.....	10
Terminų ir santrumpų žodynas	12
1 Įvadas	13
1.1 Bendruomeninių paslaugų apibrėžimai.....	13
1.2 Taikymo sritys.....	14
1.3 Aktualumas	14
1.4 Kolektyvinis intelektas.....	14
1.5 Darbo tikslas	14
2 Programinės įrangos kūrimas naudojant bendruomenines paslaugas	15
2.1 Bendruomeninio darbo pasidalinimo modeliai	15
2.1.1 Atviri/uždari	15
2.1.2 Atviro kodo sistemos	15
2.1.3 Užsakomosios paslaugos.....	15
2.1.4 Laisvai samdomi darbuotojai	16
2.1.5 Bendruomeninių sistemų vartotojai	16
2.2 Bendruomeninių sistemų modelių ar architektūrų analizė.....	16
2.2.1 Komponentai.....	16
2.2.2 Funkcijos.....	17
2.2.3 Užduočių pateikimas kandidatams.....	18
2.2.4 Užduoties atlikėjo parinkimas.....	19
2.3 Bendruomeninių paslaugų apžvalga	20
2.3.1 Microworkers	20
2.3.2 Amazon MTurk.....	20
2.3.3 oDesk	20
2.3.4 TopCoder	21
2.3.5 Sistemų palyginimas	22
2.3.6 Problemos.....	22
2.4 Bendruomeninių sistemų ir paslaugų vertinimo metodų apžvalga	23
2.4.1 Kokybiniai vertinimo metodai (apklausos ir pan.).....	24
2.4.2 Kiekybiniai vertinimo metodai (programų metrikos)	24
2.5 PVS savybių apžvalga.....	25
2.5.1 Clarizen	25
2.5.2 Redmine	25
2.5.3 Projectpier	25
2.5.4 Sistemų palyginimas	25
2.6 Duomenų sinchronizavimo metodų apžvalga	26
2.7 Įmonės organizacinis modelis.....	26

2.8	Modelio tinkamo programinės įrangos kūrimui panaudojant bendruomenines paslaugas variantas	27
2.9	Išvados	27
3	PVS su integruotomis bendruomeninėmis paslaugomis	27
3.1	Architektūros tikslai ir apribojimai	27
3.2	Sistemos išdėstymo vaizdas	28
3.3	Sistemos dinaminis vaizdas	29
3.4	Komerciniai specializuoti programų paketai	30
3.5	Sistemos veiklos diagramos	31
3.5.1	Duomenų valdymas.....	31
3.5.2	Pateikimas bendruomenei	32
3.6	Svarbesnių paketų detalizavimas	33
3.6.1	„Trogon“ paketo detalizavimas.....	33
3.6.2	„Solutions.PMS“ paketo detalizavimas	34
3.6.3	„Solutions.CrowdSourcing“ ir „Solutions.oDesk“ paketų detalizavimas.....	35
3.7	Bendruomeninių paslaugų integravimas	36
3.8	Reaguojantis dizainas.....	36
3.9	Objektiškai orientuoti stiliai.....	37
3.10	Veikimo aplinka.....	37
3.11	Išvados	37
4	Trogon sistemos tyrimas	37
4.1	Specifikacijos atitikimas	37
4.2	Testavimas	37
4.2.1	Statinė kodo analizė	37
4.3	„oDesk“ API problemos.....	41
4.4	Sistemos tobulinimai ateityje	42
4.4.1	Sinchronizavimas	42
4.5	Išvados	42
5	Bendruomeninių projektų kūrimo procesų tyrimas	43
5.1	Tikslai.....	43
5.2	Duomenų parinkimas	43
5.3	Duomenų paruošimas.....	44
5.4	Pradinė projekto stadija.....	44
5.4.1	Užduočių atlikimo kitimas	44
5.4.2	Autorių kiekio kitimas	46
5.4.3	Autoriai ir jų atliekamos užduotys	47
5.5	Projektas visoje eigoje	47
5.5.1	Bootstrap	48
5.5.2	jQuery.....	49

5.5.3	Mono	50
5.5.4	Spree.....	52
5.5.5	Typo	53
5.5.6	Apibendrinta projektų eigos informacija	54
5.6	Išvados	54
6	Išvados	55
7	Literatūra.....	56
8	Priedai	58
8.1	Užduočių atlikimo grafikai	58
8.2	„Kitokia programinės įrangos įmonė“	63
8.3	„Projektų valdymo sistemos išplėtimas bendruomeninėmis paslaugomis“	67

LENTELIŲ SĄRAŠAS

2.1 lentelė. Bendruomeninių paslaugų sistemų palyginimas	22
2.2 lentelė. Projektų valdymo sistemų palyginimas.....	25
4.1 lentelė. Komponentų kodo eilučių skaičius	38
5.1 lentelė. Eksperimentinio tyrimo projektai.....	43
5.2 lentelė. Pirmųjų trijų mėnesių užduočių kiekiai projektuose	45
5.3 lentelė. Pirmojo mėnesio užduočių autoriai ir jų atliktos užduotys	47
5.4 lentelė. Pirmųjų trijų mėnesių užduočių autoriai ir jų atliktos užduotys	47
5.5 lentelė. Projektų eigos apibendrinta informacija.....	54

PAVEIKSLŲ SĄRAŠAS

2.1 pav. Bendruomeninių paslaugų sistemų komponentai ir funkcijos (pagal Hetmank (2013)).....	16
2.2 pav. Įmonės resursų struktūra (pagal Ašeriškį ir Tamošaitį (2012)).....	26
3.1 pav. Aukšto lygio išdėstymo diagrama	28
3.2 pav. Bendra sistemos paketų diagrama	29
3.3 pav. Veiklos diagrama. Apibendrintas duomenų valdymas.....	31
3.4 pav. Veiklos diagrama. Užduoties pateikimas bendruomenei	32
3.5 pav. „Trogon“ paketo detalizuota klasių diagrama	33
3.6 pav. „Solutions.PMS“ paketo detalizuota klasių diagrama.....	34
3.7 pav. „Solutions.CrowdSourcing“ ir „Solutions.oDesk“ paketų detalizuota klasių diagrama	35
3.8 pav. PVS sąveika su bendruomeninių paslaugų sistema (Ašeriškis & Tamošaitis, 2012)	36
4.1 pav. Sistemos komponentų prižiūrimumo indekso absoliutinės reikšmės.....	39
4.2 pav. Sistemos komponentų ciklo matinio sudėtingumo absoliutinės reikšmės	39
4.3 pav. Sistemos komponentų paveldėjimo gylio absoliutinės reikšmės	40
4.4 pav. Sistemos komponentų klasių susietumo absoliutinės reikšmės	40
4.5 pav. Sistemos komponentų tarpusavio ryšiai	41
5.1 pav. Užduočių pateikimo dažnumas pirmojo mėnesio laikotarpiu	44
5.2 pav. Užduočių pateikimo dažnumas pirmųjų trijų mėnesių laikotarpiu	45
5.3 pav. Autorių kiekiai pirmojo mėnesio laikotarpiu	46
5.4 pav. Autorių kiekiai pirmųjų trijų mėnesių laikotarpiu.....	46
5.5 pav. „Bootstrap“ vartotojų atlikti užduočių kiekiai	48
5.6 pav. „Bootstrap“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu	48
5.7 pav. „jQuery“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 101, dešimčių grupėmis, imant vidurkį).....	49
5.8 pav. „jQuery“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu.....	49
5.9 pav. „Mono“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 1001, šimto grupėmis, imant vidurkį).....	50
5.10 pav. „Mono“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu	51
5.11 pav. „Spree“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 101, dešimčių grupėmis, imant vidurkį).....	52
5.12 pav. „Spree“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu.....	52
5.13 pav. „Typo“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 101, dešimčių grupėmis, imant vidurkį).....	53
5.14 pav. „Typo“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu	53
8.1 pav. „Bootstrap“ užduočių atlikimas (dienos intervalais) viso projekto metu.....	58
8.2 pav. „Bootstrap“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu	58
8.3 pav. „jQuery“ užduočių atlikimas (dienos intervalais) viso projekto metu	59
8.4 pav. „jQuery“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu	59
8.5 pav. „Mono“ užduočių atlikimas (dienos intervalais) viso projekto metu.....	60
8.6 pav. „Mono“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu	60

8.7 pav. „Spre“ užduočių atlikimas (dienos intervalais) viso projekto metu	61
8.8 pav. „Spre“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu	61
8.9 pav. „Typo“ užduočių atlikimas (dienos intervalais) viso projekto metu.....	62
8.10 pav. „Ttypo“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu.....	62

TERMINŲ IR SANTRUMPŲ ŽODYNAS

- Daugialypė Rasch skalė – angl. „Multi-faceted Rasch Scaling”.
- JQuery – tarpnaršyklinė javascript biblioteka.
- JQuery ui – tarpnaršyklinė javascript biblioteka skirta animacijoms ir vaizdiniais objektams.
- oDesk API – bendravimo su oDesk platforma sąsaja teikiama oDesk (servisas).
- DotNetOpenAuth – biblioteka leidžianti atlikti autentifikavimas pagal OAuth 1.0 protokolą, naudojamą oDesk API.
- WikiCreola – .NET skirtas Wikipedia sintaksės variklis.
- API – angl. „Application Programming Interface”, aplikacijų programavimo sąsaja.
- PVS – projektų valdymo sistema.
- Laisvai samdomų darbuotojų paslaugos – angl. „Freelancing”.
- Užsakomosios paslaugos – angl. „Outsourcing”.
- Bendruomeninės paslaugos – angl. „CrowdSourcing”.
- OOCSS – angl. „Object Oriented Cascading Style Sheets”, objektiškai orientuoti stiliai

1 ĮVADAS

Bendruomeniniam mąstymui vis labiau skverbiantis į mūsų gyvenimą, bendruomenių teikiamus privalumus bandoma panaudoti įvairiose srityse. Viena iš tokių sričių yra projektų valdymas. Apjungiant kelias sritis į visumą ar vien tik papildant atskirais elementais, siekiama sukurti naujus ar patobulinti esamus veiklos modelius. Išsiaiškinus galimus bendruomeninių projektų kūrimo procesus galima įvertinti jų naudingumą ir galimybes.

1.1 Bendruomeninių paslaugų apibrėžimai

Pirmasis bendruomeninių paslaugų terminą paminėjo Howe (2006), apibrėždamas tai kaip: užduočių, kurias tradiciškai atlikdavo konkretus subjektas (paprastai darbuotojas), perdavimas neapibrėžtai, dažniausiai didelei, grupei žmonių („miniai“). Čia svarbu, kad kvietimas atlikti užduotį būtų atviro kvietimo forma, o ne konkreitiems subjektams.

Vėliau Howe pasiūlė tokius du apibrėžimus (kaip cituota Schenk & Guittard, 2011):

- Bendruomeninės paslaugos yra darbo, kurį tradiciškai atlikdavo paskirtas atstovas (paprastai darbuotojas), paėmimas ir kaip užsakomųjų paslaugų pateikimas neapibrėžtai, paprastai didelei žmonių grupei, atviro kvietimo forma.
- Atviro kodo principų taikymas srityse nesusijusiose su programine įranga.

Minia šiuo atveju apibrėžiama kaip didelis būrys įvairios patirties ir gebėjimų anonimiškų asmenų linkusių atlikti ar prisidėti prie pateiktų užduočių atlikimo.

Bendruomeninių paslaugų vykdyme paprastai dalyvauja trys grupės (Schenk & Guittard, 2011):

- Tiekėjai – asmenys sudarantys minią.
- Užsakovų įmonė – įmonės gaunančios tiesioginę naudą iš minios indėlio.
- Bendruomeninių paslaugų įgalintojas – tarpinė platforma sukurianti ryšį tarp tiekėjų ir užsakovų įmonių.

Bendruomeninės paslaugos skirstomos į integravimo ir selektyviasias (Schenk & Guittard, 2011):

- Integravimo bendruomeninės paslaugos – individualūs elementai turi labai mažą reikšmę rezultatui, tačiau apjungtas didelis jų kiekis teikia realią naudą, t.y. išnaudojamas minios dydis.
- Selektyviosios bendruomeninės paslaugos – išnaudojami individualūs minios gebėjimai, kurių dėka sukuriami sprendimo variantai, iš kurių galima rinktis tinkamiausią.

Kadangi terminas pradėtas naudoti įvairiose srityse, priklausomai nuo konteksto mokslinėse publikacijose išskiriami kiti apibrėžimai. Hetmank (2013) naudojamus bendruomeninių paslaugų sistemų apibrėžimus pagal perspektyvas sugrupavo į keturias kategorijas:

1. Organizacinė perspektyva – išryškinamas tarpininko vaidmuo tarp užduoties pateikėjo ir galimų jos atlikėjų.
2. Techninė perspektyva – apibrėžiami programinės įrangos komponentai, techninės funkcijos ar duomenų objektai, kurie sudaro įgyvendinamas bendruomeninių paslaugų sistemas.
3. Proceso perspektyva – aprašo veiksmus atliekamus su sistemos objektais ar vartotojais.
4. Į žmogų orientuota perspektyva – pabrėžia žmogaus intelektą ir kolektyvinį intelektą, kaip esmines dalis sistemoje.

1.2 Taikymo sritys

Bendruomeninės paslaugos taikomos tiek paprastoms, tiek ir sudėtingoms užduotims atlikti. Jos taikomos įvairiose srityse, kurių dalis yra apžvelgiamos ir įvairiose publikacijose (Brabham, 2008, bei Schenk & Guittard, 2011):

- IT – dizainai, algoritmų kūrimas, sistemų įgyvendinimas, testavimas.
- Ekologija – eko-idėjų generavimas ir išdirbimas.
- Gamyba, marketingas – marškinėlių dizainas, virusinis marketingas, reklamų kūrimas.
- Turinio generavimas ir apdorojimas – mikro-vertimai, paveikslėlių žymėjimas ir vadinimas, žemėlapių vietovių sudarymas.
- Moksliniai tyrimai – ligos biologinių žymeklių sukūrimas, duomenų perrinkimas.
- Sprendimai kitoms nestandartinėms situacijoms – naftos vamzdyno taisymas po katastrofos vandenyno dugne.
- Kita.

Pasinaudojant bendruomeninėmis paslaugomis galima įgyvendinti nebūtinai visą projektą, bet ir vien tik jo dalį.

1.3 Aktualumas

Žmonės jau nuo seno jungėsi į bendruomenes, tačiau per pastarąjį dešimtmetį populiarėjant internetui ir jam tampant neatskiriama žmonių gyvenimo dalimi internetu paplito naujas – internetinių bendruomenių – virusas. Šiandien žmonės dažnai priklauso kelioms virtualioms, internetinėms bendruomenėms, kurios gali būti socialinės (skaitmeninėje erdvėje jungiančios draugus ir pažįstamus), ideologinės (jungiančios panašių idėjų žmones) ir tikslinės (jungiančios žmones, turinčius panašius tikslus).

Net ir mažos įmonės vis dažniau dalyvauja tarptautiniuose bendradarbiavimo projektuose. Dažnai juntamas resursų stygius. Įvairioms problemoms efektyviai ar išvis išspręsti nepakanka konkrečių turimų samdomų darbuotojų galimybių.

Norint kurti projektus įjungiant šią papildomą jėgą į įmonės veiklą, gali tekti keisti ar pritaikyti jos procesus. Tam reikia išsiaiškinti galimos sistemos reikalavimus efektyviam resursų panaudojimui, tinkamus naudoti procesus ir resursų galimybes.

1.4 Kolektyvinis intelektas

Bendruomeninių paslaugų viena iš stipriųjų dalių yra kolektyvinis intelektas. Kolektyvinis intelektas – visos minios bendras intelektas. Kaip apžvelgiama Brabham (2008) – minios pasiektas vidutinis rezultatas (sprendžiant intelektines užduotis), paprastai bus bent toks pat geras, kaip protingiausio asmens imtyje. Skirtingų sričių ir patirčių atstovų žinių surinkimas ir apjungimas į vieną sukuria minios išmintį. Pernešant vienose disciplinose naudojamus būdus spręsti problemas į kitas sritis, jie gali būti naujove ir išspręsti problemas.

Žinoma nereikia pamiršti ir galimo neigiamo bendruomeninių paslaugų naudojimo efekto. Minia gali nepritari ti keliamam tikslui ir kurti priešingą norimam efektą, pasinaudodama suteikta galimybe.

1.5 Darbo tikslas

Darbo tikslas – pateikti bendruomeninių paslaugų integravimo į projektų valdymo sistemą (PVS) variantą. Apžvelgti jam tinkamos projekto struktūros modelį. Ištirti bendruomeninių projektų kūrimo procesą – bendruomenės narių pateikiamų sprendimų pasiskirstymą projekto maste.

2 PROGRAMINĖS ĮRANGOS KŪRIMAS NAUDOJANT BENDRUOMENINES PASLAUGAS

2.1 Bendruomeninio darbo pasidalinimo modeliai

2.1.1 Atviri/uždari

Programinės įrangos įgyvendinimui gali būti panaudoti atviri ar uždari darbo pasidalinimo modeliai. Uždaras darbo pasidalinimo modelis yra, kai darbas atliekamas tiesiog įmonėje, įmonės darbuotojų. Atviru laikoma, kai kūrimas atliekamas bendruomenės. Tarpinis variantas yra, kai dalis įgyvendinimo yra perduodama atlikti bendruomenei, t.y. suteikiama galimybė dalyvauti kūrime kažkam, ne iš įmonės.

2.1.2 Atviro kodo sistemos

Sistemos, kurios yra laisvai platinamos kartu su programiniu kodu ir atitinka atviro kodo apibrėžimą. Kaip apibendrina Vasa (2010), laisvas platinimas apima keturias dalis:

1. Nemokama.
2. Galima laisvai, be apribojimų modifikuoti.
3. Galima laisvai platinti.
4. Galima naudoti programinę įrangą taip kaip norima.

Kiekvienas norintis gali pasinaudoti kodu (licencijos leidžiamose ribose), kurti pakeitimus, tobulinti. Egzistuoja keletas tokių sistemų vystymo variantų. Vienas jų yra, kai leidžiama visiems pateikti atnaujinimus, kuriuos bendruomenė įvertina ir jei pakeitimai tinkami, jie perkeliama į sistemą. Kitas variantas, kai už įjungimą į sistemą yra atsakinga branduolio grupė.

Norint atskirti bendruomeninių paslaugų kuriamos sistemos apibrėžimą, labiau tinkamas naujai taikomas Free-Libre-Open Source Software arba FLOSS (Schenk & Guittard, 2010). Pastarasis terminas apima anksčiau minėtą atviro kodo apibrėžimą. Skirtingai nuo bendruomeninių paslaugų kuriamos sistemos įmonėms, čia dažniausiai taikomas ne patentinis galutinio produkto licencijavimas, bet paliekama galimybė jį toliau laisvai vystyti.

2.1.3 Užsakomosios paslaugos

Užsakomosios paslaugos – tai darbų pateikimas atlikti kažkam kitam (už įmonės ribų) – darbo resursai yra konkreti išorinė organizacija/komanda. Taip įgyvendinamas visas projektas ar jo apibrėžta dalis.

Naudingas dėl galimo darbuotojų ar darbuotojų su reikiamais gebėjimais trūkumo vietoje, ar mažesnių kaštų įgyvendinant projektą kitur – pigesnė darbo jėga kitoje šalyje. Išorinis įgyvendintojas gali pasiūlyti naujas idėjas ir sprendimus, paspartinti procesą.

Tačiau galimas ir neigiamas variantas – prasta kodo kokybė, nesusitarimas tarpusavyje tarp to kas turi būti įgyvendinta, dideli komunikacijos su įgyvendintoju kaštai, atsirandantys dėl skirtingos darbo vietos bei laiko ar kultūros.

Norint valdyti procesą, reikalingus įgyvendinti darbus, bei jų eigą, reikia registruoti ir derinti veiklą su pagrindine įmone. Reikalinga darbų ataskaitų sistema, darbo progreso ataskaitos. Skirtingos organizacijos gali naudoti skirtingus procesus, kurios reikia suderinti. Tam reikia pereiti prie bendros darbų registravimo sistemos, ar atlikti esminių taškų sinchronizavimą naudojamose sistemose.

Įgyvendintas dalis reikia apjungti į bendrą sistemą ir užtikrinti teisingą jų veikimą.

2.1.4 Laisvai samdomi darbuotojai

Naudojimas laisvai samdomų darbuotojų paslaugomis – tai darbų pateikimas atlikti kažkam kitam (už įmonės ribų) – darbo resursu, šiuo atveju, yra konkretus asmuo. Taip įgyvendinamas projektas ar jo dalis.

Atskiras nuo organizacijos asmuo turi savo darbo stilių. Dažnai jis pateikia tik galutinį ar tarpinius produktus nedalyvaudamas organizacijos veikloje. Taigi, įmonėje darbo apskaitos matas yra užduotis pateikiama laisvai samdomam darbuotojui.

Galimas variantas, kai projektavimas atliekamas įmonėje, o atiduodama įgyvendinti konkrečias specifiкуotas dalis. Kitas variantas, kai perduodama visas projektas. Tada laisvai samdomas darbuotojas turi suderinti visu klausimus ir gauti patvirtinimą iš įmonės atstovo ar atstovų.

2.1.5 Bendruomeninių sistemų vartotojai

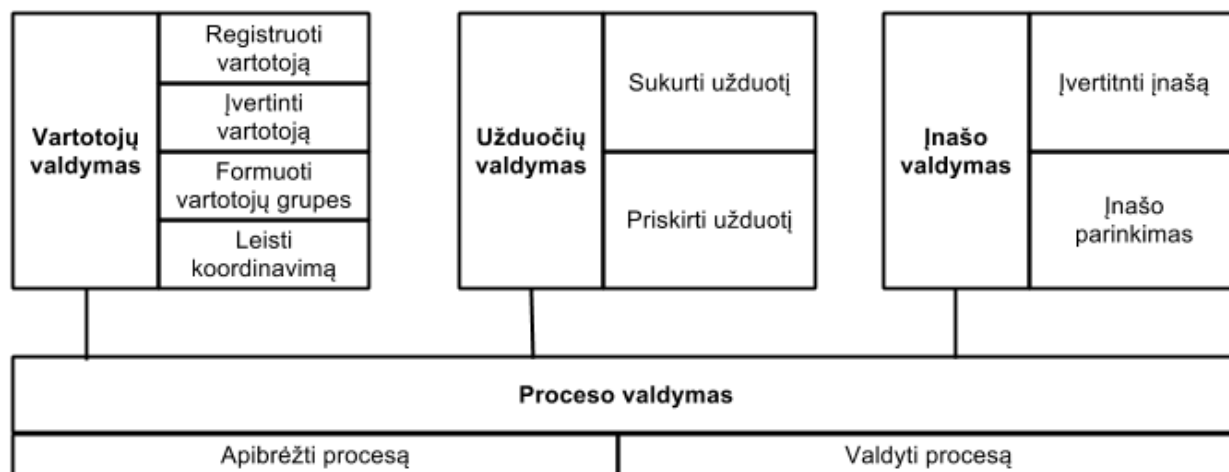
Bendruomeninėmis sistemomis naudojasi tiek didelės tiek ir mažos įmonės, darbams atlikti. O darbų/užduočių įgyvendintojų spektras apima nuo visiškų mėgėjų iki srities profesionalų.

Įgyvendintojams motyvacija atlikti užduotis gali būti tiesioginė finansinė, netiesioginė finansinė, vidinė/pasitenkinimo, socialinio statuso didinimas bendruomenėje (Schenk & Guittard, 2011).

2.2 Bendruomeninių sistemų modelių ar architektūrų analizė

Mokslinėje erdvėje yra siūlomos įvairių modelių ir architektūrų bendruomeninės sistemos ir jų dalys. Analizuojant ir apibendrinant galima išskirti pagrindines sistemų dalis ir funkcijas. Toliau jos yra pateikiamos pagal Hetmank (2013).

2.2.1 Komponentai



2.1 pav. Bendruomeninių paslaugų sistemų komponentai ir funkcijos (pagal Hetmank (2013))

Išskiriami 4 pagrindiniai bendruomeninių paslaugų sistemų komponentai, pavaizduoti 2.1 pav.:

- Vartotojų valdymo – atsakingas už funkcijas susijusias su vartotojų objektais. Šiame komponente būtų naudinga įgyvendinti ir socialinio bendravimo funkcijas, siekiant labiau įtraukti vartotojus.
- Užduočių valdymo – atsakingas už funkcijas susijusias su užduočių objektais.
- Įnašo valdymo – atsakingas už pateikiamų sprendimų valdymą.
- Proceso valdymo – būtinas apibrėžiant sudėtingas užduotis ir padeda išlaikyti kokybę.

Komponentinis šių sistemos dalių įgyvendinimas leidžia plėsti sistemos galimybes ir prijungti naujas, tokias kaip alternatyvias komponentų realizacijas ieškant tinkamiausio sprendimo vykdant veiklą.

2.2.2 Funkcijos

Išskirti komponentai turėtų įgyvendinti jiems keliamą minimalų funkcionalumą pavaizduota 2.1 pav., norint laikyti sistemą bendruomeninių paslaugų sistema. Toliau, apžvelgiant pagal komponentus, pateikiamos funkcijos kurios turėtų būti įgyvendintos juose.

2.2.2.1 Vartotojų valdymas

- Registruoti vartotoją – saugomi tiek užduočių atlikėjo tiek ir užsakovo duomenys. Dėl didesnio tarpusavio pasitikėjimo juos galima apjungti su socialinių tinklų informacija. Registracija nėra būtina visiems dalyviams (pvz., veikiant konkurso principu registruotis turi tik pateikiantys savo sprendimą/kodą, iki tol būti registruotu vartotoju nėra būtina), tačiau turint išsamesnę informaciją galima geriau įvertinti sistemos galimus įgyvendinti darbus, jų sritis.
- Įvertinti/reitinguoti vartotoją – reikalinga nustatant vartotojo gebėjimus, darbo kokybę. Pradinis įvertinimas gali būti atliktas atliekant testus (pvz., sukurti kažkokį programinį sprendimą ar atsakant į klausimus), ar tiesiog pagal vartotojo užpildomą informaciją apie save. Vėliau vertinimas gali būti atliekamas pagal užduotyje dalyvavusiųusių pusių atsiliepimus. Vertinimai turi būti kaupiami ir sudaryti bendrą įvertinimą. Žinoma galimi vertinimo skirstymai pagal sritis (pvz., programavimo kalbą, architektūros sukūrimą, projekto sukūrimą ir kt.).
- Formuoti vartotojų grupes – grupių formavimas gali pasitarnauti kaip papildomas motyvatorius ar surenkant reikiamo dydžio komandą. Grupės gali būti sudaromos tiek atviros, tiek uždaros. Ir žinoma, pati užduotis, jos tipas, sudėtingumas ir mastas, gali reikalauti nuo vieno žmogaus iki didelės grupės norint ją atlikti.
- Leisti koordinavimą – projekto užduotis reikia koordinuoti tarp užsakovo ir bendruomenės. Vykdant pasikeitimą informacija ir pateikiant atgalinį ryšį galima pasiekti, kad programa būtų kuriama taip kaip norėta. Ryšys gali būti tiek iš užsakovo bendruomenei, tiek ir iš bendruomenės užsakovui. Galimi įgyvendinti ir kiti įrankiai didinant sistemos sociališkumą.

2.2.2.2 Užduočių valdymas

- Sukurti užduotį – kuriant užduotį reikia ją pateikti struktūrizuotai, pateikti aprašą ir apribojimus (laiko, atlygio, įrankių naudojamų įgyvendinimui, informacijos neviešinimo, reikalingų atlikėjų apibrėžimo). Taip pat reikia numatyti užduoties suskaidymą ir surinkimą į bendrą sistemą. Teisingai suformuluota užduotis yra vienas iš reikalavimų norint gauti kokybišką programos įgyvendinimą.
- Priskirti užduotį – užduotis turi būti priskiriama reikiamiems atlikėjams reikiamu laiku. Tai galima atlikti susiejant vartotojų informaciją su užduoties specifikacija. Priskyrimo metu tikrinama ar atlikėjas turi reikiamą patirtį ir žinias, bei jo užimtumą.

2.2.2.3 Įnašo valdymas

- Įvertinti įnašą – pateikiamas atgalinis ryšys užduoties atlikėjui, taip gerinant kokybę ir parenkant tinkamiausią sprendimą. Reiktų išskirti įvertinimo tipą – pats atlikėjas, kažkas iš bendruomenės, užsakovas. Įvertinimas gali būti pateiktas laisva forma, ar klausimynu. Įvertinimą pateikti galima užduoties vykdymo laikotarpiu arba po įvykdymo.

- Įnašo parinkimas – tinkamiausio įgyvendinimo parinkimui gali būti panaudojama daugiausiai pateiktas sprendimas ar kontrolinės grupės išrinktas sprendimas. Tai turėtų būti atliekama automatiškai. Kontrolinės grupės atveju, įvertinimas pateikiamas kaip užduotys bendruomenei.

2.2.2.4 *Proceso valdymas*

- Apibrėžti procesą – apibrėžiama vykdymo eiga ir dalinių sprendimų pateikimas. Tai gali atlikti arba užsakovas arba pati bendruomenė.
- Valdyti procesą – tinkamas procesas priklauso nuo užduoties ir jos įvykdymo būsenos. Norint rasti tinkamiausią procesą reikia vykdant iteracijas įvertinti bendruomenės galimybes ir pateikiamą užduotį.

2.2.3 **Užduočių pateikimas kandidatams**

Užduoties paskyrimo funkcijai yra keli galimi variantai. Pasirinktas sprendimas turi pritraukti maksimalų tinkamų kandidatų skaičių, be reikalo neįkyrėti. Toliau aptariami pagrindiniai galimi užduočių pateikimo kandidatams variantai.

2.2.3.1 *Pateikimas visko visiems.*

Šiuo metodu pateikiant užduotis, kandidatai gauna daug perteklinės informacijos, t.y. jie turi papildomai filtruoti užduočių srautą. Tačiau šiuo atveju yra tikimybė, kad užduotį atlikti apsiims kitos srities atstovas. Pastarasis variantas naudingas „nestandartinėms“ užduotims tuo pačiu ir nevisai tikslaus užduoties specifikavimo atveju, pvz.: telekomunikacijų srities specialistai gali pasiūlyti kitokį duomenų perdavimo variantą, gal ir daug tinkamesnį konkrečiai sistemai, nei medicinos srityje dirbantys, t.y. sujungus sričių žinias galima atrasti naujų sprendimų ar taikymų. Standartinėms užduotims, t.y. problemoms, kurių sprendimai aiškūs, o tiesiog reikalinga juos įgyvendinti, pvz., suasmeninto dizaino pritaikymas aiškiai apibrėžtam komponentui. Tokiu atveju tai geriausiai galės atlikti šios srities specialistas.

2.2.3.2 *Pateikimas pagal pasirinktas dominančias sritis.*

Bendru atveju, galimi užduočių atlikėjai pasirenka juos dominančias sritis. Sričių apibrėžimas gali apimti konkrečią programavimo kalbą (C#, Java), metodiką (objektinis, funkcinis programavimas), veiklos sritį (internetiniai sprendimai, matematiniai uždaviniai) ir t.t. Jei platforma yra skirta kažkuriai sričiai, vis tiek yra dalis, kuri nėra apibrėžta. Tokiu atveju skaidymas vykdomas pagal tokias neapibrėžtas sritis. Taip kandidatams tiek paprasčiau atsirinkti dominančias užduotis, tiek galima įgyvendinti informavimo pranešimų apie naujas užduotis sistemą. Tačiau nestandartinių sprendimų ar nevisai teisingo sričių specifikavimo atveju, prarandami kandidatai. Taip pat kandidatai iš anksto turi pasirinkti dominančias sritis.

2.2.3.3 *Pateikimas pagal atitikimą reikalavimams.*

Kandidatai pateikia kuo išsamesnę informaciją apie savo galimybes, juos dominančius užduočių parametrus. Tuo pačiu užsakovai, pateikdami užduotis išsamiai aprašo reikalavimus. Tiek užsakovų tiek ir kandidatų informacija turi būti pateikiama sistemos protokolu (Ebner, Leimeister, Bretschneider & Krcmar, 2008), (Psaier, Skopik, Schall & Dustdar, 2011). Pateikiami reikalavimai apima tokius aspektus kaip: veiklos sritis, numatomas įvykdymo laikas, užsakymo kainos rėžiai (užsakovai ir kandidatai pateikia juos tenkinančius), nuobaudos už vėlavimą, reikalingos patirtys, atliktų užduočių reitingas ir kt. (Psaier, Skopik, Schall & Dustdar, 2011). Šiuo atveju paprasčiau atsirinkti dominančias užduotis. Tačiau reikia teisingai įvertinti parametrus tiek užduoties pateikėjams, tiek ir kandidatams. Realizacija sudėtingesnė, nei kitų minėtų būdų. Reikalingas informacijos aprašymo protokolas sistemoje.

2.2.4 Užduoties atlikėjo parinkimas

Kita užduoties paskyrimo dalis yra atlikėjo parinkimas norint gauti teisingą atsakymą. Toliau aptariami keli galimi kandidatų parinkimo variantai.

2.2.4.1 *Užduoties atlikėjas neparenkamas.*

Veikiama varžybų principu. Galimos kelios variacijos:

- Pateikus rezultatus, iš jų išrenkamas nugalėtojas (gali būti keli) pagal teisingiausią, labiausiai tinkamą ar kitus kriterijus atitinkantį sprendimą.
- Pateikiami tarpiniai rezultatai prieinami visiems. Dalyviai gali pasinaudoti idėjomis gerinant savo sprendimą.
- Užsakovas komentuoja gaunamus rezultatus. Pateikia pastabas sprendimams. Dalyviai atitinkamai tobulina sprendimus.

Variantas tinkamas kai ieškoma naujų sprendimų, turint tik idėją, neturint realizacijos specifikacijos. Tikimasi gauti keletą sprendimų, taip surandant tinkamiausią.

Reikalinga didelė bendruomenė, nes reikalinga, kad dalyvautų keli galintys atlikti užduotį tinkamai. Esant specifinės srities užduočiai, neturint didelės bendruomenės gali būti sunku surinkti dalyvių.

2.2.4.2 *Kandidatas pasirenka užduotį, be aktyvaus užsakovo.*

Kandidatas iš jam prieinamų užduočių pasirenka užduotį, ir ją atlieka. Parinkimo procese iš esmės nedalyvauja užduoties pateikėjas. Jis tik pateikia informaciją kandidatams. Savaime variantas labai tinkamas mažoms, nekritinėms užduotims, taip pat užduotims turinčioms aiškią įgyvendinimo specifikaciją. Norint jį naudoti svarbioms užduotims atlikti reikalinga sistema leidžianti sumažinti riziką dėl kandidato netinkamumo (Psaier, Skopik, Schall & Dustdar, 2011).

2.2.4.3 *Kandidatas pasirenka užduotį, su aktyviu užsakovu.*

Kandidatai pateikia savo klausimus, pastabas. Pagal tai tikslinami (siaurinami ar platinami) reikalavimai (Ebner, Leimeister, Bretschneider & Krcmar, 2008), (Psaier, Skopik, Schall & Dustdar, 2011).

Variantas tinkamas užduotis pateikiant pagal atitikimą reikalavimams arba kai nėra iš anksto tiksliai žinomi visi reikalavimai (pvz., užsakovas negali įvertinti trukmės, nes pilnai nežino darbo specifikos). Galima išskaidyti etapais, t.y. papildžius informaciją užduotis laikinai užšaldoma (neleidžiama pasirinkti).

2.2.4.4 *Kandidato modelio parinkimui svarbi informacija*

Renkantis ar sudarant kandidato pasirinkimo modelį sistemoje reikia atsižvelgti į tokius aspektus kaip:

- Užduoties kritiškumas (kaip svarbu gauti sprendimą laiku, kaip svarbu gauti teisingą sprendimą, ir t.t.).
- Užduoties standartiškumas (ar ieškoma naujų sprendimų, ar tiesiog įgyvendinamas standartinis elementas).
- Reikalingas srities išmanymas (užduoties atlikėjas turi teisingai traktuoti naudojamus specifinius srities terminus, kitur galimai naudojamus kita prasme).
- Kokius realius įsipareigojimus prisiima kandidatas pasirinkdamas užduotį (įsipareigoja tinkamai atlikti užduotį iki galo ar gali bet kuriuo metu atsisakyti be pasekmių).

2.3 Bendruomeninių paslaugų apžvalga

Bendruomeninės paslaugos yra panaudojamos įvairiose srityse, taip pat ir IT. Jomis pasinaudojant galima įgyvendinti projektus. Pasinaudojant bendruomene galima tobulinti algoritmus (Netflix Prize), projektuoti bei įgyvendinti projektą ar jo dalį (rentacoder.com, odesk.com). Taip pat reikalui esant galima laikinai pasinaudoti specifiniais specialistų įgūdžiais iš išorės ar pritraukti darbus, kaip siūloma Ašeriškio & Tamošaičio (2012).

Toliau apžvelgiamos bendruomeninių paslaugų sistemos, kuriomis pasinaudojant galima tai įgyvendinti.

2.3.1 Microworkers

Koncepcija. Microworkers bet kuris vartotojas gali būti tiek užduočių pateikėjas, tiek užduočių vykdytojas. Microworkers susideda iš mažų, greitai atliekamų užduočių už kuriuos gaunamas mažas atlygis. Atlygis gali svyruoti nuo 0,10 iki 2 dolerių.

Procesas. Bet kuris vartotojas gali sudaryti užduotį, kuri publikuojama viešai. Publikuojant užduotį, nurodoma, kiek sėkmingų įvykdymų reikia ir atlygis už atlikimą. Bet kuriuo metu vartotojas gali nutraukti užduoties vykdymą. Vartotojai, surinkę pakankamai didelę sumą, ją gali išsigryninti.

Charakteristikos. Mažos užduotys reikalaujančios mažai pastangų, nereikalaujančios išsilavinimo ir mažas atlygis. Visiems atvira registracija.

Sėkmės priežastis. Paprastumas, greitas atlikimas ir populiarumas. Tačiau šis labiau naudingas jauniems arba užsienio šalių darbuotojams, nes valandinis uždarbis mažas siekiantis vidutiniškai 2,5 dolerio (Hoßfeld, Hirth, & Tran-Gia, 2011). Taip pat statistinių duomenų rodomas didelis prognozuojamas augimas (Hoßfeld, Hirth, & Tran-Gia, 2011).

Motyvacija. Greitas užduoties pateikimas ir nebrangus darbas. Tačiau žmonės linkę neišlaikyti ir nebūti suinteresuoti ilgą laiką.

Apribojimai. Nesurenkama išsilavinusi bendruomenė ir ji nėra aktyviai administruojama.

2.3.2 Amazon MTurk

Koncepcija. Amazon MTurk (Hoßfeld, Hirth, & Tran-Gia, 2011) tai darbų, reikalaujančių žmogaus intelekto, sistema. HIT – tai klausimas, į kurį turi būti galima pateikti atsakymą. Svetainė padalinta į dvi dalis – vartotojų atliekančių užduotis portalas ir darbus teikiančių vartotojų portalas.

Procesas. Skiriamos dvi rolės – užduočių tiekėjai ir užduočių vykdytojai. Užduočių tiekėjai pateikia užduotis, šios yra vykdomos vartotojų ir pateikiami rezultatai. Pastarieji patvirtinami arba atmetami.

Charakteristikos. Paprasta sistema, įvairios vertės darbai nuo 0,01 iki 15 dolerių. Patenka užduotys reikalaujančios žmogaus intelekto, tokios kaip paveikslėlių palyginimas, paveikslėlių žymėjimas, balso įrašymas, apžvalgų rašymas ir kitos. Egzistuoja kvalifikacijų sistema.

Sėkmės priežastis. Populiarumas, didelė bendruomenė iš 190 valstybių. Užduotys dažniausiai nereikalauja specifinių žinių, todėl prieinamos didžiąjai bendruomenės daliai.

Motyvacija. Pinigai – pagrindinis darbuotojų motyvatorius. Užsakovus domina nebrangus darbas ir dideli išteklių, kuriuos nesunkiai galima pasitelkti.

Apribojimai. Platus spektras sutraukia daug skirtingų žmonių, tačiau nėra nišos atlikti specifinių žinių reikalaujančius darbus.

2.3.3 oDesk

Koncepcija. oDesk simuliuoja realius darbo santykius. oDesk valdo grupę kvalifikuotų darbininkų ir renką įmonių grupę samdančią darbuotojus.

Procesas. Yra dvi vartotojų grupės – darbdaviai ir samdiniai. Darbdaviai pateikia užduotį. Tada samdiniai pateikia kandidatūras. Darbdavys atsirenka darbuotoją, kurį nori samdyti atlikti valandiniam arba fiksuotam darbui. Samdinys užsiregistravęs sudaro savo pilnavertį profilį, į kurį įeina įgūdžių testų rezultatai, atlikti darbai ir kita informacija. Laikui bėgant profilis surenka informaciją apie samdinį. Samdinys gauna darbo pasiūlymus ir juos vykdo.

Charakteristikos. Samdomo darbuotojo principai, valandinis atlygis. Geri darbuotojai yra gerai apmokami.

Sėkmės priežastis. Adekvatūs pinigai už atliktą darbą. Darbo atlikimo ir atlyginimo už darbą garantijos. Didelė kvalifikuotų darbuotojų bendruomenė, dauguma kurių yra pigiau apmokami negu atitinkami specialistai vietos darbo rinkoje.

Motyvacija. Darbas namuose, adekvatus atlyginimas ir įgaunama reali darbo patirtis. Pigiau atmokamas darbas, lengviau administruojamas darbuotojų srautas, didesnis kvalifikuotų darbuotojų pasirinkimas ir darbo laiko skaičiavimas su ekrano vaizdais – tokios priežastis motyvuoja darbdavius rinktis oDesk.

2.3.4 TopCoder

Koncepcija. TopCoder projekto dalys pateikiamos kaip varžybų užduotys. Materialų atlygį iš gauna varžybų prizininkai (Fried, 2010).

Procesas. Pateikiamas užsakymas (viso projekto arba dalies). Projektai įvertinami ir suskaidomi dalimis dirbant su TopCoder komanda. Suskaidytos dalys pateikiamos etapais. Jos formuluojamos kaip varžybų užduotys. Visi norintys dalyvauti (registruoti vartotojai ar komandos) gali pateikti savo sprendimus. Iš pateiktų sprendimų išrenkami geriausi (tai atlieka paslaugos tiekėjas). Apdovanojami (apmokami) prizininkai. Reikalui esant skelbiami tarpiniai konkursai, pvz., klaidų paieška ir taisymas.

Charakteristikos. Konkursinis sprendimas. Atlygį gauna tik geriausiais atrinkti sprendimai (1,2 vietos).

Sėkmės priežastis. Išsprendžiamos įvairios ir nestandartinės problemos (pvz., NetflixPrize). Didelė IT bendruomenė. Bendruomenė, be dalyvavimo konkursuose, kuria naudingą turinį – mokomuosius, pažintinius straipsnius. Pigesni ar/ir geresni sprendimai užsakovams, nei įprastiniu būdu įgyvendinant.

Motyvacija. Noras varžybose įrodyti savo gebėjimus ir stiprinti statusą bendruomenėje aplenkiant kitus. Naudinga patirtis sprendžiant užduotis. Įrodžius savo gebėjimus, atsiranda realios galimybės įsidarbinti didelėse kompanijose (Fried, 2010).

Apribojimai. Maža dalis gauna atlygį. Nepritraukiami specialistai, kuriems nesvarbus statusas sistemoje, tačiau už atlygį galintys pasiūlyti sprendimus. Nenaudinga įgyvendinant iš anksto aiškiai specifiкуotus sprendimus (nėra iš ko sudaryti tinkamo konkurso), nebent klaidų paieškos organizavimas.

2.3.5 Sistemų palyginimas

2.1 lentelė. Bendruomeninių paslaugų sistemų palyginimas

Sistema	Galimas užduočių sudėtingumas	Finansinis atlygis	Užduočių pateikimo sudėtingumas	Ar tinka programinės įrangos kūrimui?
Microworkers	Mažas (paspaudimai/balsavimai, žymėjimai, trumpos apžvalgos)	Mokama teisingai atlikus užduotį (validuoja užsakovas)	Paprastas – suvedama į standartizuotą formą, kuri toliau vykdoma automatizuotai	Dalinai (mažo sudėtingumo užduotys, nėra buriami programavimo specialistai)
MTurk	Mažas (paspaudimai/balsavimai, žymėjimai, trumpos apžvalgos)	Mokama teisingai atlikus užduotį (validuoja užsakovas)	Paprastas – suvedama į standartizuotą formą, kuri toliau vykdoma automatizuotai	Dalinai (mažo sudėtingumo užduotys, nėra buriami programavimo specialistai)
oDesk	Nuo mažo (paspaudimai, trumpos apžvalgos) Iki sistemų projektavimo ir sukūrimo	Mokama už atliktą darbą arba laiką (užsakovas nurodo pradžioje)	Vidutinis – reikia išsirinkti kandidatus	Taip
TopCoder	Algoritmų kūrimas, tobulinimas, sistemų projektavimas ir sukūrimas	Mokama varžybų principu – prizininkai gauna atlygius, o dauguma nieko	Sudėtingas – reikia dirbti su tiekėjo komanda	Taip

Sistemų palyginimo rezultatai apibendrinti 2.1 lentelėje. Sistemos su mažu galimu užduočių sudėtingumu kartu suteikia ir mažesnę finansinę atlygį už pačias užduotis. Tačiau ir turint didesnio sudėtingumo užduotis, finansinis atlygis gali būti nedidelis. Taip yra dėl kitų darbų atlikimo motyvatorių sistemoje. Ties mažo sudėtingumo užduotimis besikoncentruojančios sistemos, pačios netinka visos programinės įrangos kūrimui, nes nebeatitinka viso proceso sudėtingumo lygio. Tačiau dalį naudojamų metodų galima taikyti apjungus su kitais (pvz., realių testavimo duomenų sugeneravimas, rezultato validavimas ir pan.).

2.3.6 Problemos

Gautą rezultatą iš bendruomenės reikia įvertinti. Taip pat, kuriant sistemą, bendruomenė pateikia jos dalis, kurias reikia sujungti į visumą ir užtikrinti veikimą.

2.3.6.1 Bendruomenės valdymas ir formavimas

Viena labai svarbių problemų yra bendruomenės valdymas ir formavimas. Pirmiausia, tai labai svarbu, nes įmonės nori garantijų, kad žmonės, atliekantys jų užduotis, būtų pakankamai kvalifikuoti.

Bendruomenės narių kvalifikaciją galima valdyti keliais būdais:

- Uždaras kandidatavimo procesas – pirmas žingsnis link uždaros rinktinės bendruomenės, čia surenkame bazinę informaciją apie būsimą darbuotoją ir pagal ją patvirtiname arba atmetame kandidatą, panašiai kaip iStockPhoto.

- Vidinis darbuotojų kvalifikacijos nustatymas testais – čia pateikiame specializuotus testus, kurių rezultatais papildomas profilis, panašiai kaip oDesk.
- Kvalifikacijos valdymas ir naujinimas pagal atliktus darbus.

Bendruomenės narių formavimui dažniausiai naudojamos atsiliepimų, rekomendacijų ir reitingavimo sistemos. Atsiliepimų sistema leidžia pagirti arba papeikti buvusį darbuotoją. Rekomendacijų sistema leidžia bet kam pateikti savo nuomonę apie darbuotoją ir jo asmenines savybes. Reitingavimo sistema leidžia darbuotojus lyginti tarpusavyje pagal išvestinį parametą. Į šį parametą patektų tokie pirminiai parametrai kaip sėkmingumas, atlygis, patikimumas ir kiti.

2.3.6.2 Teisinės bazės nebuvimas

Viena iš didžiausių problemų tiesiogiai nesusijusių su programinės įrangos kūrimu yra tai, kad nėra suformuotos teisinės bazės, kuria galima būtų remtis formuojant bendruomeninių paslaugų valdymo platformą. Net ir JAV jokių teisinių pamatų dar nepradėta kloti, nors ten teismai jau turi sukaupę šiokią tokią patirtį paremtą egzistuojančia teisine baze. Yra trys aktualūs teisiniai aspektai (Wolfson, & Lease, 2011).

Pirmiausiai, darbo įstatymai ir darbo santykiai. Tokios platformos nėra socialiai atsakingos, nes tai nėra darbdavio ir samdomo darbuotojo santykiai. Žmogus negauna jokių socialinių garantijų, kurios būtų jam priklausančios normaliu atveju, taip pat negarantuoja jokios karjeros galimybės, aišku išvengiama mokesčių ir jų našta numetama samdiniui ir darbo santykiai neturi jokių ribojimų darbo kodekso atžvilgiu, nes šie santykiai nėra klasifikuojami kaip darbdavio-darbuotojo santykiai (Wolfson, & Lease, 2011).

Antra, autorystė ir patentai gautiems sprendimams vis tiek teisiškai priklauso kūrėjui. Šis teisinis aspektas mus liečia labiau iš autorystės teisių pusės, InnoCentive atveju patentai būtų svarbūs. Esmė yra tame, kad tai nėra darbdavio-darbuotojo santykiai, ir nėra aiški formulė, kaip tie darbo santykiai turi būti reglamentuoti. Nustatant naudojimosi sąlygas būtina įvertinti, kad būtų suderinta autorystės teisių problema (Wolfson, & Lease, 2011).

Trečia, duomenų saugumas, t.y. tam tikriems darbams tenka perduoti duomenis, kurie remiantis daugumos išsivysčiusių valstybių teisine baze negali būti perduoti trečioms šalims apie tai neinformavus vartotojų ir jiems nesutikus. Ši problema dažnai sutinkama kai norima perduoti duomenis trečiajai šaliai (bendruomenės nariui) analizuoti arba naudoti darbo procese (Wolfson, & Lease, 2011).

2.4 Bendruomeninių sistemų ir paslaugų vertinimo metodų apžvalga

Atliktas užduotis reikia įvertinti, prieš apmokant už darbą. Be to užduoties pateikėjas gali neturėti ar nenorėti skirti resursų užduoties atlikimo kokybės patvirtinimui. Taip pat įvertinant užduotį gali kilti vertinimo nesutapimas tarp užduoties pateikėjo ir vykdytojo.

Todėl reikalingas kokybės tikrinimo mechanizmas iš bendruomeninių paslaugų tiekėjo pusės. Rezultato kokybės įvertinimui gali būti panaudojami tokie metodai:

- Įmonės atstovas įvertina kokybę.
- Bendruomenei pateikiama papildoma užduotis kokybės įvertinimui (Hirth, Hoßfeld & Tran-Gia, 2011).
- Įvertinimą atlieka paslaugos tiekėjas (kokybės vertinimo komanda).

Pateikimo bendruomenės įvertinimui tinkamos tiksliai specifikuotos užduotys. Galima pateikti keliems vertintojams ir pagal daugumos vertinimą nuspręsti. Tačiau vertinimas negali būti pateikiamas bet kokiems vartotojams. Reikalingi gebėjimai vertinant atitikimą pateiktiems reikalavimams. Tokios užduotys gali būti pateiktos tik reikiamą patirtį turintiems. Taip pat dažnai naudingiau, jei rezultato įvertinimą atliktų

nesusiję su užduoties pateikėju ir vykdytoju asmenys. Taip išvengiama šališkumo, kaip nagrinėja Hirth, Hoßfeld & Tran-Gia (2011) ir Millard (2011).

Norint užtikrinti rezultato kokybę, reikalinga ir tinkama rezultatų reitingavimo sistema – jei vis pateikiami nekokybiški rezultatai, tokiam vartotojui neturi būti leidžiama atlikinėti užduočių. Tačiau vartotojas galimai vieno tipo užduotis atlieka sėkmingai, tik problema su kito tipo užduotimis. Čia vartotojas neteisingai įvertina savo gebėjimus. Todėl būtų naudinga, kad pati sistema tiesiog nebeleistų tokiam vartotojui rinktis atitinkamo tipo užduočių.

Taip pat užduoties pateikėjas gali bandyti neigiamai vertinti užduotis, kad nereikėtų už jas atsiskaityti. Tokius vartotojus taip pat būtina nubausti: nebeteikiant paslaugos arba informuojant prie užduoties apie galimas problemas.

Reitingavimo sistemos viena iš dalių galėtų būti sistemos vartotojų vertinimai (nebūtinai susijusių užduoties atlikimu) (Ebner, Leimeister, Bretschneider & Krcmar, 2008). Čia šiuo atveju vertinama informacijos pateikimas, tikslumas. Vartotojams pateikiama informacija akcentuotų, jog gali reikėti labiau įsigilinti į problemą, jei užsakovas dažnai nepilnai specifikuoja užduotis.

2.4.1 **Kokybiniai vertinimo metodai (apklausos ir pan.)**

Vertinant sukurto kodo bei paties darbo kokybę, tai galima atlikti vykdant proceso dalyvių apklausas. Atsižvelgiant tiek į užduoties atlikėją tiek ir užsakovą.

Vienas iš būdų yra atviro formato vertinimo formos pateikimas. Toks variantas yra sudėtingai sistemizuojamas ir apjungiamas bendro įvertinimo rodiklius. Tačiau vartotojų atsiliepimai laisva forma yra svarbūs, nes kitiems sistemos dalyviams pateikia informaciją neapibrėžtą standartiniais klausimais.

Kitas būdas yra reitingavimas pagal pateiktas sritis nurodytoje skalėje. Tačiau balo skyrimas yra subjektyvus ir, kaip apžvelgia Lange & Lange (2012), net apmokyti reitinguotojai nepateikia vienodų įvertinimų. Todėl siekiant turėti apibendrintą įvertinimą, pateiktų įvertinimų nepakanka tiesiog sumuoti. Juos reikia apdoroti papildomai norint turėti kuo tiksliau išreiškiantį tikrąją situaciją įvertį.

Šiam tikslui galima naudoti daugialypės Rasch skalės metodą. Kurio metu, įvertinimo balai yra statistiškai modeliuojami kaip trijų elementų (pateiktas įvertinimas, vertintojo atlaidumas ar griežtumas ir įvertinimo priskyrimo riba/žingsnis) sąveikos rezultatas (Lange & Lange, 2012). Metodo taikymo rezultate, pašalinami nukrypimai ir gaunamas bendras įvertinimas. Taigi jis gali būti taikomas konkursinio rezultato patvirtinimo atveju ar įvertinant aktyvių bendruomenės dalyvių darbo kokybę, remiantis vertintojų gebėjimu teisingai tai atlikti.

2.4.2 **Kiekybiniai vertinimo metodai (programų metrikos)**

Vertinant rezultatą galima vertinti ne tik kaip užsakovai ir dalyviai priėmė galutinį rezultatą – kokybę, bet ir jo kiekybinius parametrus, bei pačios sistemos evoliuciją kūrimo eigoje. Taip būtų galima įvertinti kūrimo stadiją ir progresą dar prieš turint galutinį produktą ar sprendimą.

Vertinant kiekybinius parametrus galima išskirti dydžio ir sudėtingumo metrikas. Sudėtingumo metrikoms tirti, reikalingas paties kodo abstrakcijos sudarymas įvertinant jo pokytį (Vasa, 2010). Vasa išskiria dvi matuojamas sudėtingumo metrikas:

- Tūrinis sudėtingumas – vertinamas skaičiuojant skirtingų abstrakcijų kiekį.
- Struktūrinis sudėtingumas – vertinamas skaičiuojant ryšius tarp abstrakcijų.

Šios metrikos apima metodų daugkartinį panaudojimą, ryšius tarp klasių, jų susietumą.

Vertinant dydžio metrikas galima matuoti kodo eilučių skaičių, laiką tarp versijų išleidimo ar pačias realizacijas, bei pokyčius tarp jų. Vertinant kalendorinį laiką yra viena pagrindinė limitacija – nėra plačiai priimtos įvertinimo sąsajos tarp kalendorinio laiko ir pastangų (Vasa, 2010).

2.5 PVS savybių apžvalga

Rinkoje yra siūloma daug įvairaus funkcionalumo PVS. Toliau apžvelgiamos kelios iš jų ir išskiriamas pagrindinis reikalingas funkcionalumas.

2.5.1 Clarizen

Teikiama kaip paslauga. Suteikia daug įrankių, skirtų planavimui, resursų, užduočių, problemų valdymui, ataskaitų generavimui ir kt. Yra galimybė nustatytas įrankio dalis modifikuoti pagal reikalavimus – kategorijų kūrimas, laukų validacija, procesų eigų sudarymas. Kiekvienai sričiai galima rinktis iš keleto įrankių, pavyzdžiui, projekto ir laiko planavimui – Ganto grafikai, kritinio kelio metodas ir daug kitų. Įrankiai integruoti vieni su kitais, todėl informaciją peržiūrėti galima tinkamiausiu būdu. Tačiau didelė gausa priemonių sukelia ir problemų – reikia mokėti jais naudotis, pasirinkti tinkamą užduočiai.

2.5.2 Redmine

Teikiamas sprendimas naudojimui, atviro kodo. Mažiau įrankių, tačiau yra bendruomenės kuriami papildiniai. Standartiškai pateikiami pagrindiniai įrankiai – užduočių, problemų valdymas, bendradarbiavimo įrankiai (wiki, forumas), Ganto grafikas laiko planavimui. Reikalui esant funkcionalumą galima keisti pagal savo poreikius. Pagrindiniai įrankiai paprastai naudojami. Bendruomenės kuriami papildiniai įvairūs, apima ir programinio kodo tvarkymo, integravimo elementus (kaip pvz., problemų pažymėjimas išspręstomis pagal kodo įkėlimo (angl. „commit“ ar panašus terminas priklausomai nuo naudojamos sistemos) komentarus). Prireikus pagalbos, reikia kreiptis į bendruomenę arba tvarkytis savo jėgomis.

2.5.3 Projectpier

Teikiamas sprendimas naudojimui, atviro kodo. Pateikiami tik keli baziniai įrankiai projekto planavimui. Galima priskirti ir sukurti projektus, užduotis, projekto etapus, pridėti dokumentus. Įrankis paprastas naudotis, tačiau norint papildomo funkcionalumo reiktų kurti jį pačiam. Tinkamas projektams, kuriems reikia tik paprasto projekto užduočių valdymo

2.5.4 Sistemų palyginimas

2.2 lentelė. Projektų valdymo sistemų palyginimas

Sistema	Užduočių valdymas	Laiko planavimas	Dokumentų valdymas	Naudojimosi sudėtingumas
Clarizen	Registravimas, perskyrimas, susiejimas su kitomis, istorija, atskiros sritys veiklų planavimui ir problemų registravimui	Užduočių laiko informacija pateikiama grafikais, matomas atlikimo planas, Gant grafikas, galima valdyti užduoties datas bendrame vaizde, kritinis kelias	Prisegimas prie užduoties, Wiki, dokumentų versijų valdymas, keitimo teisės	Sudėtinga pasirinkti įrankius, ir juos tinkamai panaudoti
Redmine	Registravimas, perskyrimas, susiejimas su kitomis, istorija	Užduočių laiko informacija pateikiama grafikais,	Galima prisegti dokumentus prie užduoties, Wiki	Paprasta, jei tenkina bazinis funkcionalumas.

		matomas atlikimo planas		
Projectpier	Bazinis registravimas	Tik konkrečiai užduočiai nurodomas laikas	Nėra	Paprasta, jei tereikia teikiamo funkcionalumo

Remiantis projektų valdymo sistemų palyginimu pateiktu 2.2 lentelėje, didėjant teikiamoms galimybėms sistemose didėja ir proceso naudojimo sudėtingumas. Žinoma, padėti bandoma taisyti, kuriant patogesnes vartojimo sąsajas, bei didinant elementų tarpusavio integracijos lygį.

2.6 Duomenų sinchronizavimo metodų apžvalga

Informacijos sinchronizavimas su sistemomis, tarp vienos įmonės projekto valdymo sistemos ir paslaugų valdymo sistemos, bei tarp kitų įmonių projekto valdymo sistemų yra svarbi problema. Jei informacija skirsis sistemose, gali kilti konfliktai tarp užduočių įgyvendinimo (McClowry, 2007).

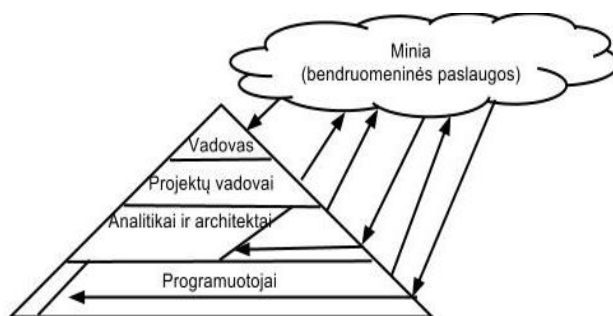
Informacijos sinchronizavimui galima panaudoti įvairius modelius. Tai gali būti nuo realaus laiko pranešimų tiesiai sistemoms iki pasikeitimų kaupimo atskirame modulyje (sinchronizavimo sistemoje) ir pateikimo tik įvykus užklausoms.

Didžiausia sinchronizavimo problema sistemoje yra tai, jog daug vartotojų vienu metu gali bandyti keisti informaciją. Kritinė sritis yra užduoties statuso keitimas. Čia yra ir informacijos konkurencijos problema. Reikalinga, kad keli vartotojai negalėtų vykdyti tos pačios užduoties (jei ji nėra tam pritaikyta), ar užsakovas savavališkai panaikinti jau vykdomą užduotį. Užduoties apsiėmimo konkurencijos problemą galima spręsti sudarant neveiklumo langą, t.y. pasirinkus vykdyti užduotį ji neiškarto atsiranda pas vartotoją. Išlaukiamas nustatytas laikas, per kurį užduotį gali apsiiminėti ir kiti. Vėliau išrenkamas tinkamiausias kandidatas. Arba galima užduotį priskirti tiesiog tam, kurio pranešimas apie užduoties apsiėmimą sistemą pasiekė pirmasis, kitiems bandantiems pranešant, jog jie nespėjo.

Sinchronizavimo modulis taip pat gali atlikti informacijos transformaciją į reikiama formatą pateikimui, vienai ar kitai sistemai (McClowry, 2007). Taip pat, prieš pateikiant informaciją kitiems, vykdoma jos validacija (pvz., ar vis dar buvo galima pakeisti užduoties statusą).

2.7 Įmonės organizacinis modelis

Projektų kūrimui panaudojant bendruomenines paslaugas modelis organizacijai pasiūlytas Ašeriškio ir Tamošaitis (2012):



2.2 pav. Įmonės resursų struktūra (pagal Ašeriškį ir Tamošaitį (2012))

Resursų struktūroje, pateiktoje 2.2 pav., rodyklės iš minios reiškia resursų paėmimą, o rodyklės į minią – resursų perdavimą kitiems. Resursai gali būti užduotys ar darbuotojai. Rodyklės piramidės viduje rodo galimybę darbuotojų kiekį keisti priklausomai nuo darbo, t.y. dalimi darbuotojų galima laikyti minią. Minios nariu gali būti tiek pavienis asmuo, tiek ir kita įmonė.

2.8 Modelio tinkamo programinės įrangos kūrimui panaudojant bendruomenines paslaugas variantas

Projektų kūrimui panaudojant bendruomenines paslaugas technologinis sprendimas pasiūlytas Ašeriškio ir Tamošaičio (2012):

Naujos kartos programinės įrangos kūrimo įmonė turėtų turėti programinę įrangą, kuri įmonei leistų: 1) valdyti savo veiklos procesus ir projektus; 2) užtikrinti sąvokų vienareikšmį suvokimą organizacijoje; 3) gebėti kardinaliai padidinti savo pajėgumus atsiradus poreikiui; 4) įtraukti darbų srautą esant poreikiui; 5) projektų vadovas turi galėti operatyviai valdyti organizaciją; darbuotojai sistema turi naudotis ne prievarta, o savo noru; 6) sistema turi padėti motyvuoti darbuotojus; 7) padėti valdyti visą projektų informaciją; 8) leisti kaupiti žinias; 9) padidinti skaidrumą organizacijoje.

Pastaroji sudaryta iš dviejų esminių dalių: PVS – galima platus suvokimas remiantis skirtingais aspektais, pvz., iš sistemų mokslo perspektyvos [4]; bendruomeninių darbo paslaugų valdymo sistemos [11], [1], [3].

Straipsnyje taip pat siūloma nekurti pačios bendruomeninės sistemos, o ją integruoti į kuriamą projektų valdymo sistemą, sudarant galimybę ateityje prijungti ir kitas bendruomenines sistemas panaudojimui, įgyvendinant projektus.

2.9 Išvados

Projekto vykdymo darbus galima dalinti įmonės viduje arba dalį išskelti išoriniams atlikėjams. Darbo atlikimui naudojantis bendruomeninėmis paslaugomis, jų tiekėjas turi įgyvendinti pagrindines 4 valdymo sritis: vartotojų; užduočių; įnašo; proceso. Užduotys sistemoje turi būti pateikiamos užduoties reikalavimus atitinkantiems kandidatams. Nėra teikiamo naudojimui PVS su bendruomeninėmis paslaugomis sprendimo, o tik atskiros PVS ir bendruomeninių paslaugų sistemos.

3 PVS SU INTEGRUOTOMIS BENDRUOMENINĖMIS PASLAUGOMIS

Projekto metu realizuota projektų valdymo sistema su integruotomis bendruomeninėmis paslaugomis. Sukurta sistema leidžia vykdyti projekto valdymą užduotimis, užduotis pateikti, bei priimti iš bendruomeninių paslaugų valdymo sistemos.

Sistema projektuota taip, kad būtų galimas pakartotinis elementų panaudojimas ir plėtimas.

3.1 Architektūros tikslai ir apribojimai

Sistemai keliami tokie tikslai ir reikalavimai:

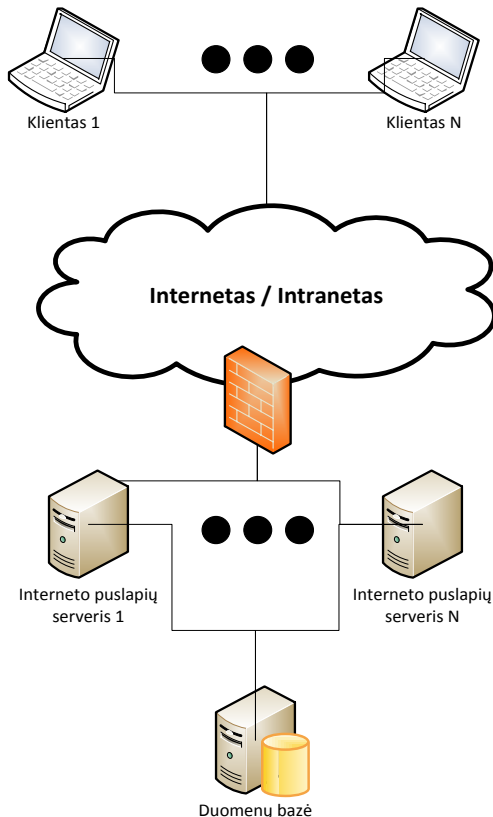
- Sistema remiasi trijų lygmenų architektūra, kurią sudaro aplikacijos lygmuo, sprendimų lygmuo ir duomenų lygmuo.
- Sistema remiasi sprendimų izoliavimo metodu, t.y. atskiri sistemos elementai kuriami atskirai ir yra tarpusavyje kiek galima mažiau priklausomi. Tokia sistemos kompozicija pasiekama tokiu būdu:
 - atskiros problemos izoliuojamos ir joms ruošiami sprendimai;
 - sprendimai gali būti vienos arba kelių pakopų, t.y. atskiriama bendrinė dalis nuo praktiniam uždaviniui spręsti reikalingos programos;
 - atskiri sprendimai sujungiami tik aplikacijos lygyje, nebent sprendimui gauti reikalingas kitas bendrinio pobūdžio sprendimas.
- Naudojami trečios šalies sprendimai, turi būti integruojami per papildomą abstrakcijos lygmenį.
- Sistema turi laikytis pagrindinių interneto aplikacijų principų.

- Sistema turi išnaudoti komponentinio programavimo privalumus.
- Sistema kuriama, kaip uždaro kodo sistema, todėl negali naudoti GPL šeimos licencijuojamų komponentų.

Sistemos architektūrą sudaro sprendimai, baziniai elementai ir specifinis, kuriamas produktas.

Sprendimų išskaidymas atskirais komponentais leidžia juos keisti, pildyti ir pernaudoti atskirai nuo visos sistemos.

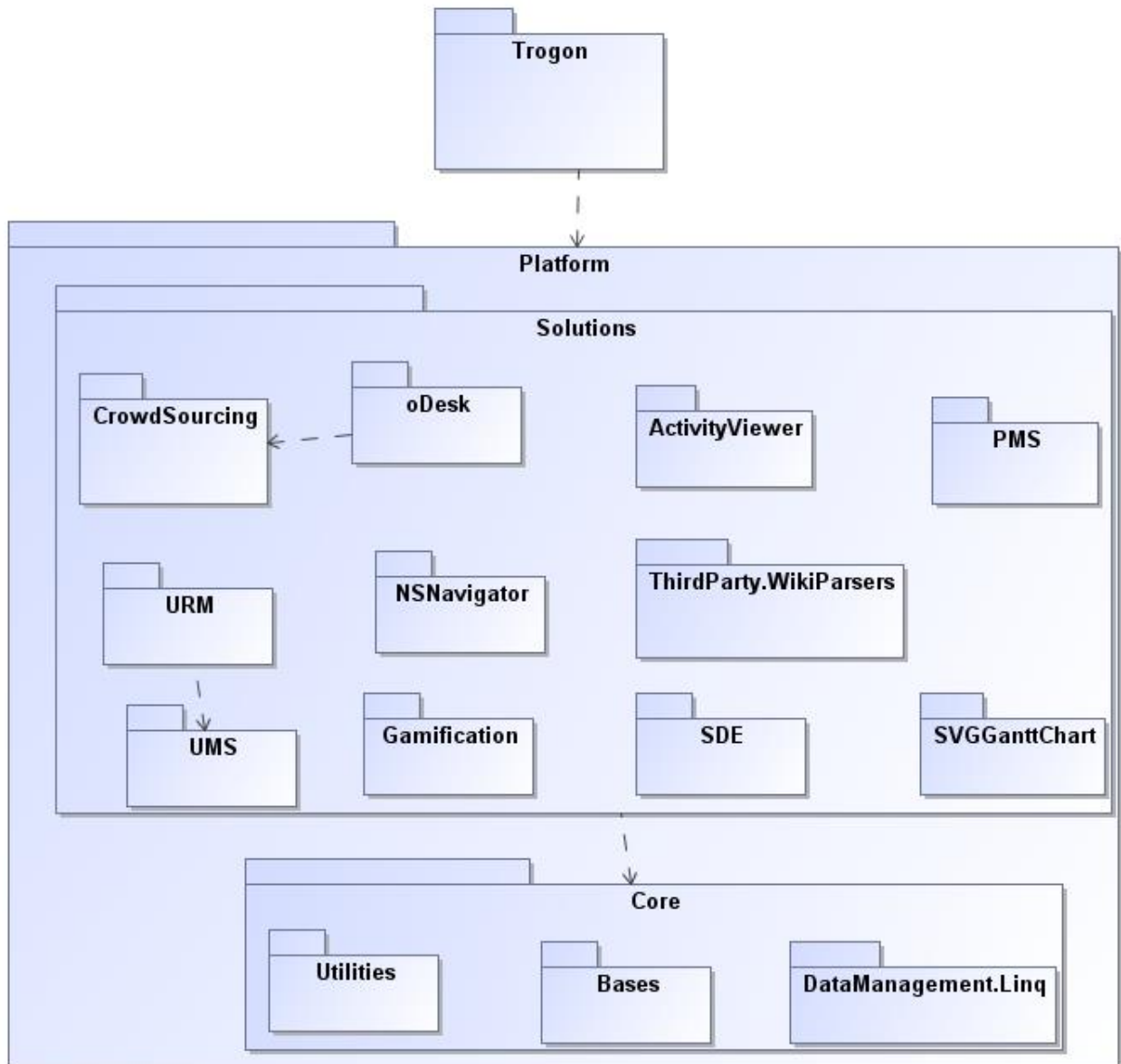
3.2 Sistemos išdėstymo vaizdas



3.1 pav. Aukšto lygio išdėstymo diagrama

Sistema priklausomai nuo poreikio gali veikti intranete arba internete. Sistema saugoma ugniasiene. Sistemos išplečiamumui naudojama klasikinė išdėstymo schema. Plečiant naudojimą ant kelių serverių, kodas replikuojamas kiekviename serveryje. Aukšto lygio išdėstymą apibendrinanti schema pateikta 3.1 pav.

3.3 Sistemos dinaminis vaizdas



3.2 pav. Bendra sistemos paketų diagrama

Sistemos paketų diagramą (3.2 pav.) sudaro šie elementai:

- Trogon – atvaizdavimo puslapiai.
- CrowdSourcing – bendruomeninių paslaugų valdymas.
- oDesk – specializuotas karkasas oDesk API integracijai į sistemą.
- ActivityViewer – įvykių atvaizdavimas.
- PMS – angl. „Project Management System“ duomenų ir komponentų lygmuo skirtas projektų valdymui.
- UMS – angl. „User management system“ vartotojų valdymas.
- URM – angl. „User Right Mangement“ sistemizuotas vartotojų teisių valdymas.
- NSNavigator – vartotojo pritaikomas greitosios navigacijos elementas.

- WikiParsers – wiki sintaksės (naudojama kaupti informacijai) interpretavimas.
- Gamification – lengvas žaidimizacijos elementų integravimas.
- SDE – angl. „Standart Date Engine“ greitas LINQ į duomenis lentelėmis ir formomis transformavimas.
- SVGGanttChart – Ganto grafiko atvaizdavimas užduotims.
- CORE – bazinė sistemos komponentų bazė.
- „Utilities“ – dažnai pasitaikančių funkcijų ir klasių paketas, kurios naudingos kuriant ne vieną sistemą.
- „Bases“ – įvairių bendrų komponentų ar elementų primityvai.
- „DataManagement.LINQ“ – paketas išplečiantis duomenų konteksto primityvą standartinėmis operacijomis su duomenų elementais ir kitos tam tikslui reikalingos klasės.

3.4 Komeriniai specializuoti programų paketai

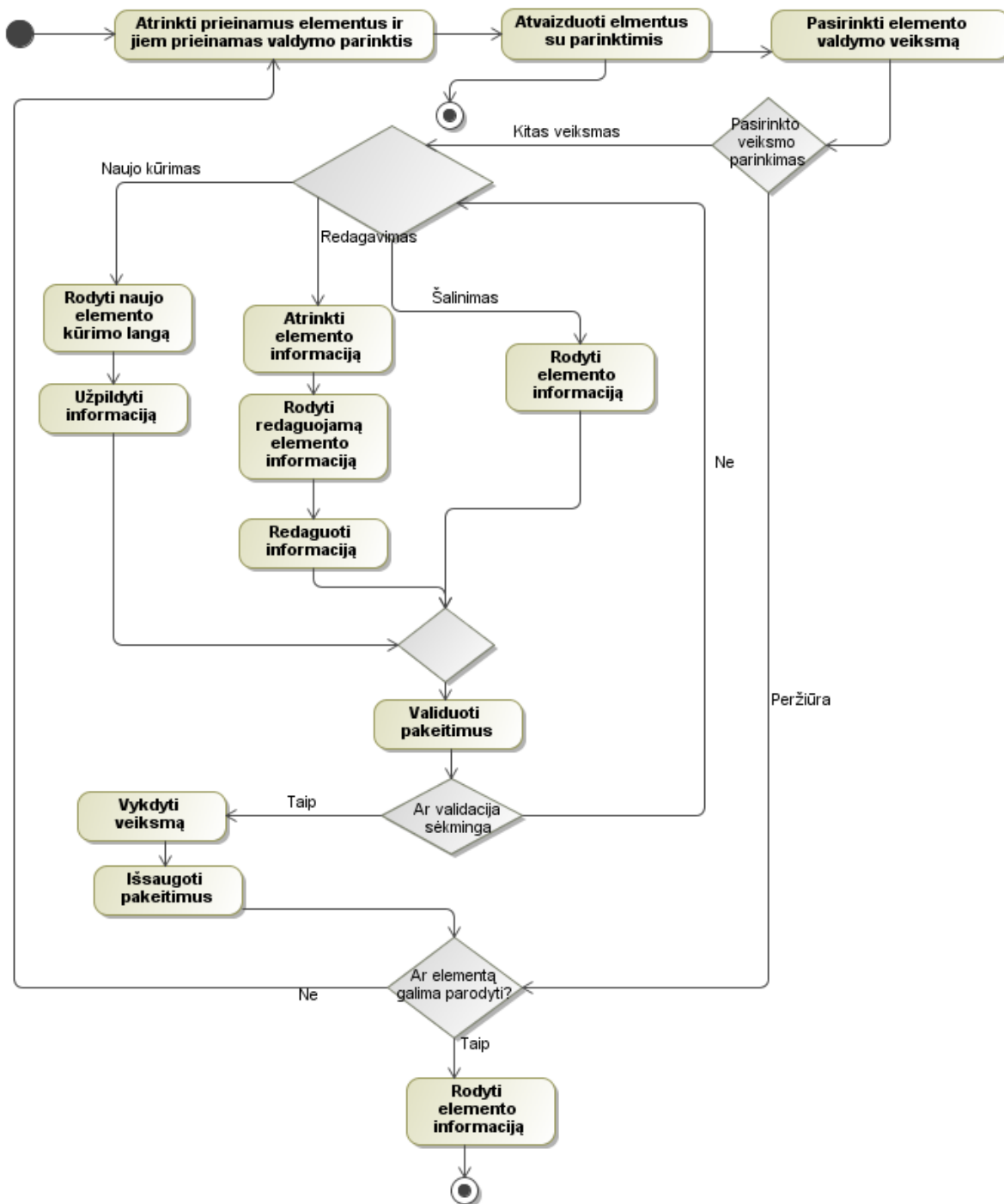
Sistemos realizacijai naudojama trečiųjų šalių komponentai:

jQuery, jQuery ui, oDesk API, DotNetOpenAuth, WikiCreola

Šie komponentai panaudoti greitesniam realizavimui ir sistemos standartizavimui, neperkuriant savo elementų, o panaudojant esamus ištestuotus ir vystomus.

3.5 Sistemos veiklos diagramos

3.5.1 Duomenų valdymas

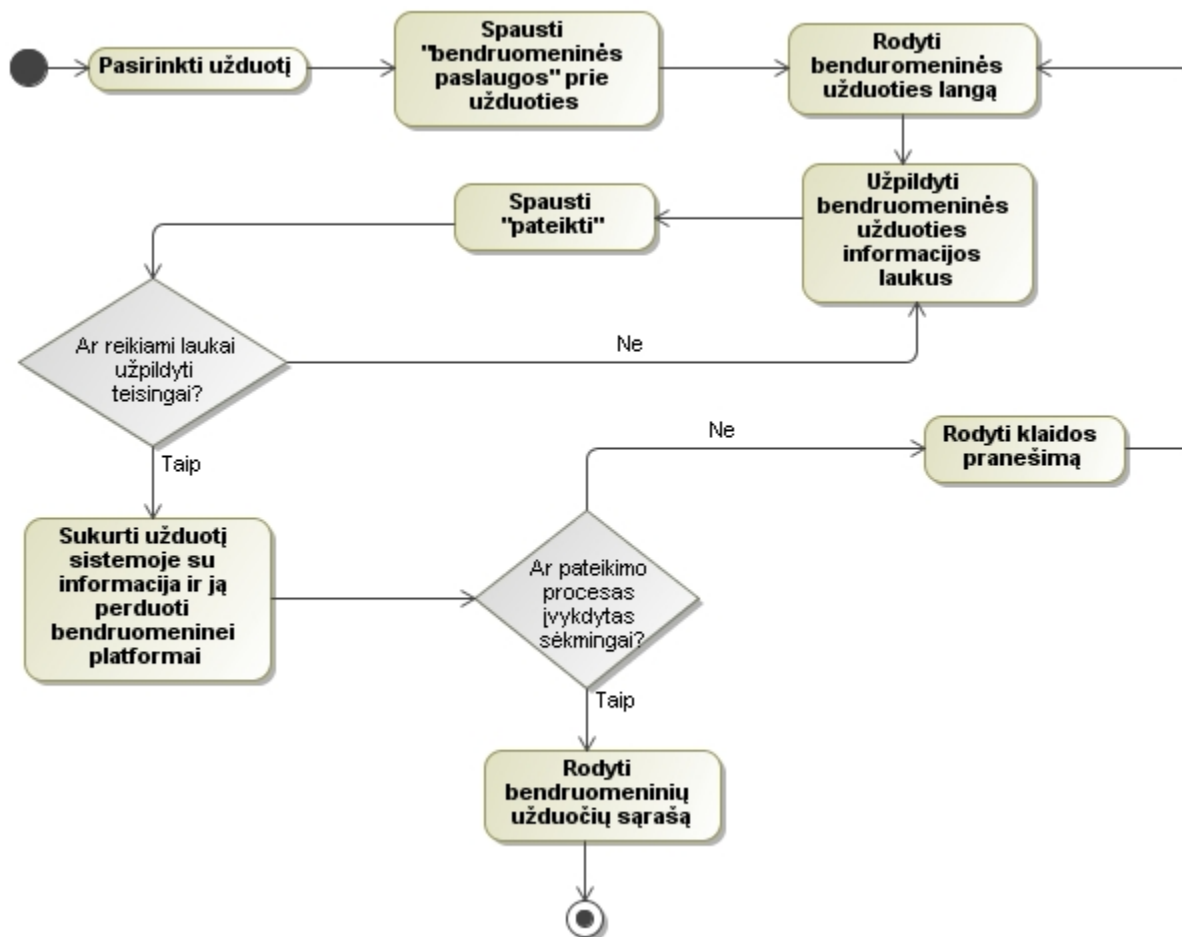


3.3 pav. Veiklos diagrama. Apibendrintas duomenų valdymas

Duomenų valdymas sistemoje atliekamas konkrečiuose puslapiuose. 3.3 pav. pateikta apibendrinta duomenų valdymo veiklos diagrama. Veiklos duomenų valdymo elementams apibendrintu lygiu yra vienodos. Čia apimami šie administravimo objektai: darbai, klientai, modeliai, projektai, puslapiai, užduotys, bendruomeninės užduotys.

Atėjus į duomenų valdymo puslapį atvaizduojami galimi valdyti elementai. Juos galima peržiūrėti, kurti, redaguoti, šalinti ir, užduočių atveju, pateikti bendruomenei. Valdymo pasirinkimai atvaizduojami konkrečioms objektams pagal jų palaikomas funkcijas. Naujo elemento kūrimo ar redagavimo atveju, pateikiamas konkretaus elemento informacijos laukų vaizdas, kuriame reikia užpildyti norimą/reikalaujamą informaciją. Jei užtikrinamos veiksmui reikalingos sąlygos – reikiami ir teisingi duomenys, trynimo patvirtinimas – vykdomas veiksmas ir sistemoje išsaugomi pakeitimai. Įvykdžius veiksmą, jeigu galima (nepašalintas elementas), rodoma elemento informacija.

3.5.2 Pateikimas bendruomenei



3.4 pav. Veiklos diagrama. Užduoties pateikimas bendruomenei

Užduoties pateikimas bendruomenei iš vartotojo perspektyvos yra paprastas ir lengvai prieinamas. Jo veiklos diagrama pateikta 3.4 pav. Žinoma šis veiksmas galimas tik esant prisijungus prie bendruomeninės sistemos (t.y. turint leidimus). Kitu atveju šis funkcionalumas nerodomas vartotojui.

Prisijungimo prie sistemos yra reikalaujama automatiškai, ateinant į bendruomeninių užduočių valdymo puslapį. Prisijungus duomenys išlaikomi sesijoje ir vėliau pasinaudojant turima informacija automatiškai atliekama autorizacija.

Esminė vartotojams veiklos dalis yra užduoties informacijos papildymas. Pateikiant užduotį, pagrindiniai parametrai apie užduoties informaciją yra panaudojami iš užduoties aprašo sudaryto kuriant užduotį PVS. Tačiau norint pateikti ją bendruomenei, reikalinga papildoma informacija, kad bendruomenės nariai galėtų įvertinti ar jie gali ir nori vykdyti šią užduotį. Vartotojo užpildomi tokie parametrai: planuojamas

vykdymo krūvis ir atlygis. Be šių duomenų sistema surenka papildomą informaciją pagal nustatymus, kuri nerodoma vartotojui, tokią kaip užduoties kategorija, atvaizdavimo tipas ir pan. Automatinis surinkimas panaudojamas, kad neapkrauti vartotojo ir supaprastinti procesą. Tai atlikti galima, nes sistemos dabartinė versija skirta darbui su interneto programavimo užduotimis.

Kuriant užduotį, ji yra išsaugoma sistemoje, taip pat informacija perduodama bendruomeninei platformai. Pastarojoje taip pat sukuriama užduotis su nurodytais parametrais ir įtraukiama į užduoties atlikėjų paieškos procesą.

Jei viskas atlikta sėkmingai (užpildyta reikiama informacija, veikia bendruomeninės sistemos procesai, ir turimos reikiamos teisės), pereinama į bendruomeninių užduočių sąrašą.

3.6 Svarbesnių paketų detalizavimas

3.6.1 „Trogon“ paketo detalizavimas



3.5 pav. „Trogon“ paketo detalizuota klasių diagrama

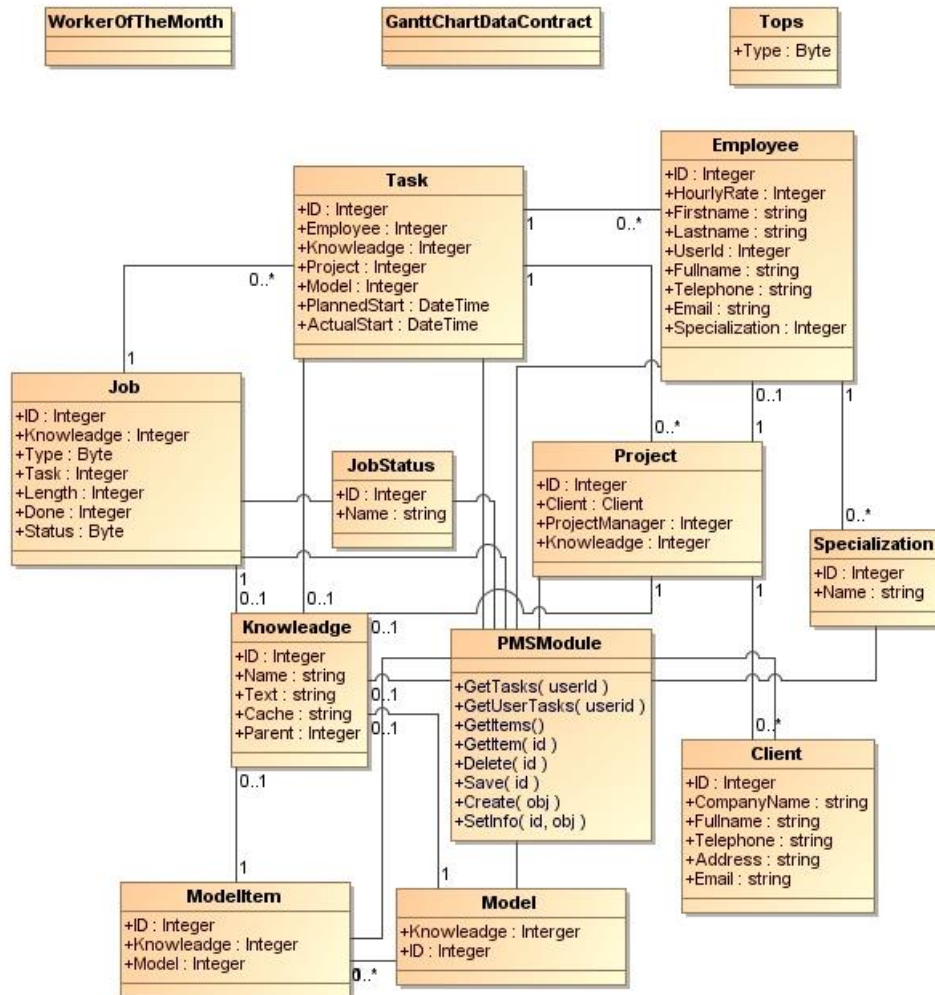
„Trogon“ paketas, kurio dalys pateiktos 3.5 pav., apima puslapius kuriamoje sistemoje. Jie įgyvendinami pasinaudojus visais anksčiau, 3.2 pav., pateiktais paketais.

Puslapių paskirtis:

- MyTasks – mano asmeninių užduočių sąrašas.
- Knowledge – organizacijos informacinis žinynas.
- Jobs – visų darbų sąrašas.
- Projects – visų projektų sąrašas.
- Clients – visų klientų sąrašas.
- Project – projekto užduočių sąrašas.
- Employees – visų darbuotojų sąrašas.
- Models – visų modelių sąrašas.
- Task – visi užduoties darbai.
- oDesk – bendruomeninių darbo paslaugų sistemos sąrašai.
- Security – vartotojų teisių valdymas.
- Pages – informaciniai puslapiai.
- MyWeekPlan – mano savaitės darbų planas.
- Login – prisijungimo puslapis.

- MyDesktop – mano darbastalis su veiklų atvaizdavimu.

3.6.2 „Solutions.PMS“ paketo detalizavimas



3.6 pav. „Solutions.PMS“ paketo detalizuota klasių diagrama

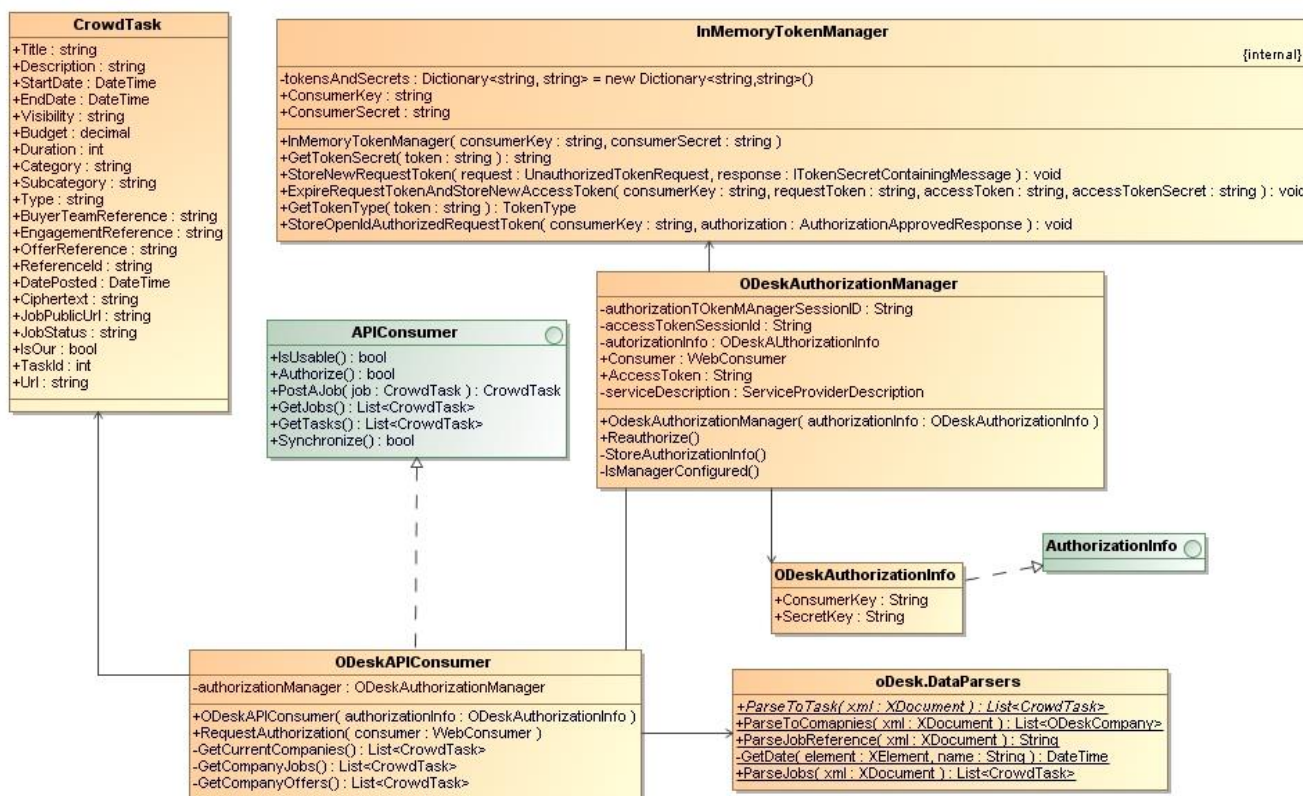
„PMS“ paketas, kurio klasių diagrama pateikta 3.6 pav., apima visus projektų valdymo sistemos duomenų tipus ir kitus specifinius elementus, kurie reikalingi projektų valdymo sistemos realizavimui. Viena išskirtinių savybių yra Knowledge klasė. Ji naudojama duomenims rinkti iš sistemos įvykių ir paruošti bendrai sistemos paieškai.

Projektai turi jiems priskirtas užduotis, bei užsakovą.

Užduotys skaidomos į darbus. Taip mažinamas sistemos vykdomas vienetas, kuris detalizuojamas ir kuriam vertinamos laiko sąnaudos reikalingos atlikimui. Darbuotojai gali tiksliau specifikuoti atliktas dalis. Užduotims nurodomi jų atlikėjai.

Pagrindinis duomenų valdymas atliekamas per PMSModule.

3.6.3 „Solutions.CrowdSourcing“ ir „Solutions.oDesk“ paketų detalizavimas



3.7 pav. „Solutions.CrowdSourcing“ ir „Solutions.oDesk“ paketų detalizuota klasių diagrama

Bendruomeninių paslaugų paketai siejasi tarpusavyje. Jų bendra klasių diagrama pateikta 3.7 pav. CrowdSourcing pakete išreiškiama bendravimo sąsaja ir jos atvaizdavimas sistemoje. APIConsumer sąsaja pateikia pagrindines funkcijas, kurias turi įgyvendinti realizacijoje konkretus bendruomeninių paslaugų sistemos teikimo elementas. Taip leidžiama prijungti kitas norimas sistemas ar kuriant platesnį integracinį mechanizmą nekeisti pačios sistemos veikimo. Naudojamos pagrindinės funkcijos skirtos užtikrinti prisijungimui/autorizacijai išorinėje sistemoje, jos statuso nustatymui ir duomenų mainams tarp sistemų. ODesk pakete realizuojamos, tiekėjo funkcijos reikalingos pateikti duomenims pagal apibrėžtą sąsają.

Naudojama bendruomeninės užduoties informacijos realizacija „CrowdTask“, pateikia naudotinus jos elementus reikalingus susiejimui su PVS esančiomis užduotimis ir atgalinio ryšio užtikrinimui.

InMemoryTokenManager leidžia išsaugoti autorizacijos parametrus sesijoje. Autorizacija realizuota ODeskAuthorizationManger. Saugojimas sesijoje užtikrina, kad vartotojai neturintys prieigos prie bendruomeninių paslaugų, jomis nesinaudotų.

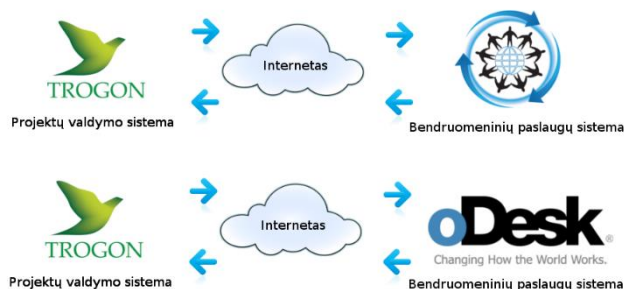
oDeskAPIConsumer pateikiami prieigos taškai panaudojami informacijos paėmimui ir pateikimui.

Čia vykdoma realus funkcijų įgyvendinimas, kuriuo sistema naudojasi per sąsają.

Informacijos, gaunamos iš trečiosios šalies, apdorojimui ir pavertimui į sistemoje naudojamą formatą realizuota DataParsers klasė. Joje elementai išgaunami iš informacijos srauto ir pateikiami sisteminiiais objektais.

3.7 Bendruomeninių paslaugų integravimas

Toliau apžvelgiamas projektų valdymo sistemos išplėtimas bendruomeninėmis paslaugomis, remiantis kuriama sistema pasiūlyta Ašeriškio & Tamošaičio (2012). Jos tikslas yra supaprastinti bendruomeninių resursų panaudojimą įmonėje.



3.8 pav. PVS sąveika su bendruomeninių paslaugų sistema (Ašeriškis & Tamošaitis, 2012)

Sistemai reikalinga turėti sąsają bendravimui su bendruomeninius išteklius teikiančia sistema. Apibendrinta modelio idėja pateikta 3.8 pav.

Pradinius sistemos įgyvendinimo atveju pasirinktas oDesk bendruomeninių paslaugų tiekėjas. Pagrindinė sistemos išplėtimo bendruomeninėmis paslaugomis dalis šiuo atveju yra informacijos tarp PVS ir oDesk sistemų sinchronizacija ir sistemų funkcijų tarpusavio papildymas.

Prie PVS prijungiant bendruomenines paslaugas reikalinga pasiekti tam tikrą (bazinį) funkcionalumą. Pagrindinės reikalingos funkcijos yra šios:

- Užduočių pateikimas.
- Užduočių atnaujinimas.
- Užduočių pašalinimas.
- Užduoties iš bendruomenės paėmimas.
- Užduočių peržiūra.

Trogon projektų valdymo sistemoje atvaizduojami naujausi Web programavimo srities darbai iš oDesk sistemos. Pretendavimas atlikti užduotį atliekamas oDesk sistemoje užpildžius reikiamą informaciją.

Užduotis pateikti galima tiek per Trogon sistemą tiek ir pačioje oDesk sistemoje. Užduočių nebūtina pateikti bendruomenei. Sukūrus užduotį Trogon sistemoje ji figūruoja tik joje. Pateikiant užduotį bendruomenei užpildoma papildoma informacija, tokia kaip: skiriamas biudžetas, reikalavimai užduoties atlikėjams.

Darbuotojų atrinkimas, interviu, kiti derinimo bei apmokėjimo darbai atliekami oDesk sistemoje.

Sinchronizuojama porcijomis – užduočių paketais, atnaujinama jei informacija yra pakitusi priklausomai nuo paskutinio atnaujinimo laiko. Informacijos sinchronizavimo metu surenkama reikalinga informacija iš oDesk platformos. PVS sukuriama ar atnaujinama bendruomeninė užduotis ir jų būsenos. Jei pradeda vykdyti bendruomeninė užduotis, ji bus dedama į bendruomeninių užduočių projektą. Šias užduotis reikia perskirti realiems projektams. Taip pat jas galima suskaidyti į smulkesnius darbus, tačiau pastarieji figūruos tik Trogon sistemoje ir neturės įtakos informacijai oDesk sistemoje.

3.8 Reaguojantis dizainas

Realizuojant sistemą panaudotas reaguojantis dizainas. Tai leidžia pritaikyti sistemos atvaizdavimą įvairiems įrenginiams (standartiniams kompiuteriams, planšetiniams kompiuteriams, išmaniesiems telefonams).

Reaguojama į įrenginio rezoliuciją ir pagal tai rodoma arba slepiama informacija. Taip vartotojui pateikiama visa aktuali informacija ir jis gali naudotis sistema turimame įrenginyje.

3.9 Objektiškai orientuoti stiliai

Sistemos elementams atvaizduoti panaudojami objektiškai orientuoti pakopiniai stiliai (OOCSS). Taip sumažinamas pasikartojamumas aprašant sistemos dizaino elementus ir tampa paprasčiau juos keisti vieningai visoje sistemoje.

3.10 Veikimo aplinka

Serveryje, kuriame diegiama sistema, naudojama Windows Server 2008 R2 operacinė sistema, bei Microsoft SQL Server 2008 R2 duomenų bazė. Serveris, turi galimybę būti pasiekiamas iš intraneto kompiuterių. Serveris yra prijungtas prie interneto, ir jame įdiegta sistema gali atlikti kreipinius į internetą.

Tinklo kompiuteriuose veikia Windows 7 operacinė sistema. Puslapių pasiekimui naudojamos atnaujinamos Google Chrome, Mozilla Firefox, bei Microsoft Internet Explorer 9 naršyklės.

3.11 Išvados

PVS su integruotomis bendruomeninėmis paslaugomis įgyvendinimui yra tinkama komponentinė architektūra, kurios esminės dalys yra projekto užduočių valdymas, jų informacijos rinkimas, bendruomeninės sistemos užduočių pateikimas, paėmimas, atnaujinimas. Baziniam sistemos funkcionalumui pasiekti yra tinkama integracija su oDesk bendruomeninių paslaugų platforma. Turint tokią integraciją, pateikiant užduotį bendruomenei pakanka papildomai įvesti užduoties vykdymo kainą ir planuojamą savaitinį darbo krūvį, reikalingą užduoties atlikimui.

4 TROGON SISTEMOS TYRIMAS

4.1 Specifikacijos atitikimas

Sistema atitinka sutartą specifikaciją, kuri projekto eigoje kiek kito, pritaikant analizės lydimus sprendimus, juos aptarus su užsakovu. Pakeitimai vykdyti siekiant įgyvendinti esminį funkcionalumą sistemos veikimui ir vartotojų patogumui.

4.2 Testavimas

Daliai sistemos elementų sudaryti vienetų testai, patikrinantys objektų vykdomų funkcijų veikimą.

Vartotojo sąsajai testuoti naudojamas Selenium WebDriver. Programiškai aprašoma testavimo eiga, tikrinami elementai, jiems suteikiamos reikšmės ir laukiami rezultatai.

Šie sukurti testai sėkmingai įvykdomi pateikiamoje programos versijoje. Tačiau, norint turėti lengviau palaikomą sistemą, reiktų didinti šiais testais padengiamo kodo kiekį.

Kita testavimo dalis atlikta pasinaudojant statinę kodo analizę.

4.2.1 Statinė kodo analizė

Analizuojami pagrindiniai sistemos komponentai pagal prižiūrimumą, ciklo matinį sudėtingumą, paveldėjimo lygį, klasių susietumą.

Į analizę nėra įtraukiami ASP.NET puslapiai – internetinis sistemos puslapis išnaudojantis komponentus. Šiuose puslapiuose aprašomas elementų, kurių įgyvendinimas vykdomas komponentuose, išdėstymas ir konfigūravimas.

Toliau apžvelgiami minėti rodikliai išdėstant komponentus visuose grafikuose ta pačia tvarka – pagal kodo eilučių skaičių realizuotoje sistemoje.

4.2.1.1 *Pagrindiniai sistemos komponentai*

Naudojant statinę kodo analizę, analizuojami šie pagrindiniai sistemos darbui naudojami komponentai: WikiParserWrapper, SVGGanttGraph, CrowdSourcing, NS-ActivityViewer, Toolbar, NS-Navigator, SDK\Core, SDE, PMS.

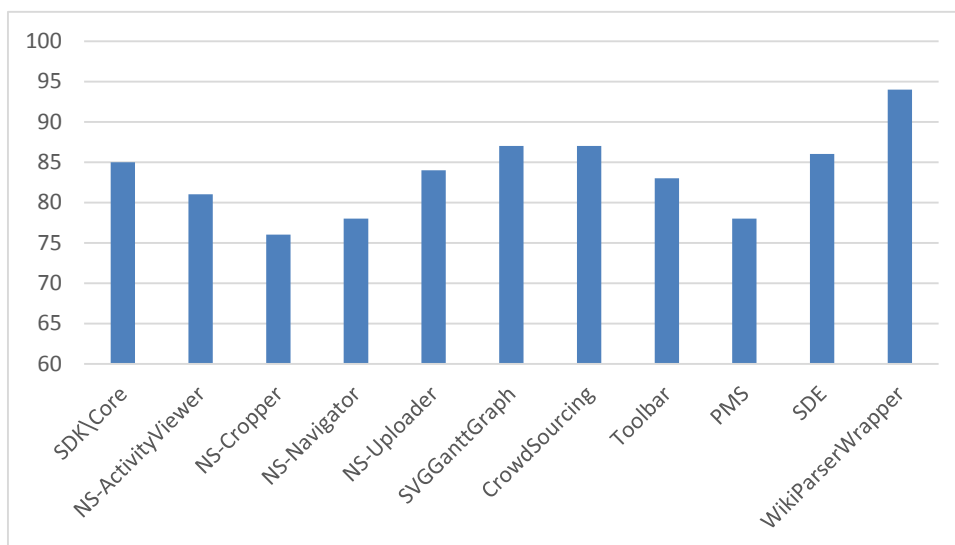
4.2.1.2 *Kodo eilučių skaičius*

4.1 lentelė. Komponentų kodo eilučių skaičius

Komponentas	Kodo eilučių skaičius
WikiParserWrapper	9
SVGGanttGraph	116
NS-Uploader	122
NS-Cropper	173
CrowdSourcing	200
NS-ActivityViewer	228
Toolbar	331
NS-Navigator	468
SDK\Core	847
SDE	937
PMS	2155

Sistemos realizacijai naudojami sukurti komponentai sudaryti iš 5586 kodo eilučių. Detalesnis išskaidymas pateiktas 4.1 lentelėje. Patiems komponentams realizuoti reikalingas eilučių skaičius priklauso nuo funkcionalumo, kurį jis pateikia. Daugiausia kodo realizuota PMS – projektų duomenų valdymo – komponente. Kiti Komponentai pasinaudoja suteikiama prieiga pasiekti duomenis per šį komponentą.

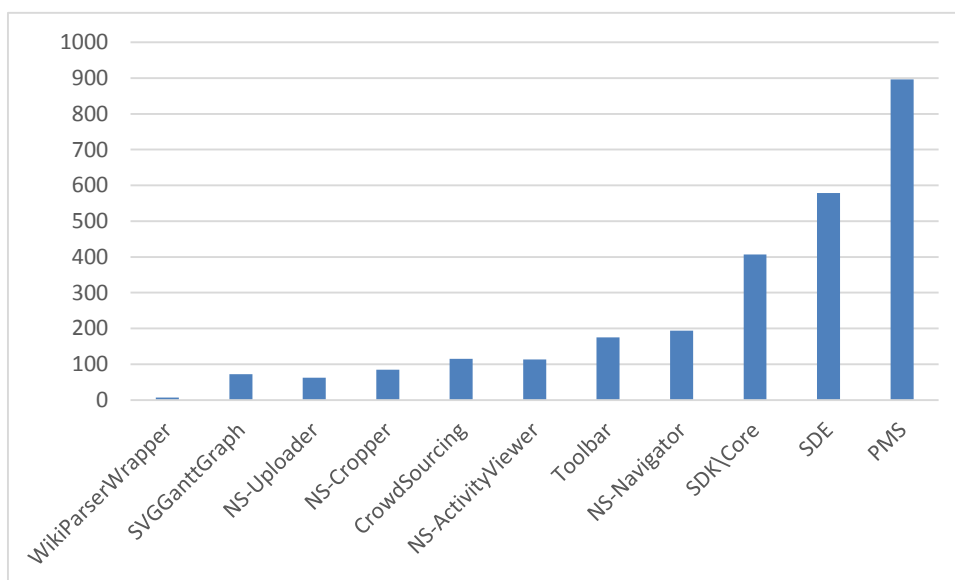
4.2.1.3 Prižiūrimumas



4.1 pav. Sistemos komponentų prižiūrimumo indekso absoliutinės reikšmės

Sistemos prižiūrimumo indekso absoliutinės reikšmės pateiktos 4.1 pav.. Prižiūrimumo vidutinis indeksas 84. Taip pat kaip matoma iš absoliutinių reikšmių visuose komponentuose jis didesnis nei 75. Taigi sistema yra prižiūrima. Ir tinkama plėtojimui ateityje.

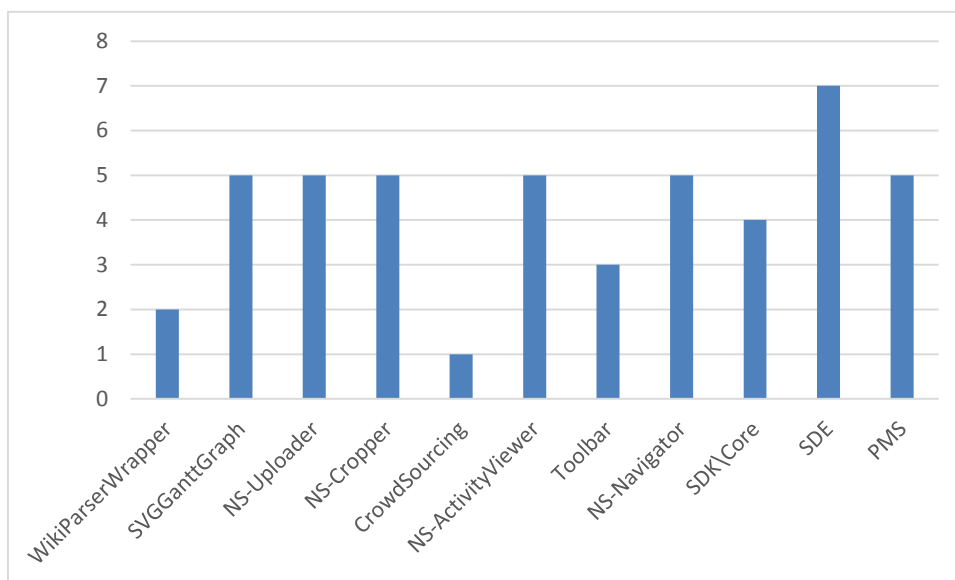
4.2.1.4 Ciklominis sudėtingumas



4.2 pav. Sistemos komponentų ciklominio sudėtingumo absoliutinės reikšmės

Sistemos ciklominio sudėtingumo indekso absoliutinės reikšmės pateiktos 4.2 pav.. Sistemos komponentų lygyje vyrauja aukštas ciklominis sudėtingumas. Tai parodo kad patys komponentai būtų sunkiai arba visai netestuojami. Tačiau žiūrint ciklominį sudėtingumą komponentų klasėse, jis smarkiai mažėja. Ir šiuo atveju tik dalis klasių pagal ciklominio sudėtingumo reikšmę būtų sunkiai testuojamos.

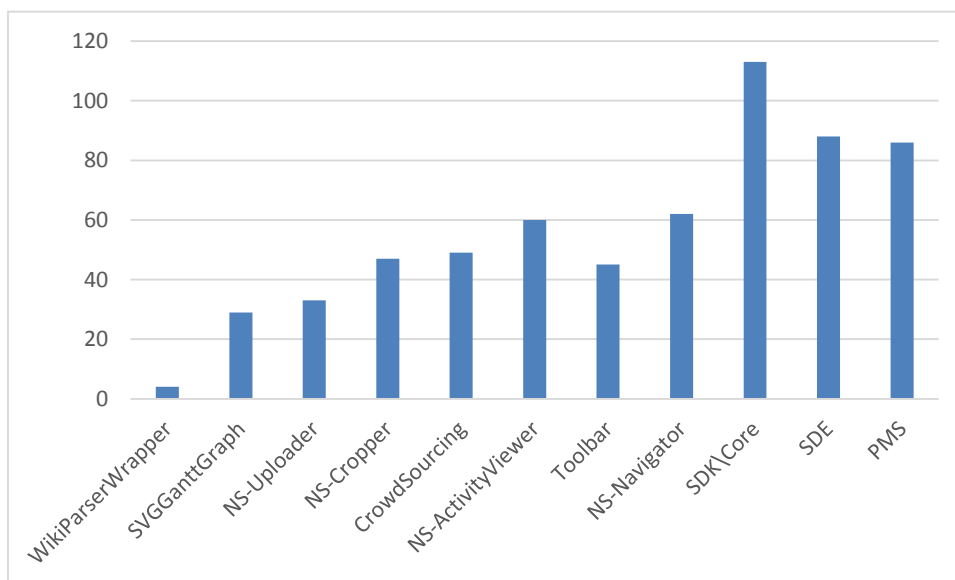
4.2.1.5 Paveldėjimo gylis



4.3 pav. Sistemos komponentų paveldėjimo gylio absoliutinės reikšmės

Sistemos komponentų paveldėjimo gylio absoliutinės reikšmės pateiktos 4.3 pav.. Sistemoje yra išnaudojamas paveldėjimas, tačiau reikia atkreipti dėmesį, kad paveldima ir iš standartinių .Net karkaso elementų, taigi, prižiūrimų objektų paveldėjimas klasėje kiek žemesnis. Prižiūrimumui tai problemų nekelia, o tik supaprastina kodo supratimą.

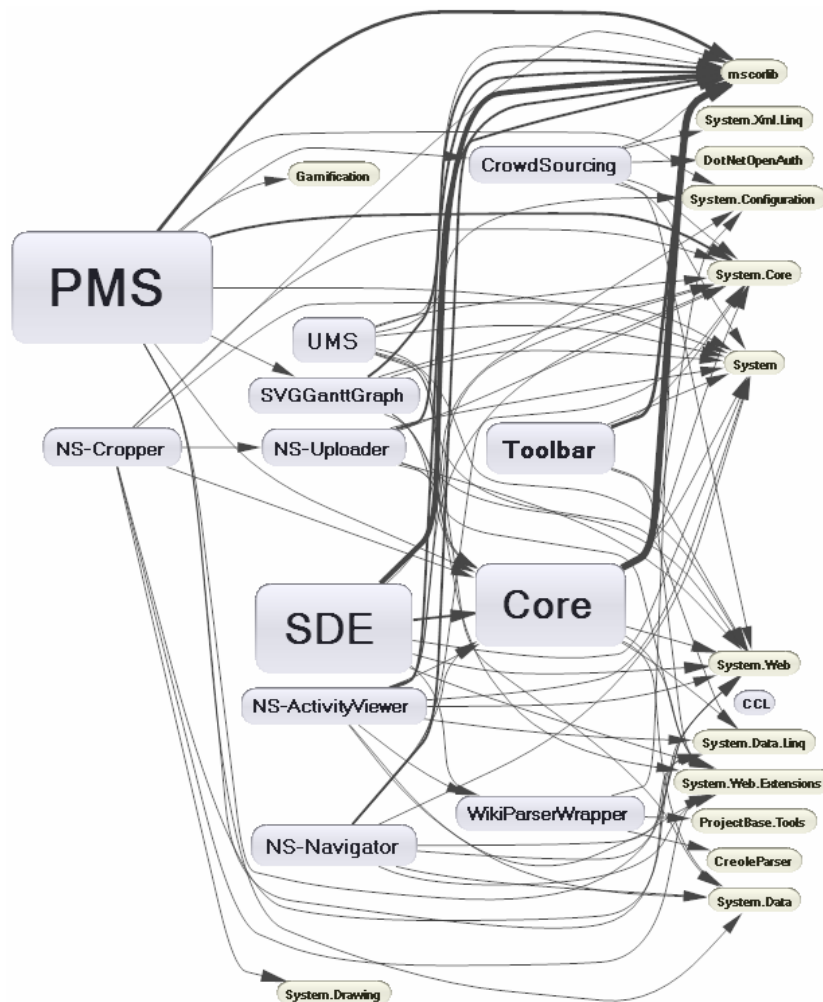
4.2.1.6 Klasių susietumas



4.4 pav. Sistemos komponentų klasių susietumo absoliutinės reikšmės

Klasės komponentuose yra smarkiai susietos tarpusavyje. Klasių susietumo absoliutinės reikšmės pateiktos 4.4 pav.. Gerinant prižiūrimumo indeksą ir tuo pačiu testuojamumą, reiktų keisti susietas klases, mažinant susiejimo lygį. Atskiros klasės turi mažesnę susietumą, tačiau bendrai komponente jis gaunamas aukštas. Taip pat galimai reiktų apsvarstyti labiausiai susietų komponentų išskaidymo į dar detalesnes sritis galimybę.

4.2.1.7 Ryšiai tarp komponentų



4.5 pav. Sistemos komponentų tarpusavio ryšiai

Sistemoje išnaudojami ryšiai tarp pagrindinių/bazinių ir kitų sistemos komponentų. Komponentų ryšių diagramoje, pateiktoje 4.5 pav. matomas bazinių komponentų išnaudojimas daugelyje kitų komponentų. Tai parodo teisingą išskaidymą.

4.3 „oDesk“ API problemos

Naudojantis trečiosios šalies sistemomis reikia prisitaikyti prie jos teikiamo funkcionalumo. Šiuo atveju oDesk pateikia bendravimui skirtą sąsają (angl. „API“). Ši sąsaja yra nuolat tobulinama ir plėtojama, tačiau dalies norimo funkcionalumo pasiekimas yra sudėtingas ar neįmanomas. Todėl negalima įgyvendinti nuoseklios sąsajos vien tik PVS – norint pasiekti dalį funkcionalumo tenka naudotis pačia bendruomeninių išteklių platforma.

Čia turima omenyje funkcionalumas toks kaip:

- Bendruomeninės užduoties apsiėmimas vykdyti tiesiai iš PVS.
- Bendruomeninės užduoties, kurios atlikimą norima apsvarstyti, bet nesukurtos mūsų, informacija apie darbuotojus nėra prieinama.
- Užduoties atlikėjų informacijos pateikimo trūkumas.

Kita API problema kylanti iš to paties nuoseklaus plėtimo – vienu metu skirtingiems elementams veikai skirtingų API versijų kreipiniai, nes ne viskas yra iškart atnaujinama.

4.4 Sistemos tobulinimai ateityje

4.4.1 Sinchronizavimas

Prie sistemos prijungiant papildomas bendruomeninių paslaugų teikimo sistemas ir jas kelias naudojant vienu metu reikalingas sistemos sinchronizavimo metodo išplėtimas. Tai būtų reikalinga, nes dabartinė informacijos pateikimo iš PVS bendruomeninei platformai dalis atliekama realiu laiku, o perduodant kelioms sistemoms, norint užtikrinti trumpą atsako laiką, reiktų šios informacijos perdavimą vykdyti asinchroniškai.

Tam būtų galima papildomai sukurti sinchronizavimo komponentą, kurio pagalba būtų laikinai kaupiami pakeitimai ir perkeliama į sistemas. Tai reikalinga, nes yra galimybė, jog kuri nors iš panaudojamų išorinių sistemų gali laikinai neveikti, o informacija kitose sistemos jau būti pateikta. Norint užtikrinti bendrumą reiktų vykdyti pakartotinę nepavykusių elementų sinchronizaciją.

4.5 Išvados

Sistema atitinka galutinę specifikaciją. Komponentų lygiu sistema yra sunkiai prižiūrima tačiau klasių lygyje yra prižiūrima ir tinkama toliau vystyti. Reiktų pertvarkyti komponentus, kad sistema jų lygyje taip pat būtų prižiūrima. Dalyje klasių reiktų keisti įgyvendinimą, kad jas būtų galima lengviau padengti testais. Būtų naudingas didesnis visos sistemos padengimas testais.

5 BENDRUOMENINIŲ PROJEKTŲ KŪRIMO PROCESŲ TYRIMAS

5.1 Tikslai

Eksperimento tikslas yra ištirti bendruomenės narių pateikiamų sprendimų pasiskirstymą projekte maste, narių prisijungimą prie projekto.

Siekiama nustatyti:

1. Kaip kinta užduočių atlikimo per dieną skaičius projekto pradžioje (pirmi trys mėnesiai).
2. Kaip kinta skirtingų užduočių atlikėjų per dieną skaičius projekto pradžioje (pirmi trys mėnesiai).
3. Kokią darbo dalį bendruomeniniuose projektuose atlieka projekto vystymo pradininkai (per pirmąjį mėnesį pradėję atlikti užduotis).
4. Kokią darbo dalį bendruomeniniuose projektuose atlieka pagrindiniai projekto atlikėjai (atlikę > 2% projekto užduočių).
5. Kokią darbo dalį bendruomeniniuose projektuose atlieka ne pagrindiniai projekto atlikėjai (atlikę ≤ 2% projekto užduočių).
6. Kokia yra koreliacija tarp užduočių atlikėjų ir vidutinio atliktų užduočių skaičiaus projekto eigoje.

5.2 Duomenų parinkimas

Šiam tikslui panaudojama atviro kodo principu kuriamų projektų informacija. Informacija imama iš projektų kodo saugyklos GitHub.

GitHub yra didelė (šiuo metu yra saugoma virš 6 milijonų projektų ar jų variantų (Mojombo, 2013) atviro kodo kūrimo programų duomenų bazė. Ja naudojasi daug programuotojų. Taip pat daugelis tyrėjų naudoja GitHub atviro kodo ir bendruomeninio kūrimo sričių tyrimui:

- „Pasinaudojant Github programuotojams skirtu API, buvo surinkta profilių informacija“, kuri panaudota „kad geriau suprasti kaip „socialinės“ funkcijos, tokių sistemų kaip Github, veikia bendradarbiavimą“ (Weber, 2012).
- „Duomenis mes surinkome pasinaudojant viešai prieinamu Github API, kuris pateikia tokią pačią informaciją, kaip ir Github internetinis puslapis“ (Takhteyev & Hiltz, 2010).
- Ir kitos publikacijos, kaip: McDonald & Goggins (2013, Balandis); Marlow, Dabbish & Herbsleb (2013, Vasaris); Thung, LO & JIANG (2013); Begel, Bosch & Storey (2013); Dabbish, Stuart, Tsay & Herbsleb (2012, Vasaris); Choi, Choi, Moon, Hahn & Kim (2013, Vasaris); Majumder, Datta & Naidu (2012, Rugpjūtis) ir kita.

Laikoma, kad į saugyklą pridėdami sprendimai yra baigtinės užduočių pateiktys. Pasinaudojant bendruomeninių projektų kūrimo principais, kodo pakeitimams po pateikimo užsakovui (šiuo atveju bendruomeniniam projektui) būtų formuluojami naujų užduočių pavidalu.

Tyrimo metu analizuojami 5 projektai, kurių trumpi aprašai pateikti 5.1 lentelėje.

5.1 lentelė. Eksperimentinio tyrimo projektai

Pavadinimas	Aprašas
Bootstrap	Karkasas internetinių puslapių vartotojo sąsajai kurti
jQuery	Tarpnaršykinė javascript biblioteka
Mono	Tarp platforminis .Net karkasas ir darbui su juo skirti įrankiai

Typo	Tinklaraščio variklis
Spree	E-komercijos sistema

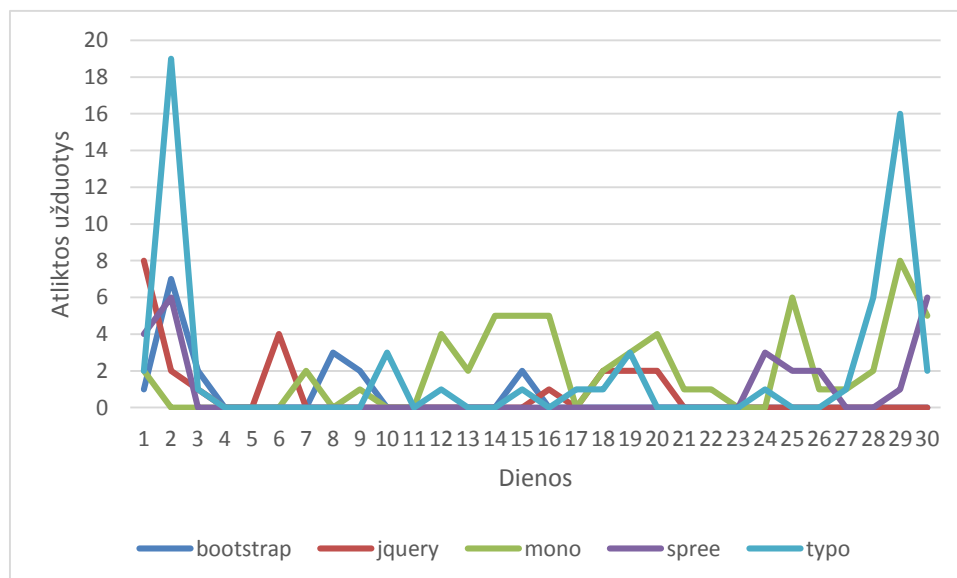
Projektus kuria bendruomenė. Ji prisideda prie iniciatoriaus sukurtos kodų bazės plėtojimo. Visų parinktų projektų trukmė yra didesnė nei 2 metai. 4 projektai yra susiję su internetiniais puslapiais. Kad būtų galima aiškiau išskirti bendras projektų tendencijas, 1 projektas parinktas skirtas programinei įrangai tiesiogiai nesisiejant su internetiniais puslapiais.

5.3 Duomenų paruošimas

Projektų informacija paimama iš GitHub saugykloje laikomų projektų panaudojant Git versijavimo sistemos komandas. Tada laikant, jog atskiri pateikimai yra atskiros užduotys, jie perkeliama į Trogon sistemą tolimesnei analizei. Reikia atkreipti dėmesį, kad taip importuojamoms užduotims negalima priskirti laikinių charakteristikų (pradžios datos ir vykdymui skirto laiko).

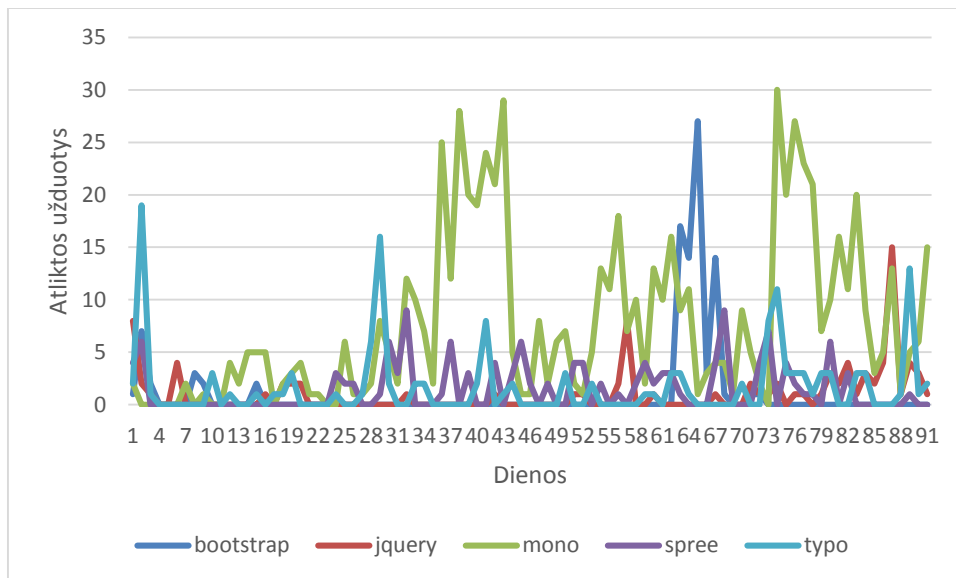
5.4 Pradinė projekto stadija

5.4.1 Užduočių atlikimo kitimas



5.1 pav. Užduočių pateikimo dažnumas pirmojo mėnesio laikotarpiu

Analizuojant užduočių dienomis kiekius pateikiamus pirmojo mėnesio laikotarpio grafike (5.1 pav.), pirmosiomis projekto dienomis pastebimas didesnis atliekamų užduočių kiekis. Tačiau vėliau jis sumažėja. Taip atsitinka, nes primomis dienomis pateikiamos susikaupę turimos ir pradinio projekto pataisymai reikalingos užduotys.



5.2 pav. Užduočių pateikimo dažnumas pirmųjų trijų mėnesių laikotarpiu

Užduočių pateikimo dažnumas pirmųjų trijų mėnesių laikotarpiu (5.2 pav.), rodo jog atlikimo kiekis laikotarpyje dalyje projektų išlieka panašus, tačiau kituose projektuose vyksta dideli šuoliai.

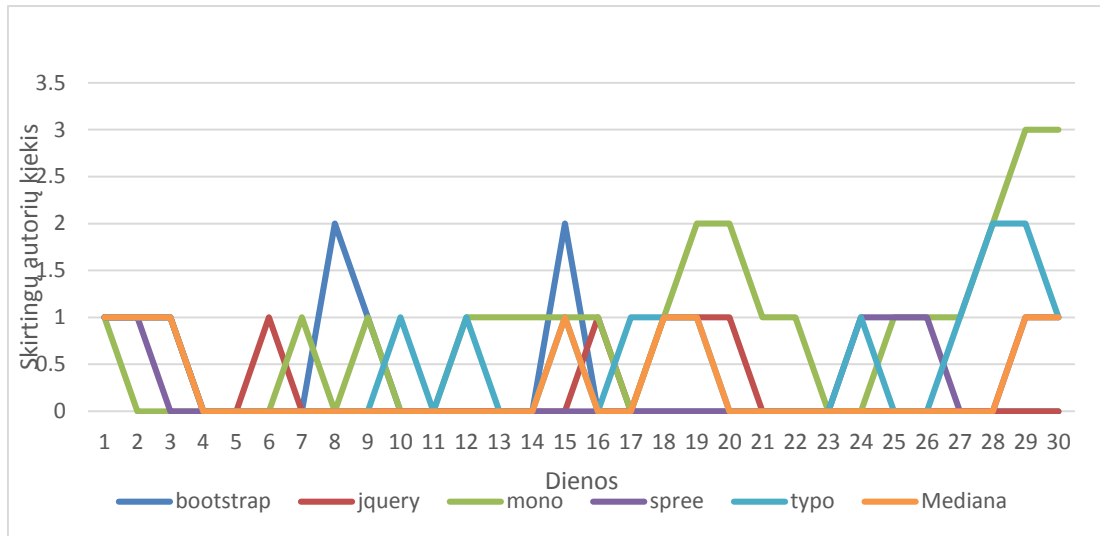
Projekto pradžioje užduočių įvykdymo kiekiai varijuoja. Visuose projektuose matomas palaikomas bent pastovus tempas ir atsirandantys suaktyvėjimai. Per pirmuosius tris mėnesius projektuose įvykdyti užduočių kiekiai pateikti 5.2 lentelėje.

5.2 lentelė. Pirmųjų trijų mėnesių užduočių kiekiai projektuose

Projektas	Atliktų užduočių kiekis
Bootstrap	95
jQuery	89
Mono	685
Typo	131
Spree	147

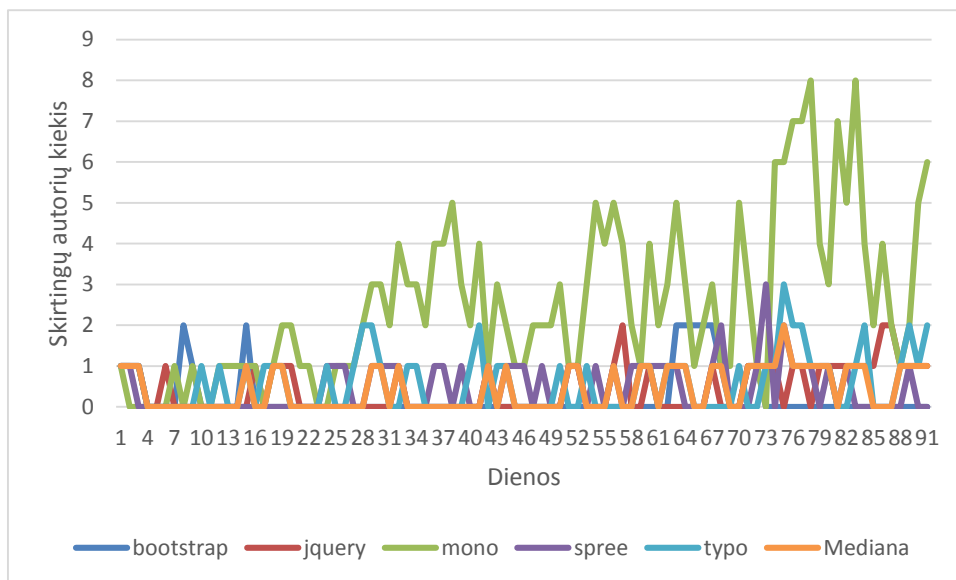
Pateiktų užduočių mediana 131. „Mono“ projekte buvo atliekama žymiai daugiau užduočių.

5.4.2 Autorių kiekio kitimas



5.3 pav. Autorių kiekiai pirmojo mėnesio laikotarpiu

Skirtingų užduočių atlikėjų skaičius pirmojo mėnesio laikotarpiu, kurio apibendrintas grafikas pateiktas 5.3 pav., mažai kinta. Dažnai projekto pradžioje užduotis atlieka tik pagrindinis projekto iniciatorius, ar jų grupelė, t.y. vienas ar du asmenys.



5.4 pav. Autorių kiekiai pirmųjų trijų mėnesių laikotarpiu

Trijų mėnesių laikotarpyje, kurio apibendrintas grafikas pateiktas 5.4 pav., dauguma projektų vis dar turi mažai kintantį užduočių atlikėjų kiekį per dieną. Išimtis yra „Mono“ projektas, kuriame jau nuo 28 dienos skirtingų autorių skaičius ima augti ir eigoje pasiekia 8.

5.4.3 Autoriai ir jų atliekamos užduotys

Analizuojant projektų vykdymo pradžią, reikia pažvelgti ir į autorių bei jų atliekamų užduočių kiekius.

5.3 lentelė. Pirmojo mėnesio užduočių autoriai ir jų atliktos užduotys

Projektas	Autorių skaičius	Atliktos užduotys
Bootstrap	2	17
jQuery	1	22
Mono	6	60
Typo	2	56
Spree	1	23

Apibendrinti pirmojo mėnesio užduočių ir autorių kiekiai pateikti 5.3 lentelėje. Projektų pradžioje autoriai išlieka tie patys, kurie ir pradėjo projekto veiklą. Tačiau „Mono“ projekte, kuris iš pradžių sulaukė didesnio bendruomenės dėmesio, pradeda dalyvauti ir nauji užduočių atlikėjai.

5.4 lentelė. Pirmųjų trijų mėnesių užduočių autoriai ir jų atliktos užduotys

Projektas	Autorių skaičius	Atliktos užduotys
Bootstrap	2	95
jQuery	4	89
Mono	32	685
Typo	3	147
Spree	4	131

Apibendrinti pirmojo mėnesio užduočių ir autorių kiekiai pateikti 5.35.4 lentelėje. Šiuo laikotarpiu projektai pritraukia šiek tiek naujų užduočių atlikėjų, tačiau pagrindiniai užduočių atlikėjai išlieka tie patys. Kaip ir pirmojo mėnesio atveju, matomas „Mono“ projekto naujų užduočių atlikėjų įsijungimas įtakojantis ir atliktų užduočių kiekį.

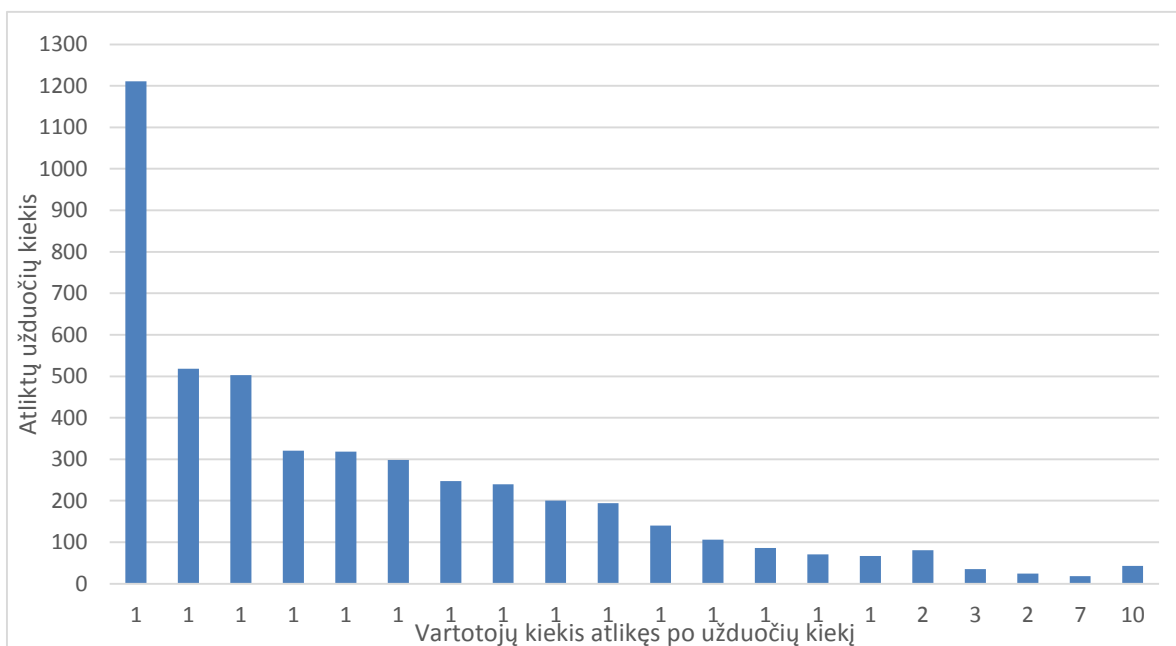
5.5 Projektas visoje eigoje

Toliau analizuojami vartotojų indėliai viso projekto eigoje.

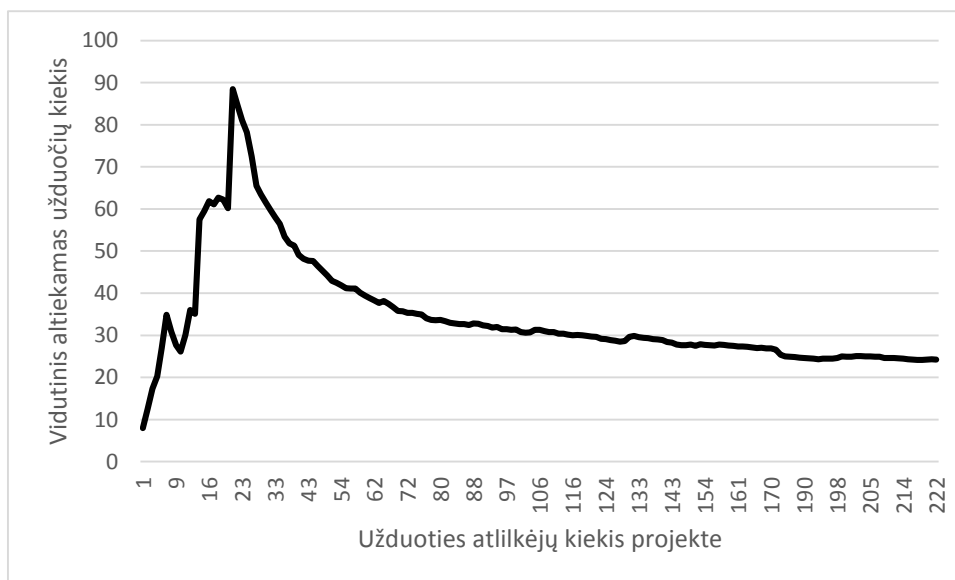
Iš pradžių pateikiami ir analizuojami atskirų projektų duomenys. Vėliau apibendrinami dėsniumai tarp projektų.

Viso užduotis atliko 306 asmenys. Atlikta užduočių 4599. Projektas vykdomas 746 dienas.

5.5.2 jQuery



5.7 pav. „jQuery“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 101, dešimčių grupėmis, imant vidurkį)



5.8 pav. „jQuery“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu

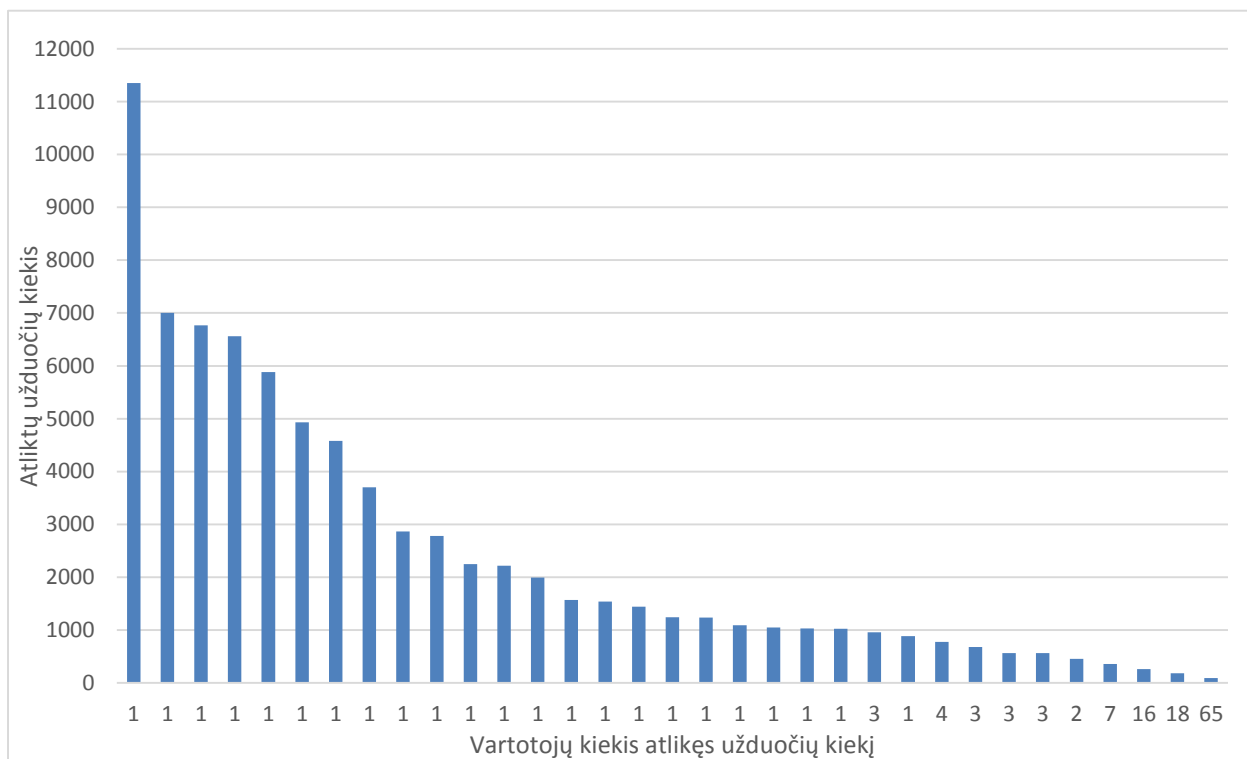
Kaip matoma „jQuery“ projekte vartotojų atliktų užduočių apibendrintame grafike (5.7 pav.), viso užduotis atliko 222 asmenys. Atlikta užduočių 5393. Projektas vykdomas 2607 dienas.

Projekto iniciatorius yra vienas užduočių atlikėjas, jis įgyvendino 22% užduočių.

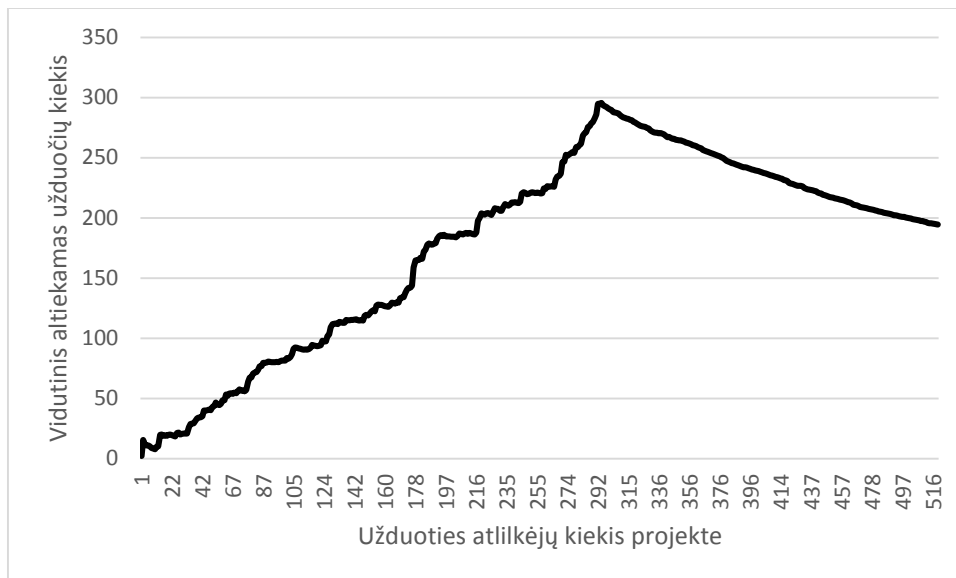
Pagrindiniai atlikėjais galime laikyti 11, kurie atliko daugiau nei po 100 užduočių. Daugiausia vienas asmuo atliko 1211 užduotis (jis buvo projekto iniciatorius). Jie bendrai atliko 4190 užduočių. Kiti vartotojai atliko bendrai 1203 užduotis, t.y. 22% viso projekto.

Vidutinis užduočių atlikimo priklausomybės nuo užduočių atlikėjų skaičiaus grafikas pateiktas 5.8 pav.. Projekto pradžioje šis dydis smarkiai auga. Galim spręsti, kad projektas pritraukia reikiamus atlikėjus. Tačiau projekto eigoje pasiekus 23 užduočių atlikėjus, kas įvyko daugiau nei po trijų metų vystymo, vidurkis smarkiai krenta. Prisijungusieji atlikėjai atlieka mažiau užduočių.

5.5.3 Mono



5.9 pav. „Mono“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 1001, šimto grupėmis, imant vidurkį)



5.10 pav. „Mono“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu

Kaip matoma „Mono“ projekte vartotojų atliktų užduočių apibendrintame grafike (5.9 pav.), užduotis atliko 519 asmenų. Atlikta užduočių 100993. Projektas vykdomas 4355 dienas.

Projekto iniciatoriais galime laikyti 6 atlikėjus, jie bendrai įgyvendino 10% užduočių.

Pagrindiniai atlikėjais galime laikyti 12, kurie atliko daugiau nei po 2000 užduočių. Daugiausia vienas asmuo atliko 11355 užduotis (jis nebuvo projekto iniciatorius). Bendrai pagrindiniai užduočių atlikėjai atliko 60879 užduotis. Kiti vartotojai atliko bendrai 40114 užduotis, t.y. 40% viso projekto.

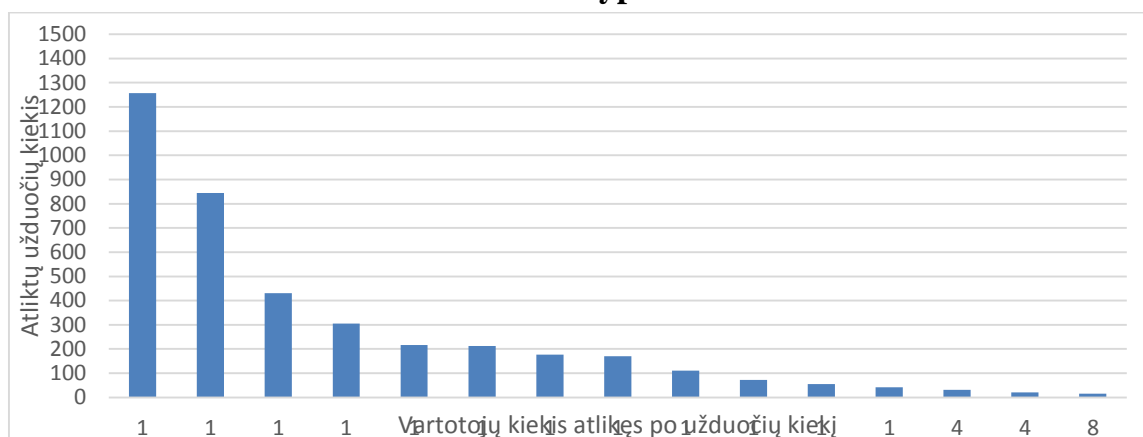
Vidutinis užduočių atlikimo priklausomybės nuo užduočių atlikėjų skaičiaus grafikas pateiktas 5.10 pav.. Projekto eigoje vidurkis vartotojui smarkiai gauna net ir didėjant vartotojų skaičiui. Galim spręsti, kad projektas pritraukia pajėgius atlikėjus. Projekto eigoje pasiekus 300 užduočių atlikėjų, kas įvyko daugiau nei po devynerių metų, vidurkis pradeda mažėti. Prie projekto prisijungiantieji aktyviai vykdo užduotis.

Užduočių atlikimas yra aktyvus viso projekto metu imant visą jo mastą. Žinoma yra ir mažesnio aktyvumo sričių. Bet matoma kad bendruomenei projektas yra įdomus ir naudingas, dėl to aktyviai vykdomas.

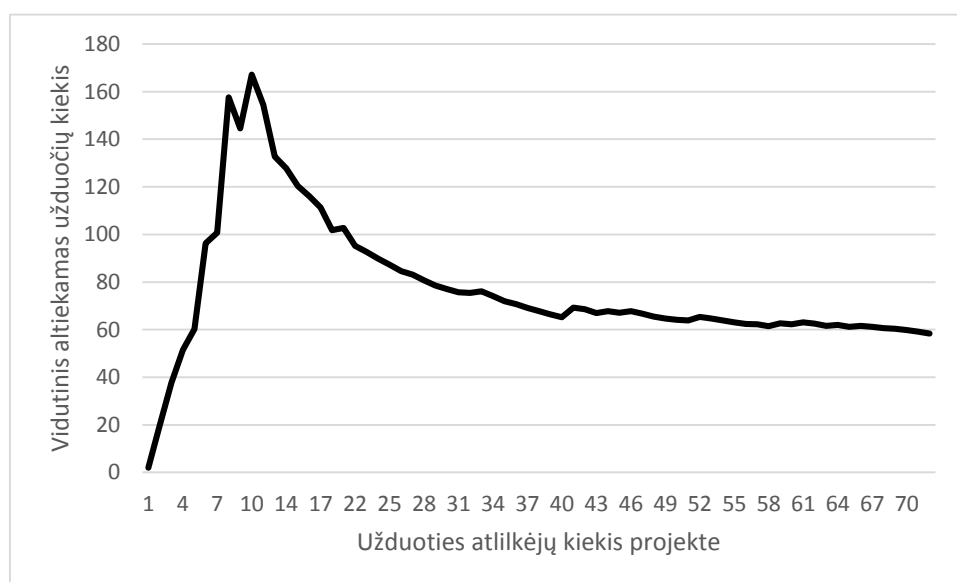
Nauji vykdytojai prie projekto jungiasi aktyviai nuo pat projekto pradžios ir aktyvumas išlieka projekto vystymo laiku. Projektui pereinant į tolimesnį etapą skirtingų vartotojų kiekis ima mažėti.

Vidutinis užduočių atlikimo priklausomybės nuo užduočių atlikėjų skaičiaus grafikas pateiktas 5.12 pav.. Projekto pradžioje šis dydis svyruoja gan smarkiai, tačiau pradėdant prisidėti naujiems užduočių atlikėjams jis stabilizuojasi.

5.5.5 Typo



5.13 pav. „Typo“ vartotojų atlikti užduočių kiekiai (agreguotos reikšmės mažesnės nei 101, dešimčių grupėmis, imant vidurkį)



5.14 pav. „Typo“ užduočių kiekis vidutiniškai pagal autorių skaičių viso projekto metu

Kaip matoma „Typo“ projekte vartotojų atliktų užduočių apibendrintame grafike (5.13 pav.), užduotis atliko 72 asmenys. Atlikta užduočių 4219. Projektas vykdomas 3028 dienas.

Projekto iniciatoriai yra du užduočių atlikėjai, jie bendrai įgyvendino 6% užduočių.

Pagrindiniai atlikėjais galime laikyti 9. Jie atliko daugiau nei po 100 užduočių. Daugiausia vienas asmuo atliko 1257 užduotis (jis nebuvo projekto iniciatorius). Projekto iniciatorius atliko 213 užduočių ir pagal atlikimą yra šeštas. Pagrindiniai užduočių įgyvendintojai bendrai atliko 3724 užduotis. Kiti vartotojai atliko bendrai 495 užduotis, t.y. 12% viso projekto.

Vidutinis užduočių atlikimo priklausomybės nuo užduočių atlikėjų skaičiaus grafikas pateiktas 5.14 pav.. Projekto pradžioje šis dydis auga gan smarkiai, tačiau naujiems užduočių atlikėjams jungiantis jis ima kristi, nes naujai prisijungę atlikėjai, lyginant su pagrindiniais atlieka daug mažiau užduočių.

5.5.6 Apibendrinta projektų eigos informacija

5.5 lentelė. Projektų eigos apibendrinta informacija

Matavimas/Projektas	Bootstrap	jQuery	Mono	Spree	Typo	Projektų vidurkiai
Atlikta užduočių	4599	5393	100993	12139	4219	25468,6
Skirtingi užduočių atlikėjai	306	222	519	400	72	303,8
Pradininkų indėlis, %	83	22	10	17	6	27,6
Pagrindiniu atlikėjų kiekis	2	11	12	10	9	8,8
Pagrindinių atlikėjų indėlis, %	83	78	60	79	88	77,6
Ne pagrindinių atlikėjų indėlis, %	17	22	40	21	12	22,4

Apibendrinta projektų eigos informacija pateikta 5.5 lentelėje. Pagrindiniai atlikėjai atlieka didžiąją dalį užduočių visuose projektuose. Šios grupės indėlis kiek mažesnis „Mono“ atveju, kurio vystyme aktyviai dalyvauja didelis kiekis asmenų. Tačiau pagrindinės grupės visuose projektuose yra nedidelės 2-12 atlikėjų, bendrai atliekančių 60-88% viso projekto užduočių.

5.6 Išvados

Pagal atliktą eksperimentinį tyrimą daromos išvados iškeltiems eksperimento tikslams yra tokios:

1. Daugumai (80%) projektų užduočių atlikimo kiekiai pirmaisiais mėnesiais auga pamažu, palaikant bent tolygų vystymą ir prisidedant aktyviais užduočių atlikimo etapais.
2. Skirtingų užduočių atlikėjų per dieną skaičius projektuose kinta nežymiai, išskyrus „Mono“ atvejį. Pastarasis nuo projekto pradžios sugebėjo pritraukti didelį kiekį užduočių atlikėjų.
3. Projekto iniciatoriai dažnai aktyviai dalyvauja projekto eigoje. Jų santykinis indėlis priklauso nuo to kaip projekte dalyvauja naujai prisijungusieji užduočių atlikėjai. Vidutiniškai siekia 27,6%, tačiau imtyje imant projektus su aktyvia bendruomene ir aktyviais iniciatoriais (t.y. „jQuery“, „Mono“, „Spree“), indėlio vidurkis yra 16,3%.
4. Pagrindiniai projekto atlikėjai įgyvendina didžiąją dalį projekto. Vidutiniškai tai yra 77,6% visų užduočių.
5. Atitinkamai nepagrindiniai projekto atlikėjai vidutiniškai įgyvendina 22,4%. Tai yra nemaža projekto dalis surenkama iš mažiau aktyvių atlikėjų.
6. Augant atlikėjų skaičiui, vieno užduoties įgyvendintojo vidutiniškai atliekamas užduočių kiekis priklauso nuo prisijungiančiųjų bei pagrindinių vykdytojų indėlių. Projektui pasiekus kažkokią jo kritinę apimtį (kiekvienam projektui individualiai), Vidutiniškai atlikėjo įgyvendinamų užduočių kiekis mažėja. Tačiau šios kritinės apimties pasiekimas gali siekti keletą metų projekto vystymo.

6 IŠVADOS

Bendruomeninių paslaugų panaudojimas programinės įrangos projektų vystymui yra galimas. Siekiant rezultato, reikalinga suburti aktyvią bendruomenę, kuri galėtų vykdyti reikiamas užduotis. Taip pat svarbu suvaldyti šią bendruomenę ir įmonės veiklą.

Tam tinkama projektų valdymo sistema su integruotomis bendruomeninėmis paslaugomis. Apjungus šias sritis galima stebėti ir veiklą ir aktyviai valdyti resursus. Pačios bendruomenės valdymas ir subūrimas reikalauja didelių investicijų, todėl parankiau bendradarbiauti su bendruomenines paslaugas teikiančiomis sistemomis pasinaudojant jų teikiamais metodais. Tačiau šiuo metu tokios sistemos dar nėra pakankamai išvystytos, kad patenkinti pilnos integracijos poreikį.

Pasinaudojant bendruomeninėmis paslaugomis galima pilnai įgyvendinti projektą, tačiau reikia suburti įgyvendinimo grupės pagrindą, kuris tai užtikrina. Pagrindą projekto eigoje turėtų sudaryti 9 žmonės. Tam kad projektas būtų vystomas stabiliai, 80% atvejų pakanka dar 22 mažiau aktyvių užduočių atlikėjų. Augant projektui ir jo bendruomenei, ne bendruomenės „šerdis“ indėlis tampa vis žymesnis.

7 LITERATŪRA

- Ašeriškis, D., & Tamošaitis, J. (2012) Kitokia programinės įrangos įmonė. Mag&Doc IT2012: tarp universitetinės konferencijos pranešimų medžiaga, 49-52.
- Begel, A., Bosch, J., & Storey, M. A. (2013). Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. *Software, IEEE*, 30(1), 52-66.
- Brabham, D. C. (2008). Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, 14(1), 75-90.
- Choi, J., Choi, J., Moon, J. Y., Hahn, J., & Kim, J. (2013, Vasaris). Herding in open source software development: an exploratory study. In *Proceedings of the 2013 conference on Computer supported cooperative work companion* (pp. 129-134). ACM.
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, Vasaris). Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.
- Ebner, W., Leimeister, M., Bretschneider, U., & Krcmar, H. (2008, January). Leveraging the wisdom of crowds: designing an IT-supported ideas competition for an ERP software company. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* (pp. 417-417). IEEE.
- Fried, D. (2010). Crowdsourcing in the Software Development Industry. *Nexus of Entrepreneurship and Technology Initiative Fall*.
- Hetmank L. (2013) Components and Functions of Crowdsourcing Systems – A Systematic Literature Review. Žiūrėta 2013-02-12, prieiga internete: <http://www.wi2013.de/proceedings/WI2013%20-%20Track%201%20-%20Hetmank.pdf>
- Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2011, June). Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on* (pp. 316-321). IEEE.
- Hoßfeld, T., Hirth, M., & Tran-Gia, P. (2011, September). Modeling of crowdsourcing platforms and granularity of work organization in future internet. In *Proceedings of the 23rd International Teletraffic Congress* (pp. 142-149). ITCP.
- Howe, J. (2006). Crowdsourcing: Crowdsourcing: A Definition. Žiūrėta 2013-01-23, prieiga internete: http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html
- Lange, R., & Lange, X. (2012, March). Quality Control in Crowdsourcing: An Objective Measurement Approach to Identifying and Correcting Rater Effects in the Social Evaluation of Products and Services. In *AAAI Spring Symposium, Stanford University*.
- Majumder, A., Datta, S., & Naidu, K. V. M. (2012, Rugsjūtis). Capacitated team formation problem on social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1005-1013). ACM.
- Mcclowry S. (2007) Architecture Patterns for Data Synchronisation, [žiūrėta 2013-02-12], prieiga internete: http://mike2.openmethodology.org/wiki/Architecture_Patterns_for_Data_Synchronisation
- Millard, W. B. (2011). The Wisdom of Crowds, the Madness of Crowds: Rethinking Peer Review in the Web Era. *Annals of Emergency Medicine*, 57(1), A13-A20.
- Mojombo (2013). GitHub: Five years [žiūrėta 2013-01-20], prieiga internete: <https://github.com/blog/1470-five-years>.
- Psaier, H., Skopik, F., Schall, D., & Dustdar, S. (2011, August). Resource and agreement management in dynamic crowdcomputing environments. In *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International* (pp. 193-202). IEEE.

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, (1), 93-107.

Takhteyev, Y., & Hilts, A. (2010). Investigating the geography of open source software through GitHub, [žiūrēta 2013-02-15], prieiga internete: <http://takhteyev.org/papers/Takhteyev-Hilts-2010.pdf>

Thung, F., LO, D., & JIANG, L. (2013). Network Structure of Social Coding in GitHub. 17th European Conference on Software Maintenance and Reengineering (CSMR).

Tsay, J. T., Dabbish, L., & Herbsleb, J. (2012, Vasaris). Social media and success in open source projects. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion* (pp. 223-226). ACM.

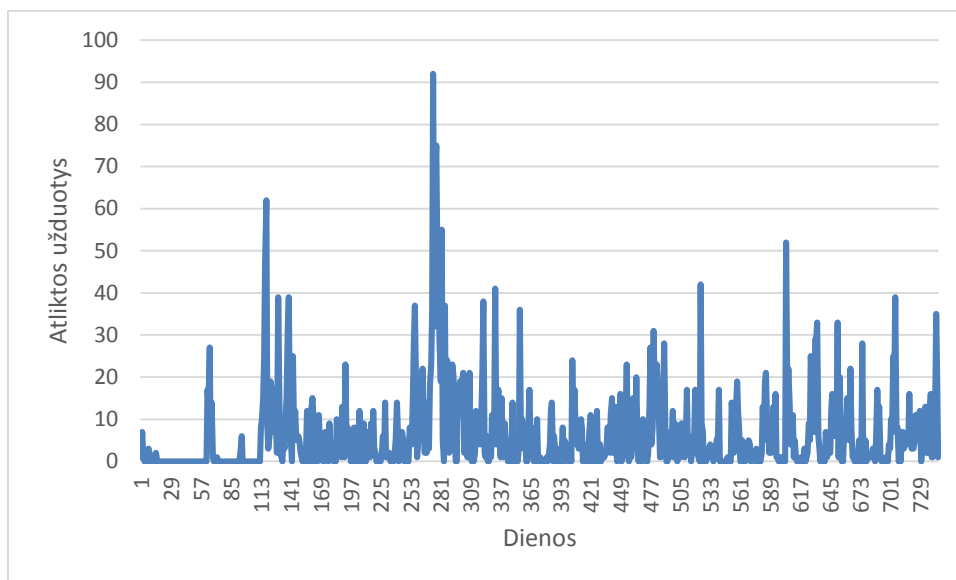
Vasa, R. (2010). Growth and change dynamics in open source software systems. *Faculty of Information and Communication Technologies*, 254.

Weber, N. M. (2012). Combined methods, thick descriptions: Languages of collaboration on Github. *Proceedings of the American Society for Information Science and Technology*, 49(1), 1-4.

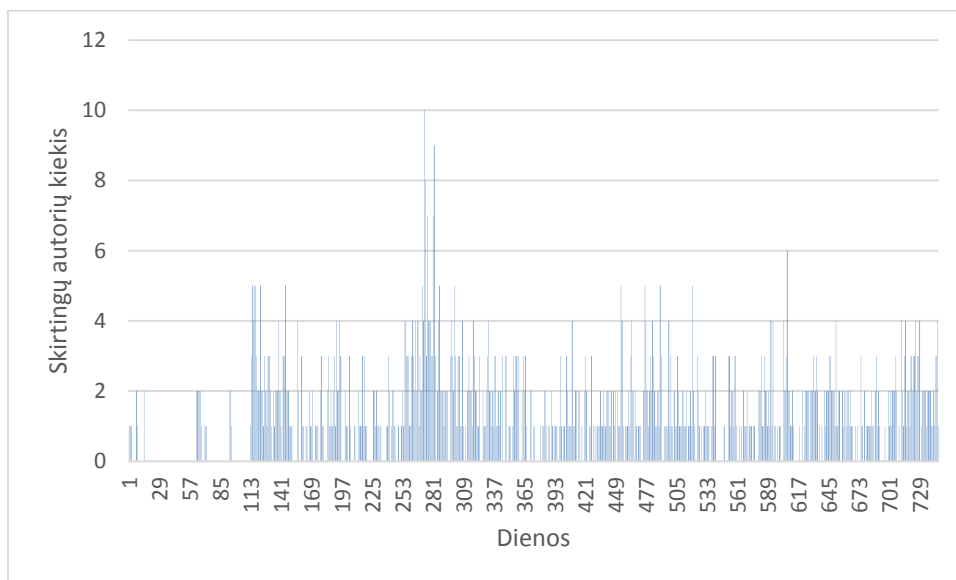
Wolfson, S. M., & Lease, M. (2011). Look before you leap: Legal pitfalls of crowdsourcing. *Proceedings of the American Society for Information Science and Technology*, 48(1), 1-10.

8 PRIEDAI

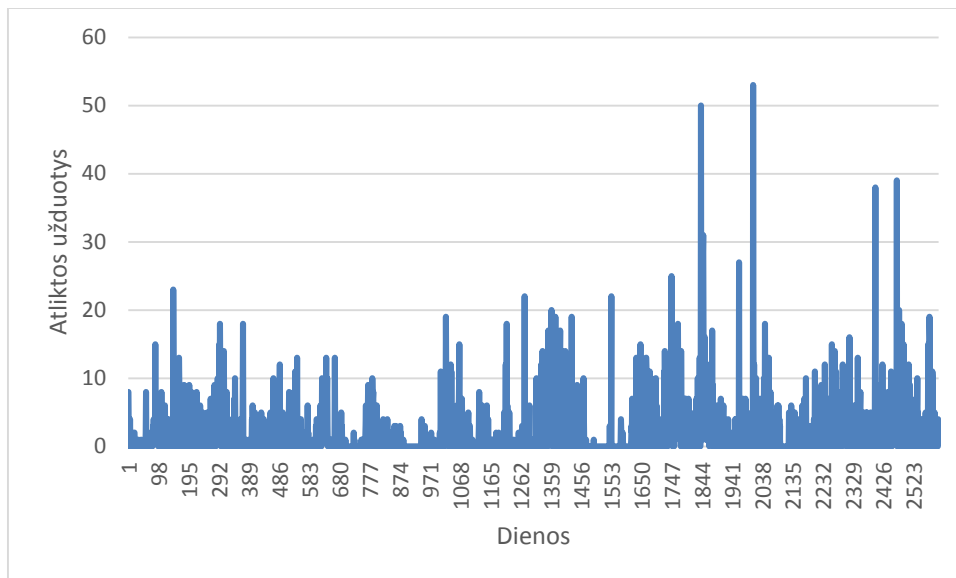
8.1 Užduočių atlikimo grafikai



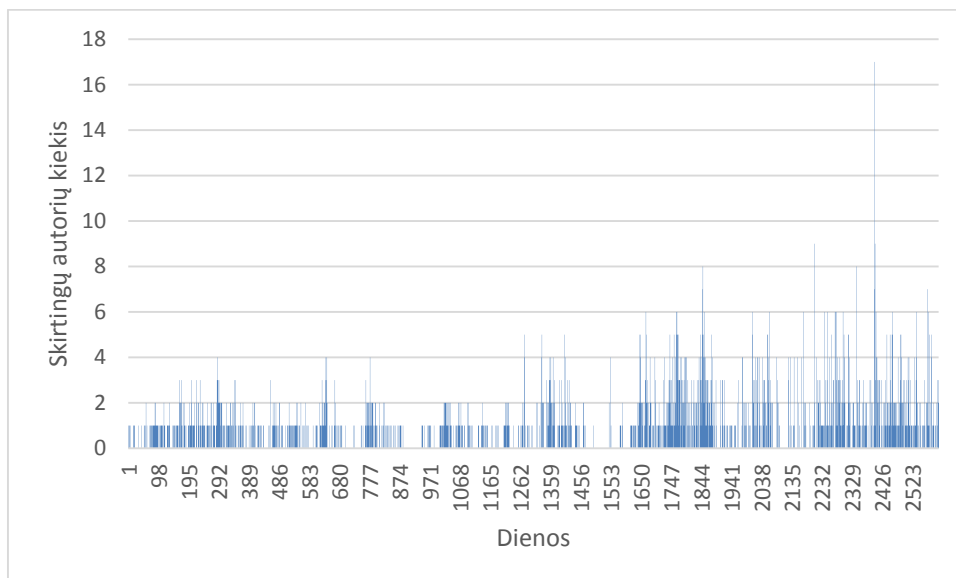
8.1 pav. „Bootstrap“ užduočių atlikimas (dienos intervalais) viso projekto metu



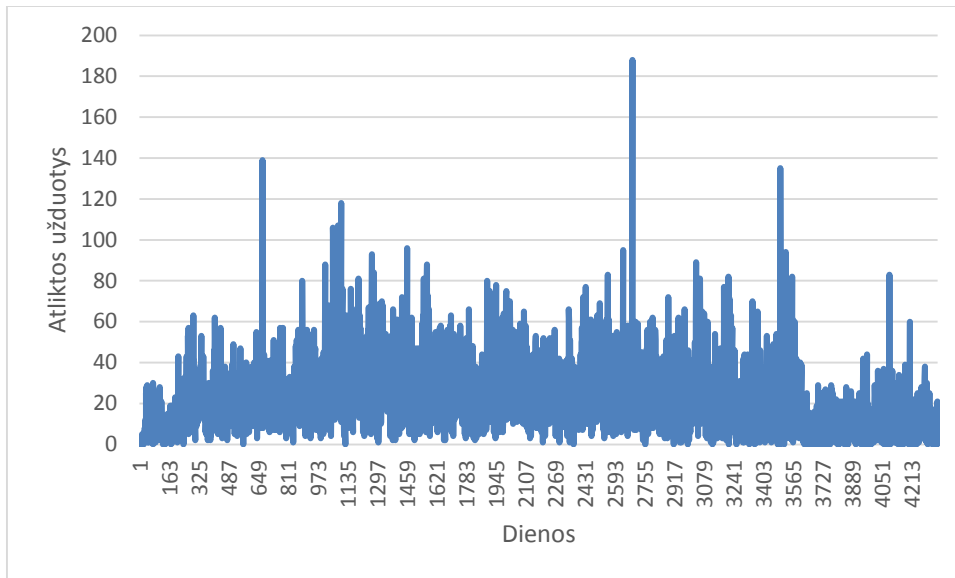
8.2 pav. „Bootstrap“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu



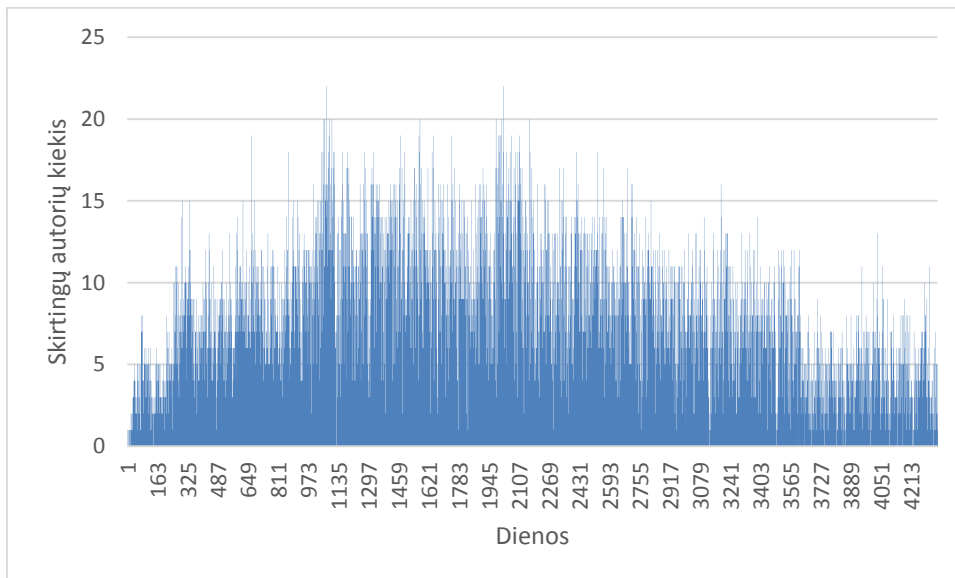
8.3 pav. „jQuery“ užduočių atlikimas (dienos intervalais) viso projekto metu



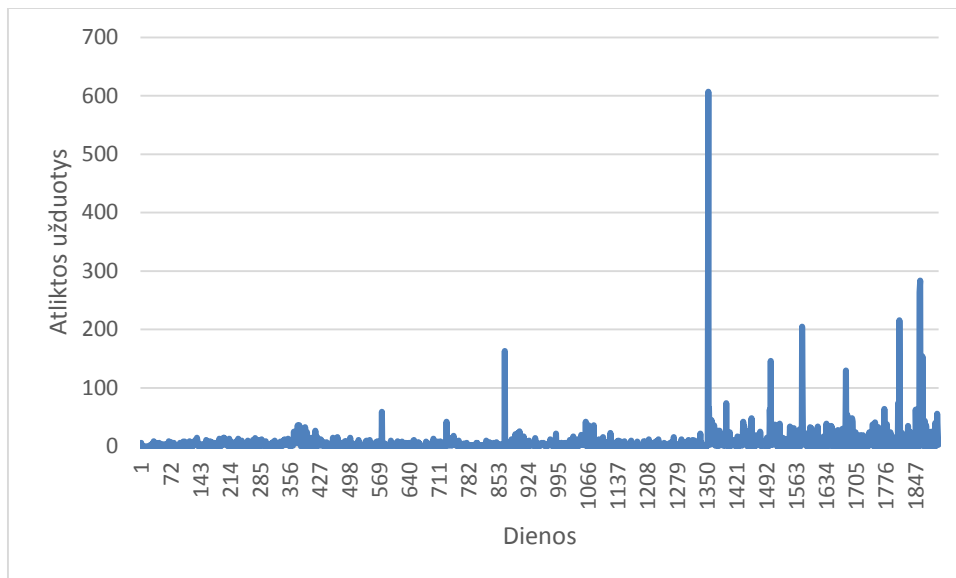
8.4 pav. „jQuery“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu



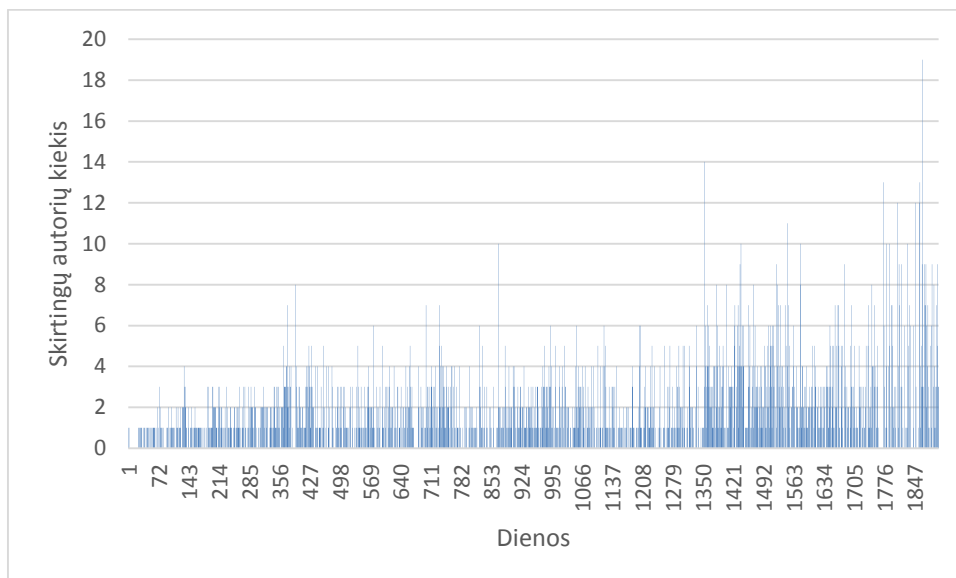
8.5 pav. „Mono“ užduočių atlikimas (dienos intervalais) viso projekto metu



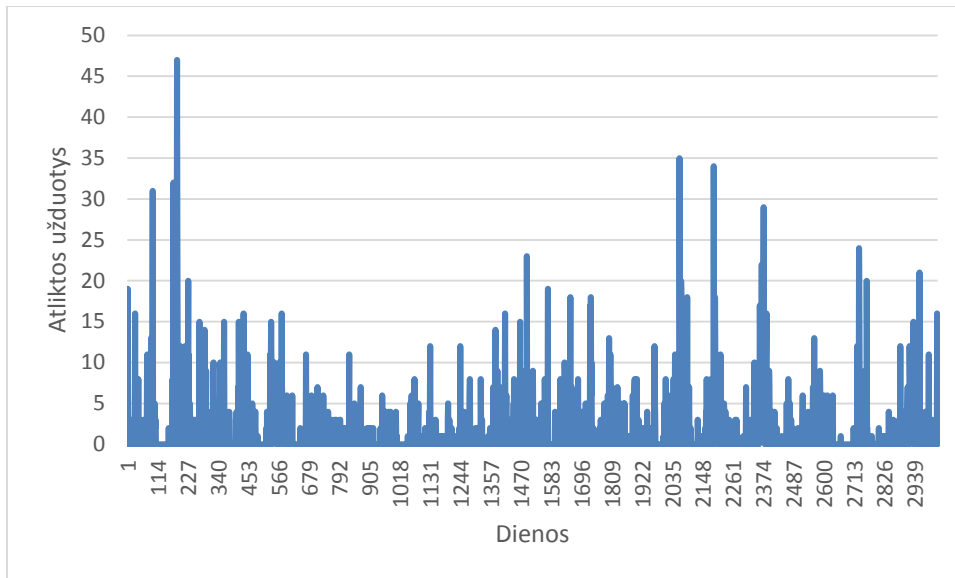
8.6 pav. „Mono“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu



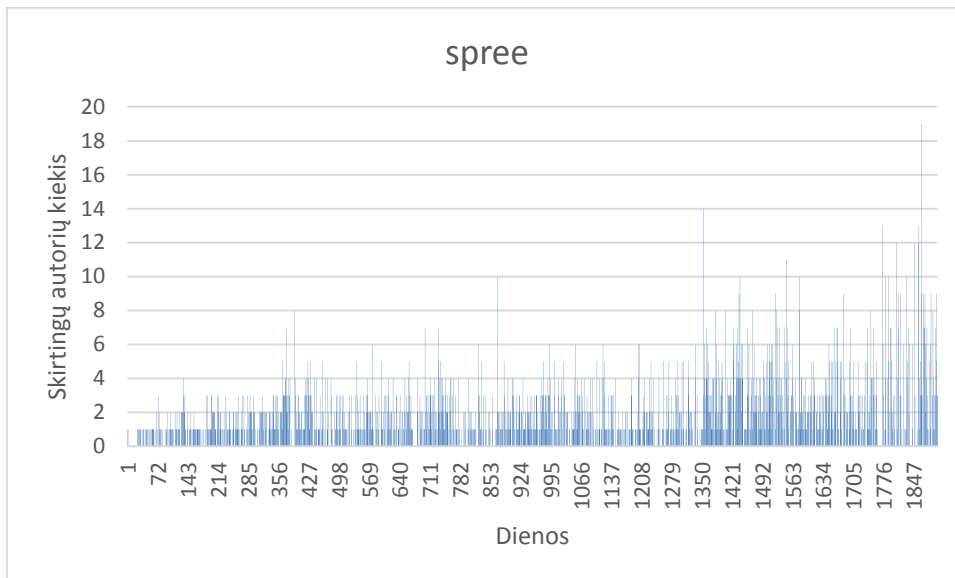
8.7 pav. „Spree“ užduočių atlikimas (dienos intervalais) viso projekto metu



8.8 pav. „Spree“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu



8.9 pav. „Typo“ užduočių atlikimas (dienos intervalais) viso projekto metu



8.10 pav. „Typo“ skirtingų užduočių autorių kiekiai (dienos intervalais) viso projekto metu

8.2 „Kitokia programinės įrangos įmonė“

Straipsnis pristatytas magistrantų ir doktorantų konferencijoje 2012m.

KITOKIA PROGRAMINĖS ĮRANGOS ĮMONĖ

Darius Ašeriškis¹, Justas Tamošaitis²

*Kauno technologijos universitetas, Programų inžinerijos katedra, Studentų 50-406, Kaunas, Lietuva,
daraser@gmail.com¹, justtamo@gmail.com²*

Santrauka. Plintant Web 2.0 technologijoms, jau siekiama ir kitas programinės įrangos sritis perkelti į naują lygį. Viena iš tokių sričių yra programinės įrangos projektų valdymas. Čia apjungus bendruomenines paslaugas, žaidimizaciją, Web 2.0 technologijas ir socialinių verslo modelių idėjas gaunama kitokios programinės įrangos įmonės projektų valdymo sistema. Straipsnyje apžvelgiama tokia sistema ir ją naudoti galinti įmonė.

Raktiniai žodžiai: Bendruomeninės paslaugas, projektų valdymas, Web 2.0, žaidimizacija.

1 IŽANGA

Tobulėjant technologijoms ir metodams, jas panaudojant tobulinami ir įmonių darbo procesai. Pokyčius daryti verčia pati įmonės aplinka, kintanti keičiantis darbuotojų kartoms. Ryškėja nauji jaunųjų darbuotojų bei vartotojų, kurie yra socialiai aktyvūs internetinėje erdvėje, poreikiai ir vertybės. Darbo rinkoje atsiranda vis daugiau naujos kartos atstovų, kurie turi didesnių poreikių nei pinigai, jiems reikia įmonės permatomumo ir saviraiškos, jie nori būti matomi ir siekia būti pripažinti. Anksčiau taikytos skatinimo priemonės nebetinka vis didesnei daliai darbų. Atsižvelgiant į tai, įmonės pereina iš standartinių veiklos modelių į naujus, įtraukiant naujų technologijų galimybes.

Pasinaudojant naujomis technologijomis, pereinama prie internete veikiančių įmonės sistemų. Tai net ir mažai įmonei leidžia pasiekti paprastesnį darbą su darbuotojais iš skirtingų geografinių vietų. Straipsnyje nagrinėjamas vienos iš programinės įrangos įmonės veiklos sričių – projektų valdymo – patobulinimas naudojant Web 2.0 technologijas.

2 WEB 2.0 TECHNOLOGIJOMIS GRĮSTŲ MODELIŲ APŽVALGA

Turint nepastovų darbų srautą, t. y. situacija svyruoja nuo daugybės užsakymų iki prastovų, resursų paskirstymas gali būti problema arba netgi sužlugdyti įmonę. Būtent taip ir nutiko prasidėjus 2008 m. pasaulinei ekonomikos krizei, kai darbų srautai smarkiai sumažėjo. Todėl reikia keisti įmonės struktūrą taip, kad ji taptų lankstesne. Įprastu atveju reikia išlaikyti visus darbo resursus, nes naujų pritraukimas gali užtrukti per ilgą laiką ir netenkinti naujų projektų reikalavimų, o nedidelei projekto daliai reikėti specifinių specialisto įgūdžių. Įmonės siekia optimizuoti resursų panaudojimą, todėl perteklinių darbo resursų išlaikymas tampa nepriimtinas.

Darbo resursų pritraukimui reikiamu laiku gali būti pasinaudojama bendruomeninėmis paslaugomis (angl. k. „crowdsourcing“) – užduočių, kurias tradiciškai atlikdavo konkretus subjektas (paprastai darbuotojas), perdavimas neapibrėžtai, dažniausiai didelei grupei žmonių („miniai“). Čia svarbu, kad kvietimas atlikti užduotį būtų atviro kvietimo pagrindu, o ne konkretiems subjektams [2]. Bendruomeninės paslaugos taikomos įvairiose srityse (IT – dizainai, algoritmų kūrimas, testavimas; ekologija – eko-idėjų generavimas ir išdirbimas; gamyba- marškinėlių dizainas; turinio generavimas – trumpų straipsnių rašymas, vertimai; medicinoje – duomenų perrinkimas; ir kt.), kur jos panaudojamos nuo idėjų generavimo iki konkretaus produkto sukūrimo. Pasinaudojant bendruomeninėmis paslaugomis galima įgyvendinti nebūtinai visą projektą, bet ir vien tik jo dalį.

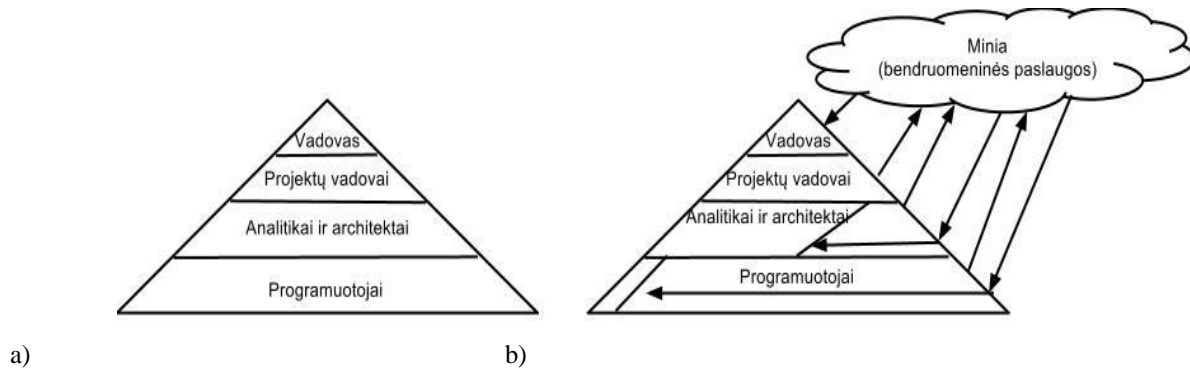
Yra daug sėkmingų bendruomeninių paslaugų panaudojimo pavyzdžių, tokių kaip: Netflix Prize – kvietimas pagerinti filmų pasiūlymus vartotojams, pagal turimą jų reitingų informaciją, algoritmą [8]. Threadless – svetainė, kurioje marškinėlių dizainus kuria visi norintys, o bendruomenė atrenka geriausius variantus [12].

Vienas iš metodų, skatinantis spręsti problemas ir labiau įtraukti darbuotojus į įmonės veiklą yra žaidimizacija (angl. k. „gamification“) – žaidimo elementų pritaikymas ne žaidimų aplinkoje, siekiant padidinti vartotojų pasitenkinimą, įsitraukimą ar pagerinti vartotojų potyrius besinaudojant sistema [5]. Žaidimizacijos pavyzdžiai: FourSquare – tinklo nariai atvykę į pvz., kavinę prisijungia ir įregistruoja savo vietą. Už tai sistemoje gaunami taškai, pagal kuriuos skaičiuojamas jų lygis [6]. RecycleBank – įmonė, įdiegusi socialiai-orientuotą verslo modelį, leidžianti pritraukti gamtos saugos idėjomis besidominčius klientus [6]. Tačiau reikia atsargiai pasirinkti žaidimizacijos elementus ir jų įgyvendinimą, pvz., vertinant tik ilgalaikį progresą, naujokams nesudarant sąlygų pasivyti senbuvius, gausime priešingą rezultatą norimam – darbuotojai nebus motyvuoti atlikti užduotis, nes negalės „pasižymėti žaidime“.

Viena iš verslo sričių, leidžianti valdyti darbo resursus ir kurti užduočių aplinką, yra projektų valdymas, kurio viena iš dalių yra projektų valdymo sistema (PVS). Mūsų magistro darbo tema yra bendruomeninės PVS sukūrimas ir tyrimas. Kurdami sistemą apjungiamo bendruomeninių paslaugų ir žaidim�acijos koncepcijas su Web 2.0 technologijomis ir socialinių verslo modelių idėjomis. Tarp keliamų uždavinių yra tinkamiausių koncepcijų elementų atrinkimas ir sklandžios jų tarpusavio sąveikos užtikrinimas.

3 KITOKIOS PROGRAMINĖS ĮRANGOS ĮMONĖS ORGANIZACINIS MODELIS

Tipinės programinę įrangą kuriančios įmonės organizacinis modelis pateikiamas 1a pav. Jo struktūra yra hierarchinė piramidė: žemesnio lygio darbuotojų yra daugiau nei aukštesniojo. Projekto įgyvendinimo darbai vykdomi įmonėje. Iš įmonės išorės gaunama tik informacija ir konsultacijos. Įmonei sunku prisitaikyti prie kintančio apkrautumo, nes reiktų atleisti darbuotojus arba priimnėti naujus. Toks procesas užtrunka nemažai laiko ir kainuoja resursų. Pagal tradicinės įmonės organizacinį modelį išorinių darbuotojų panaudojimas sunkiai realizuojamas. Su tokiais darbuotojais vis tiek sudaromos ilgalaikės sutartys, ir juos galima laikyti įprastais įmonės darbuotojais. Apatinio lygio darbuotojai gauna tik labai ribotą informaciją apie atliekamas veiklas. Taip pat turi mažai informacijos, leidžiančios save palyginti su kitais darbuotojais.



1 pav. Įmonės resursų struktūra: a) įprastinė, b) siūlomo modelio

Siūlomas naujas įmonės organizacinis modelis pateikiamas 1b pav. 1b pav. rodyklės iš minios reiškia resursų paėmimą, o rodyklės į minią – resursų perdavimą kitiems. Resursai gali būti užduotys ar darbuotojai. Rodyklės piramidės viduje rodo galimybę darbuotojų kiekį keisti priklausomai nuo darbo, t.y. dalimi darbuotojų galima laikyti minią. Minios nariu gali būti tiek pavienis asmuo, tiek ir kita įmonė.

Mūsų siūlomas įmonės modelis orientuotas tiek į vartotojus, tiek ir į darbuotojus. Organizacinį modelį vadybiniu požiūriu apibrėšime pasinaudojant tokiomis savybėmis:

- 1) Bendravimas vyksta tiek įmonės viduje, tiek ir su vartotojais. Poreikiai yra žinomi, dalinamasi turima informacija, bendrai siekiama rasti tinkamiausią variantą. Bendravimo poreikis matomas iš socialinių tinklų augimo. Šis poreikis pernešamas į darbo vietą siekiant konstruktyvaus ir įmonei naudingo rezultato.
- 2) Motyvacija ir įtraukimas – norint pasiekti geriausią darbo kokybę, darbuotojai turi būti motyvuoti ir įsitraukę į darbo veiklas. Todėl turi būti siekiama integruoti motyvacijos skatinimo sistemą tiek į naudojamus įrankius, tiek ir į pačią darbo aplinką.
- 3) Skaidrumas – įmonės veikla turi būti suprantama darbuotojams. Administracijos priimami pagrįsti sprendimai, atsižvelgiant į darbuotojų nuomones ir pasiūlymus. Pateikiamos pasirinkto varianto priežastys.
- 4) Resursų organizavimas – galima greitai perskirstyti, papildyti ar sumažinti turimus resursus (pasinaudojant bendruomeninėmis paslaugomis).

4 TECHNOLOGINIS SPRENDIMAS

Naujos kartos programinės įrangos kūrimo įmonė turėtų turėti programinę įrangą, kuri įmonei leistų: 1) valdyti savo veiklos procesus ir projektus; 2) užtikrinti sąvokų vienareikšmį suvokimą organizacijoje; 3) gebėti kardinaliai padidinti savo pajėgumus atsiradus poreikiui; 4) įtraukti darbų srautą esant poreikiui; 5) projektų vadovas turi galėti operatyviai valdyti organizaciją; darbuotojai sistema turi naudotis ne prievarta, o savo noru; 6) sistema turi padėti motyvuoti darbuotojus; 7) padėti valdyti visą projektų informaciją; 8) leisti kaupti žinias; 9) padidinti skaidrumą organizacijoje.

Pastaroji sudaryta iš dviejų esminių dalių: PVS – galima platus suvokimas remiantis skirtingais aspektais, pvz., iš sistemų mokslo perspektyvos [4]; bendruomeninių darbo paslaugų valdymo sistemos [11], [1], [3].

PVS tenka didžioji dalis reikalavimų. Mūsų sistemos pagrindiniai elementai: klientai, vartotojai (darbuotojai), projektai, užduotys, darbai, PĮ modelis ir jo elementai, veiklos, žinių vienetai. Šios informacijos esybės apibrėžia pagrindinius

informacijos vienetus, su kuriais sistema veiks. Pastebėsime, kad užduotys skirstomos į darbus. Šis sprendimas argumentuojamas tuo, kad asmeninėje praktikoje matome, kad užduotis nepakankamai įpareigoja projektų vadovus įdėti pakankamai darbo apmąstant ir įvertinant priskiriamą darbą. Kai projektų vadovui reikia įvardinti konkrečius darbus susiejamus su konkrečiu modelio elementu, jis priverstas įdėti daugiau pastangų ir taip gautas užduoties aprašas neša daugiau informacijos bei konkretumo.

Mūsų kuriama PVS iš esmės remsis žaidimo mechanika, taip pat bus surišta su žaidimo aspektais [6]. Pastarojo elemento reikia norint patenkinti tokius poreikius: 1) darbuotojai sistema turi naudotis ne prievarta, o savo noru; 2) sistema turi padėti motyvuoti darbuotojus.

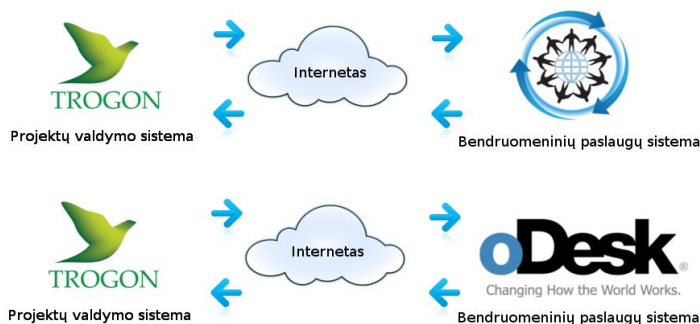
Šiems poreikiams įgyvendinti projektų valdymo sistemą papildysime tokiais elementais: 1) įvesime užduočių susiejimą su taškų sistema, t.y. projektų vadovas sukurdamas užduotis turės įvertinti užduoties laiką ir sudėtingumą, kurie bus naudojami taškų skaičiavimui; 2) darbuotojas atlikęs užduotį gaus taškų skaičių, kuris priklausys nuo užduoties bazinės vertės ir atlikimo greičio bei kokybės.

Informacijos valdymas bus atliekamas panaudojus įprastinę formą, kuri susideda iš lentelių, įvedimo, bei keitimo formų ir atvaizdavimo. Išskirsime nemažai peržiūros variantų, kad tiek projektų vadovas, tiek darbuotojai galėtų matyti informaciją greitai ir patogioje formoje. Užduočių planavimui naudosime Ganto grafikus, kuriuose bus matoma praeitos, einamos ir būsimos savaitės užduočių informacija. Projektų vadovas galės matyti visos savaitės darbų planą ir bus pašalintos problemos susijusios su nesuplanuota veikla. Darbuotojas matys savo savaitės planą, o projektų vadovui bus prieinama kiekvieno darbuotojo informacija. Sistema pasitelkusi dabartines bendruomeninių paslaugų teorijas ir jas pritaikius praktikoje, leis turėti dar vieną resursą – tai yra dalinai anoniminę bendruomenę, kuri leis išplėsti įmonės galimybes, tačiau tokiai bendruomenei reikia tinkamos PVS, kuri atliks aukščiau paminėtas funkcijas. Toks sprendimas turėtų būti ypač naudingas nedidelėms įmonėms, kurioms dažnai gali pritrūkti darbo, todėl jos galės greitai įtraukti darbus iš kito šaltinio, kad užpildytu darbų stygių arba priešingai formuoti visą savo veiklos modelį orientuotą į darbų įtraukimą. Kitas sujungimo aspektas yra darbo pajėgumų padidinimas, kuris leis įmonei kombinuoti tokius aspektus: 1) Organizacijos struktūros keitimą – teisingai vadovaujant ir valdant procesus įmanoma pasiekti, kad įmonė sumažintų arba panaikintų savo žemo lygio programuotojų skaičių ir tik laikytų darbuotojus, kurie atliktų pagrindę produktų surinkimo darbus. Kadangi bendruomenė turi didelį skaičių potencialių darbuotojų, tai įmonė gali imti vykdyti kelis projektus, o jų komponentus kurti gali patikėti bendruomenei, sau pasilikdama tik komponentų apjungimo ir sistemos testavimo funkcijas; 2) Darbų pajėgumo padidinimas – esant reikalui įmonė gali imti vykdyti kelis vidutinius projektus arba vieną stambų, kurio paprastu atveju nepajėgtų atlikti nesamdydama naujų darbuotojų. Šiuo atveju dalį darbų įmonė perduotų atlikti bendruomenei; 3) Darbų kainos mažinimas – įmonė vykdanči tradicinę veiklą, gali būti linkusi sumažinti darbų kaštus perduodama juos atlikti mažiau kvalifikuotiems arba mažiau apmokamiems darbuotojams ir taip padidinti pelną arba sumažinti konkurse siūlomą kainą projektui atlikti.

Toks veiklos modelis atrodo perspektyvus, tačiau bendruomeninių darbo paslaugų sistema turi daug naujų funkcinių reikalavimų [3], tokių kaip: minios valdymas ir kvalifikavimas, darbų valdymas, darbų vykdymo sekimas, atsiskaitymų valdymas, minios kvalifikacijos įvertinimas, atsiliepimų sistemos diegimas, skundų ir ginčų valdymas, bei kiti. Taip pat su bendruomeninių darbo paslaugų valdymo sistema kyla nauji nefunkciniai reikalavimai bei galimos problemos, tokios kaip: sistemos vientisumo išlaikymas (užtikrinti, kad atsiskaitymai vyktų tik sistemos viduje); minios pritraukimas ir investicijos tam tikslui pasiekti; darbų pritraukimas ir investicijos tam tikslui pasiekti; vardo išlaikymas; garantijų teikimas. Šie aspektai atgraso daugumą įmonių nuo tokio modelio vykdymo, tačiau yra galimas efektyvus sprendimas.

5 EFEKTYVUS SPRENDIMAS

Mes siūlome papildyti projektų valdymo sistemą galimybe prisijungti prie jau esamų atvirų bendruomeninių darbo paslaugų valdymo sistemų. 2 pav. pavaizduotas šio pakeitimo skirtumas. Šiuo veiklos atveju mes nusimetame visą bendruomeninių darbo paslaugų valdymo sistemos našą nuo savo pečių. Tai leidžia žymiai sumažinti sistemos kūrimo kaštus ir reikalingas investicijas darbams ir bendruomenei pritraukti, nes jei nėra darbuotojų nėra ir darbų, ir atvirkščiai.



2 pav. PVS sąveika su bendruomeninių paslaugų sistema

Mes naudosime oDesk [10] siūlomą API (*Application Programming Interface*), kuris mums leis pilnai integruoti šią sistemą į mūsų PVS. oDesk API yra gerai dokumentuota ir integracija nesukels didelių problemų. Bendru atveju sistema galėtų leisti įtraukti visą aibę bendruomeninių darbo paslaugų sistemų [1], pvz., 99designs – dizaino darbų atlikimui; oDesk – darbų atlikimui; Amazon Mechanical Turk – testavimui, turinio administravimui ir kitiems smulkiems darbams; MicroTasks – mažoms užduotims atlikti.

6 IŠVADOS

Pasiūlėme įmonės organizacinį modelį bei PVS variantą panaudojantį bendruomeninių paslaugų, Web 2.0 technologijų, socialiai-orientuotų verslo modelių ir žaidimizacijos metodus įmonės veiklos našumo didinimui.

Numatomi siūlomo metodo pranašumai: greitas reagavimas į darbo kiekio pokyčius; greitesnė ir/ar mažiau resursų reikalaujanti sistemų realizacija; didesnis skaidrumas įmonėje; didesnė darbuotojų motyvacija; nereikalingi resursai „minios“ kūrimui, valdymui ir jos kokybės užtikrinimui – tai atlieka trečioji šalis.

Numatomi siūlomo metodo trūkumai: reikalingas prisitaikymas prie kitų sistemų, nes „minia“ valdoma trečiųjų šalių; reikalingi skaidrumo norintys vartotojai; reikia parinkti tinkamus žaidimizacijos elementus.

Būsimi darbai: žaidimizacijos elementų gilesnė analizė ir išskyrimas tinkamų siūlomam metodui; sistemos pagal pateiktą modelį realizacija, jos tyrimas bei analizė.

Literatūros sąrašas

- [1] **Peng L., Zhang M.** An Empirical Study of Social Capital in Participation in Online Crowdsourcing. *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference, 7-9 Nov. 2010, 1 p.*
- [2] **Brabham D.** Crowdsourcing as a Model for Problem Solving : An Introduction and Cases. *The International Journal of Research into New Media Technologies, 2008, Vol. 14(1): 75–90.*
- [3] **Vukovic M.** Crowdsourcing for Enterprises Services - I, *2009 World Conference, 6-10 July 2009 , 686 - 692 p.*
- [4] **Karayaz G., Keating C.B., Henrie M.** Designing Project Management Systems. *System Sciences (HICSS), 2011 44th Hawaii International Conference, 4-7 Jan. 2011, 1 - 10 p.*
- [5] **Deterding S., Khaled R., Nacke L., Dixon D.** Gamification: Toward a Definition. *Gamification Workshop, CHI 2011.*
- [6] **Yongwen X.** Literature Review on Web Application Gamification and Analytics. *CSDL Technical Report 11-05.*
- [7] **Wolfso S. M., Lease M.** Look Before You Leap: Legal Pitfalls of Crowdsourcing. *Proceedings of the American Society for Information Science and Technology, Volume 48, Issue 1, 2011, 1-10p.*
- [8] NetflixPrize [Žiūrėta 2011-11-09], *prieiga internete <www.netflixprize.com>*
- [9] **Nikkila S., Linn S., Sundaram H., Kelliher A.** Playing in Taskville: Designing a Social Game for the Workplace. *Gamification Workshop, CHI 2011.*
- [10] oDesk [Žiūrėta 2012-02-29], *prieiga internete <www.odesk.com>*
- [11] **Howe J.** *The Rise of Crowdsourcing, Wired Magazine, Nr. 14.0,- June 2006.*
- [12] Threadless [Žiūrėta 2012-02-16], *prieiga internete <<http://www.forbes.com/2010/01/06/threadless-t-shirt-community-crowdsourcing-cmo-network-threadless.html>>*

Different Software Company

As Web 2.0 technologies are spreading, other software areas need to be pushed to a new level. One of such areas is software project management. By connecting concepts such as crowdsourcing, gamification, Web 2.0 and ideas of social-oriented business models, we get a project management system of Different Software Company. Here we review such system and a business model that can take advantage of it.

8.3.,Projektų valdymo sistemos išplėtimas bendruomeninėmis paslaugomis“

Paruoštas straipsnis.

Projektų valdymo sistemos išplėtimas bendruomeninėmis paslaugomis

Justas Tamošaitis

Programų inžinerijos katedra
Kauno technologijos universitetas
Studentų 50-406, Kaunas, Lietuva
justtamo@gmail.com

Santrauka— Panaudojant Web 2.0 technologijas, siekiama tobulinti projektų valdymo procesus. Straipsnyje apžvelgiama bendruomeninių paslaugų panaudojimas ir galima projektų valdymo sistemos išplėtimą bendruomeninėmis paslaugomis variantą įgyvendinanti sistema.

Raktiniai žodžiai— Bendruomeninės paslaugos, projektų valdymas, Web 2.0

1 ĮVADAS

Kintant darbo rinkai tobulinami ir įmonių naudojami darbo procesai. Ieškoma būdų kaip pagerinti savo veiklą. Tam panaudojama kelių metodų apjungimas ar procesų išplėtimas/papildymas naujais elementais ar idėjomis.

Programinės įrangos kūrimo sektoriuje stengiamasi pasinaudoti Web 2.0 technologijų ir idėjų teikiamais privalumais. Šie privalumai sukuria galimybę ir mažoms įmonėms pilnai dalyvauti rinkoje. Straipsnyje nagrinėjama projektų valdymo sistemos išplėtimo bendruomeninėmis paslaugomis idėja ir jos galimas įgyvendinimo variantas.

2 BENDRUOMENINIŲ IŠTEKLIŲ PANAUDOJIMO APŽVALGA

Žmonės jau nuo seno jungėsi į bendruomenes, tačiau per pastarąjį dešimtmetį populiarėjant internetui ir jam tampant neatskiriama žmonių gyvenimo dalimi internetu paplito naujas – internetinių bendruomenių – virusas. Šiandien žmonės dažnai priklauso kelioms virtualioms, internetinėms bendruomenėms, kurios gali būti socialinės (skaitmeninėje erdvėje jungiančios draugus ir pažįstamus), ideologinės (jungiančios

panašių idėjų žmones) ir tikslinės (jungiančios žmones, turinčius panašius tikslus).

Suburtos bendruomenės turi daug potencialo ir gali būti panaudotos įvairiems tikslams, pavyzdžiui, atlikti užduotims ar kurti turiniui. Bendruomeniniai išteklių gali būti panaudoti tiesiogiai arba būti pasitelkti. Panaudojimo atveju resursai naudojami saviems tikslams, pavyzdžiui, Wikipedia, kuri savo bendruomenės narius išnaudoja savo turiniui rinkti. Pasitelkimas orientuotas į sistemą, kuri leidžia utilizuoti bendruomeninius išteklius reikiamam tikslui pasiekti, pavyzdžiui, Amazon Mechanical Turk. Bendruomeninių išteklių panaudojimas antruoju atveju vadinamas bendruomeninėmis paslaugomis. Šios paslaugos tiekiamos trečioms šalims. Pastarąjį terminą pirmą kartą paminėjo Jeff Howe [6].

Kuo toliau, tuo labiau verslo pasaulis bando panaudoti bendruomeninius išteklius. Todėl formuojasi nauji verslo modeliai, kurie leidžia verslo ir bendruomeninių išteklių sąjungą. Šie nauji modeliai leidžia integruoti bendruomenes keliais būdais:

Bendruomenė tampa įmonės dalimi – šiuo atveju bendruomeninės paslaugos sujungiamos su įmonės pagrindine veikla.

Bendruomenei pavedama tik dalis darbų, kuriuos ji turi arba gali atlikti.

Bendruomenė naudojama įmonės resursams balansuoti.

Bendruomenines paslaugas galima panaudoti programinės įrangos kūrimui. Tačiau reikia įvertinti galimas problemas ir jų sprendimo būdus.

2.1 Rezultato kokybės užtikrinimas

Atliktas užduotis reikia įvertinti, prieš apmokant už darbą. Be to užduoties pateikėjas gali neturėti ar nenorėti skirti resursų užduoties atlikimo kokybės patvirtinimui. Taip pat įvertinant užduotį gali kilti vertinimo nesutapimas tarp užduoties pateikėjo ir vykdytojo.

Todėl reikalingas kokybės tikrinimo mechanizmas iš bendruomeninių paslaugų tiekėjo pusės. Rezultato kokybės įvertinimui gali būti panaudojami tokie metodai:

- įmonės atstovas įvertina kokybę;
- bendruomenei pateikiama papildoma užduotis kokybės įvertinimui [1];
- įvertinimą atlieka paslaugos tiekėjas (kokybės vertinimo komanda).

Pateikimo bendruomenės įvertinimui tinkamos tiksliai specifikuotos užduotys. Galima pateikti keliems vertintojams ir pagal daugumos vertinimą nuspręsti. Tačiau vertinimas negali būti pateikiamas bet kokiems vartotojams. Reikalingi gebėjimai vertinant atitikimą pateiktiems reikalavimams. Todėl tokios užduotys gali būti pateiktos tik reikiamą patirtį turintiems. Taip pat dažnai naudingiau, jei rezultato įvertinimą atliktų nesusiję su užduoties pateikėju ir vykdytoju asmenys. Taip išvengiama šališkumo [1], [7].

Norint užtikrinti rezultato kokybę, reikalinga ir tinkama rezultatų reitingavimo sistema – jei vis pateikiami nekokybiški rezultatai, tokiam vartotojui neturi būti leidžiama atlikinėti užduočių. Tačiau vartotojas galimai vieno tipo užduotis atlieka sėkmingai, tik problema su kito tipo užduotimis. Čia vartotojas neteisingai įvertina savo gebėjimus. Todėl būtų naudinga, kad pati sistema tiesiog nebeleistų tokiam vartotojui rinktis atitinkamo tipo užduočių.

Taip pat užduoties pateikėjas gali bandyti neišiamai vertinti užduotis, kad nereikėtų už jas atsiskaityti. Tokius vartotojus taip pat būtina nubausti: nebeteikiant paslaugos arba informuojant prie užduoties apie galimas problemas.

Reitingavimo sistemos viena iš dalių galėtų būti sistemos vartotojų vertinimai (nebūtinai susijusių užduoties atlikimu) [4]. Čia šiuo atveju vertinama informacijos pateikimas, tikslumas. Vartotojams pateikiama informacija akcentuotą, jog

gali reikėti labiau įsigilinti į problemą, jei užsakovas dažnai nepilnai specifikuoja užduotis.

2.2 Užduočių pateikimas kandidatams

Sistema turi pateikti užduotis kandidatams. Čia svarbu rasti kuo geresnį variantą kaip tai atlikti tinkamai vartotojams. pasirinktas sprendimas turi pritraukti maksimalų tinkamų kandidatų skaičių, be reikalo neįkyrėti. Toliau aptariami pagrindiniai galimi užduočių pateikimo kandidatams variantai.

2.2.1 Pateikimas visko visiems.

Šiuo metodu pateikiant užduotis, kandidatai gauna daug perteklinės informacijos, t.y. jie turi papildomai filtruoti užduočių srautą. Tačiau šiuo atveju yra tikimybė, kad užduotį atlikti apsiims kitos srities atstovas. Pastarasis variantas naudingas „nestandartinėms“ užduotims tuo pačiu ir nevisai tikslaus užduoties specifikuojimo atveju, pvz.: telekomunikacijų srities specialistai gali pasiūlyti kitokį duomenų perdavimo variantą, gal ir daug tinkamesnį konkrečiai sistemai, nei medicinos srityje dirbantys, t.y. sujungus sričių žinias galima atrasti naujų sprendimų ar taikymų. Standartinėms užduotims, t.y. problemoms, kurių sprendimai aiškūs, o tiesiog reikalinga juos įgyvendinti, pvz., suasmeninto dizaino pritaikymas aiškiai apibrėžtam komponentui. Tokiu atveju tai geriausiai galės atlikti šios srities specialistas.

2.2.2 Pateikimas pagal pasirinktas dominančias sritis.

Bendru atveju, galimi užduočių atlikėjai pasirenka juos dominančias sritis. Sritis apibrėžimas gali apimti konkrečią programavimo kalbą (C#, Java), metodiką (objektinis, funkcinis programavimas), veiklos sritį (internetiniai sprendimai, matematiniai uždaviniai) ir t.t. Jei platforma yra skirta kažkuriai sričiai, vis tiek yra dalis, kuri nėra apibrėžta. Tokiu atveju skaidymas vykdomas pagal tokias neapibrėžtas sritis. Taip kandidatams tiek paprasčiau atsirinkti dominančias užduotis, tiek galima įgyvendinti informavimo pranešimų apie naujas užduotis sistemą. Tačiau nestandartinių sprendimų ar nevisai teisingo sričių specifikuojimo atveju, prarandami kandidatai. Taip pat kandidatai iš anksto turi pasirinkti dominančias sritis.

2.2.3 Pateikimas pagal atitikimą reikalavimams.

Kandidatai pateikia kuo išsamesnę informaciją apie savo galimybes, juos dominančius užduočių parametrus. Tuo pačiu užsakovai, pateikdami užduotis išsamiai aprašo reikalavimus. Tiek užsakovų tiek ir kandidatų informacija turi būti pateikiama sistemos protokolu [4], [5]. Pateikiami reikalavimai apima tokius aspektus kaip: veiklos sritis, numatomas įvykdymo laikas, užsakymo kainos režiai (užsakovai ir kandidatai pateikia juos tenkinančius), nuobaudos už vėlavimus, reikalingos patirtys, atliktų užduočių reitingas ir kt. [5]. Šiuo atveju paprasčiau atsirinkti dominančias užduotis. Tačiau reikia teisingai įvertinti parametrus tiek užduoties pateikėjams tiek ir kandidatams. Realizacija sudėtingesnė, nei kitų minėtų būdų. Reikalingas informacijos aprašymo protokolas sistemoje.

2.3 Užduoties atlikėjo parinkimas

Kiekvienos sistemos tikslas yra gauti teisingą rezultatą. Tam reikia parinkti tinkamus kandidatus užduočiai atlikti. Toliau aptariami keli galimi kandidatų parinkimo variantai.

2.3.1 Užduoties atlikėjas neparengamas.

Veikiama varžybų principu. Galimos kelios variacijos:

- Pateikus rezultatus, iš jų išrenkamas nugalėtojas (gali būti keli) pagal teisingiausią, labiausiai tinkamą ar kitus kriterijus atitinkantį sprendimą.
- Pateikiami tarpiniai rezultatai prieinami visiems. Dalyviai gali pasinaudoti idėjomis gerinant savo sprendimą.
- Užsakovas komentuoja gaunamus rezultatus. Pateikia pastabas sprendimams. Dalyviai atitinkamai tobulina sprendimus.
- Variantas tinkamas kai ieškoma naujų sprendimų, turint tik idėją, neturint realizacijos specifikacijos. Tikimasi gauti keletą sprendimų, taip surandant tinkamiausią.

Reikalinga didelė bendruomenė, nes reikalinga jog dalyvautų keli galintys atlikti užduotį tinkamai. Esant specifinės srities užduočiai, neturint didelės bendruomenės gali būti sunku surinkti dalyvių.

2.3.2 Kandidatas pasirenka užduotį, be aktyvaus užsakovo.

Kandidatas iš jam prieinamų užduočių pasirenka užduotį, ir ją atlieka. Parinkimo procese iš esmės nedalyvauja užduoties pateikėjas. Jis tik pateikia informaciją kandidatams. Savaiame variantas labai tinkamas mažoms, nekritinėms užduotims, taip pat užduotims turinčioms aiškia įgyvendinimo specifikaciją. Norint jį naudoti svarbioms užduotims atlikti reikalinga sistema leidžianti sumažinti riziką dėl kandidato netinkamumo [5].

2.3.3 Kandidatas pasirenka užduotį, su aktyviu užsakovu.

Kandidatai pateikia savo klausimus, pastabas. Pagal tai tikslinami (siaurinami ar platinami) reikalavimai [4], [5].

Variantas tinkamas užduotis pateikiant pagal atitikimą reikalavimams arba kai nėra iš anksto tiksliai žinomi visi reikalavimai (pvz., užsakovas negali įvertinti trukmės nes pilnai nežino darbo specifikos). Galima išskaidyti etapais, t.y. papildžius informaciją užduotis laikinai užšaldoma (neleidžiama pasirinkti).

Renkantis ar sudarant kandidato pasirinkimo modelį sistemoje reikia atsižvelgti į aspektus, tokius kaip:

- užduoties kritiškumas (kaip svarbu gauti sprendimą laiku, kaip svarbu gauti teisingą sprendimą, ir t.t.);
- užduoties standartiškumas (ar ieškoma naujų sprendimų, ar tiesiog įgyvendinama);
- reikalingas srities išmanymas (užduoties atlikėjas turi teisingai traktuoti naudojamus terminus specifinius srities terminus, kitur galimai naudojamus kita prasme);
- kokius realius įsipareigojimus prisiima kandidatas pasirinkdamas užduotį (įsipareigoja tinkamai atlikti užduotį iki galo ar gali bet kuriuo metu atsisakyti be pasekmių).

2.4 Informacijos sinchronizavimas

Informacijos sinchronizavimas su sistemomis, tarp vienos įmonės projekto valdymo sistemos ir paslaugų valdymo sistemos, bei tarp kitų įmonių projekto valdymo sistemų yra svarbi problema. Jei informacija skirsis sistemose, gali kilti konfliktai tarp užduočių įgyvendinimo.

Informacijos sinchronizavimui galima panaudoti įvairius modelius. Tai gali būti realaus nuo laiko pranešimų tiesiai sistemoms iki pasikeitimų kaupimo atskirame modulyje (sinchronizavimo sistemoje) ir pateikimo tik įvykus užklausoms.

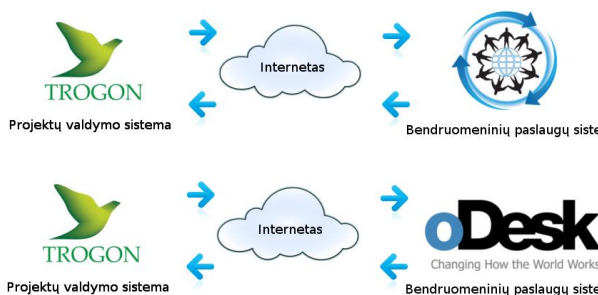
Didžiausia sinchronizavimo problema sistemoje yra tai, jog daug vartotojų vienu metu gali bandyti keisti informaciją. Kritinė sritis yra užduoties statuso keitimas. Čia yra ir informacijos konkurencijos problema. Reikalinga, kad keli vartotojai negalėtų vykdyti tos pačios užduoties (jei ji nėra tam pritaikyta), ar užsakovas savavališkai panaikinti jau vykdomą užduotį. Užduoties apsiėmimo konkurencijos problemą galima spręsti sudarant neveiklumo langą, t.y. pasirinkus vykdyti užduotį ji neiškarto atsiranda pas vartotoją. Išlaukiamas nustatytas laikas, per kurį užduotį gali apsiiminėti ir kiti. Vėliau išrenkamas tinkamiausias kandidatas. Arba galima užduotį priskirti tiesiog tam, kurio pranešimas apie užduoties apsiėmimą sistemą pasiekė pirmasis, kitiems bandantiems pranešant, jog jie nespėjo.

2.5 Bendruomeninių išteklių sistemos panaudojimas

Bendruomeninių išteklių sistemos/bendruomenės sukūrimas reikalauja investicijų. O modelio veikimui reikalinga sąsaja bendravimui su sistema.

3 SISTEMOS ĮGYVENDINIMAS

Toliau apžvelgiamas projektų valdymo sistemos išplėtimas bendruomeninėmis paslaugomis, remiantis kuriama sistema pasiūlyta [2]. Jos tikslas yra supaprastinti bendruomeninių resursų panaudojimą įmonėje.



1 pav. PVS sąveika su bendruomeninių paslaugų sistema [2]

Sistemai reikalinga turėti sąsają bendravimui su bendruomeninius išteklius teikiančia sistema.

Pradinius sistemos įgyvendinimo atveju pasirinktas oDesk bendruomeninių paslaugų tiekėjas. Pagrindinė sistemos išplėtimo bendruomeninėmis paslaugomis dalis šiuo atveju yra informacijos tarp PVS ir oDesk sistemų

sinchronizacija ir sistemų funkcijų tarpusavio papildymas.

3.1 Reikalingas funkcionalumas

Prie PVS prijungiant bendruomenines paslaugas reikalinga pasiekti tam tikrą (bazinį) funkcionalumą. Pagrindinės reikalingos funkcijos yra šios:

- užduočių pateikimas;
- užduočių atnaujinimas;
- užduočių pašalinimas;
- užduoties iš bendruomenės paėmimas;
- užduočių peržiūra;

3.2 Įgyvendinimas

Trogon projektų valdymo sistemoje atvaizduojami naujausi Web programavimo srities darbai iš oDesk sistemos. Pretendavimas atlikti užduotį atliekamas oDesk sistemoje užpildžius reikiamą informaciją.

Užduotis pateikti galima tiek per Trogon sistemą tiek ir pačioje oDesk sistemoje. Užduočių nebūtina pateikti bendruomenei. Sukūrus užduotį Trogon sistemoje ji figūruoja tik joje. Pateikiant užduotį bendruomenei užpildoma papildoma informacija, tokia kaip: skiriama kaina, reikalavimai užduoties atlikėjams.

Darbuotojų atrinkimas, interviu, kiti derinimo bei apmokėjimo darbai atliekami oDesk sistemoje.

Sinchronizuojama porcijomis – užduočių paketais, atnaujinama jei informacija yra pakitusi priklausomai nuo paskutinio atnaujinimo laiko. Informacijos sinchronizavimo metu surenkama reikalinga informacija iš oDesk platformos. PVS sukuriamos ar atnaujinamos bendruomeninės užduotys ir jų būsenos. Jei pradeda vykdyti bendruomeninė užduotis, ji bus dedama į bendruomeninių užduočių projektą. Šias užduotis reikia perskirti realiems projektams. Taip pat jas galima suskaidyti į smulkesnius darbus, tačiau pastarieji figūruos tik Trogon sistemoje ir neturės įtakos informacijai oDesk sistemoje.

3.3 Problemos

Naudojantis trečiosios šalies sistemomis reikia prisitaikyti prie jos teikiamo funkcionalumo.

Šiuo atveju oDesk pateikia bendravimui skirtą sąsają (angl. „API“). Ši sąsaja yra nuolat tobulinama ir plėtojama, tačiau dalies norimo funkcionalumo pasiekimas yra sudėtingas ar neįmanomas. Todėl negalima įgyvendinti nuoseklios sąsajos vien tik PVS – tenka norint dalį funkcionalumo pasiekti naudotis pačia bendruomeninių išteklių platforma. Kita API problema kylanti iš to paties nuosekliaus plėtimo – vienu metu skirtingiems elementams veikai skirtingų API versijų kreipiniai, nes ne viskas yra iškart atnaujinama.

Vartotojų autorizacija. Galimai skirtingas PVS ir bendruomeninių išteklių sistemos teisių vartotojams valdymo mechanizmas ir logika. Norint pasiekti suderinamumą ateityje ir galimybę vėliau prijungti kitas bendruomeninių išteklių sistemas reikalingas atsižvelgimas į šiuos modelius. Kaip galimas sprendimas įgyvendintas Trogon PVS yra leidimo prieiti prie bendruomeninių paslaugų per vieną bendrą vartotoją. Sistemoje paprastiems darbuotojams bendruomeninės paslaugos nėra svarbios, dėl parinkto sinchronizavimo mechanizmo.

3.4 Alternatyvos

Vietoj naudojimo trečiosios šalies bendruomeninių paslaugų valdymo sistemos, funkcionalumą galima įgyvendinti tarp pačių PVS. Tačiau tada reikia išspręsti problemas tarp informacijos dalinimosi tarp įmonių ir apmokėjimo už darbus.

4 IŠVADOS

Straipsnyje apžvelgtos savybės reikalingos bendruomeninių paslaugų sistemai, tinkamai

projektų valdymo sistemos išplėtimui. Apžvelgta bandomoji PVS sistema išplėsta bendruomeninėmis paslaugomis ir jos įgyvendinimo problemos.

Pagrindiniai trūkumai kyla iš dabartinių bendruomeninių išteklių tiekėjų pateikiamo funkcionalumo ribotumo.

Tolimesnė galima darbų kryptis: bendruomeninių paslaugų panaudojimas ne per tarpininką, o tarp pačių PVS, panaudojant įskiepius.

5 LITERATŪRA

- [1] Hirth, M.; Hofffeld, T., Tran-Gia, P.; *Cost-Optimal Validation Mechanisms and Cheat-Detection for Crowdsourcing Platforms; 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 316- 321 p.*
- [2] Ašeriškis, D.; Tamošaitis, J.; *Kitokia PJ kompanija Mag&Doc IT2012: tarpuniversitetinės konferencijos pranešimų medžiaga. Kaunas, 2012, p. 49-52.*
- [3] Zhang, W.; Zheng, X.; *Research of Web-based Negotiation Support System in Collaborative Commerce; IEEE International Conference on e-Business Engineering (ICEBE'06); 0-7695-2645-4/06*
- [4] Ebner, W.; Leimeister, M.; Bretschneider, U.; Krcmar, H.; *Leveraging the Wisdom of Crowds: Designing an IT-Supported Ideas Competition for an ERP Software Company, Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, 7-10 Jan. 2008, 417 p.*
- [5] Psailer, H.; Skopik, F.; Schall, D.; Dustdar, S.; *Resource and Agreement Management in Dynamic Crowdcomputing Environments; 2011 15th IEEE International Enterprise Distributed Object Computing Conference, 193 – 202 p.*
- [6] Howe J. *The Rise of Crowdsourcing. Wired Magazine, Nr. 14.06 - June 2006.*
- [7] MILLARD, W.; *The Wisdom of Crowds, the Madness of Crowds- Rethinking Peer Review in the Web Era, Annals of Emergency Medicine, Volume 57, no 1, January 2011; 13A - 20A p.*