



Kauno technologijos universitetas

Informatikos fakultetas

Rizikų valdymo metodika „Agile“ projektuose

Baigiamasis magistro projektas

Adelė Nečionytė

Projekto autorė

Asist. Lina Bisikirskienė

Vadovė

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Rizikų valdymo metodika „Agile“ projektuose

Baigiamasis magistro projektas

Veiklos skaitmeninimas ir sistemų architektūros (6211BX009)

Adelė Nečionytė

Projekto autorė

Asist. Lina Bisikirskienė

Vadovė

Prof. prakt. Jaroslav Karpovič

Recenzentas

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Adelė Nečionytė

Rizikų valdymo metodika „Agile“ projektuose

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Adelė Nečionytė

Patvirtinta elektroniniu būdu

Nečionytė, Adelė. Rizikų valdymo metodika *Agile* projektuose. Magistro projektas / vadovė asist. dr. Lina Bisikirskienė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacijos sistemos, Informatikos mokslai.

Reikšminiai žodžiai: *Agile* projektas, artefaktas, atributas, rizika, rizikos valdymas.

Kaunas, 2024. 101 p.

Santrauka

Agile projektų valdymo filosofija tampa vis populiarsnė ir taikoma įvairiose srityse, ne tik programinės įrangos kūrimo. Pagrindiniai jos principai – atvira komunikacija ir lankstus produktų pateikimo rinkai vertinimas. Skirtingai nuo tradicinio krioklio metodo, Agile projektus planuojama iteracijomis, su mažesniais etapais ir aiškiais uždaviniais. Nors Agile sumažina riziką dėl vizijos, terminų ar biudžeto pokyčių, ji neturi vieningos rizikų valdymo metodikos ir mechanizmų vėlesniuose etapuose išskylančioms problemoms spręsti. Taip pat trūksta įrankių, skirtų rizikoms sekti ir kontroliuoti įvairiuose projekto etapuose. Šiame rašto darbe aprašoma metodika, kurioje pagrindiniai Agile projektų artefaktai yra įvertinami pagal savo atributų statusus - reguliariai atnaujinant atributų duomenis, galima anksti atskleisti galimą riziką ir ją šalinti, taip didinant projekto komandos veiksmingumą ir suinteresuotųjų šalių aiškumą teikiant skaidrias ataskaitas. Metodika apima rizikos identifikavimą ir kontrolę viso rizikos gyvavimo ciklo metu. Pasiūlyta metodika patikrinama eksperimentiškai su trimis projektais.

Nečionytė, Adelė. Risk management methodology in *Agile* projects. Master's Final Degree Project / supervisor assist. dr. Lina Bisikirskienė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information Systems, Computing.

Keywords: Agile project, artifacts, attributes, risk, risk management.

Kaunas, 2024. 101 p.

Summary

The Agile project management philosophy is becoming more and more popular and is being applied in many areas, not only in software development. Its main principles are open communication and flexible assessment of the product to market. Unlike the traditional waterfall approach, Agile projects are planned in iterations, with smaller phases and clear objectives. While Agile reduces the risk of changes in vision, timelines or budget, it does not have a unified risk management methodology and mechanisms to deal with problems that arise in later phases. It also lacks tools to track and control risks at different stages of the project. This paper describes a methodology in which key artefacts of Agile projects are evaluated according to their attribute status - by regularly updating attribute data, potential risks can be detected and addressed early, thereby increasing the efficiency of the project team and providing transparent reporting to stakeholders. The methodology includes risk identification and control throughout the risk lifecycle. The proposed methodology is tested experimentally with three projects.

Turinys

Lentelių sąrašas	9
Paveikslų sąrašas	11
Santrumpų ir terminų sąrašas	12
Įvadas	13
Darbo problematika ir aktualumas	13
Tyrimo tikslas ir uždaviniai.....	14
Tyrimo problema, objektas ir sritis	14
Darbo rezultatai ir jų svarba	14
1. Probleminės srities analizė	16
1.1. „Agile“ projektų tipai.....	16
1.1.1. „Scrum“ metodika	17
1.1.2. „Kanban“ metodika	18
1.1.3. „Scrum“ ir „Kanban“ metodikų palyginimas.....	19
1.2. „Agile“ projektų artefaktai	21
1.2.1. Produkto tikslas	21
1.2.2. Produkto atliekamų darbų sąrašas	21
1.2.3. Sprinto tikslas.....	22
1.2.4. Sprinto atliekamų darbų sąrašas.....	22
1.2.5. Atlikto darbo apibrėžimas.....	22
1.2.6. Produkto prieaugis.....	23
1.2.7. Sprinto eigos diagrama	24
1.3. Rizikų gyvavimo ciklas.....	24
1.3.1. Rizikos nustatymas	25
1.3.2. Rizikos analizė ir valdymas	25
1.3.3. Rizikos sekimas.....	26
1.4. Pagrindinės rizikos „Agile“ projektuose.....	26
1.4.1. Neapibrėžti reikalavimai.....	27
1.4.2. Su projekto komanda susijusios rizikos.....	28
1.4.3. Su kūrimo procesu susijusios rizikos	29
1.4.4. Aplinkos veiksniai	31
1.4.5. Taikomos technologijos.....	33
1.5. Rizikos valdymas „Agile“ projektuose ir komandos reakcija	34
1.5.1. Pagrindinės rizikos valdymo metodikos.....	34
1.5.2. Komandos reakcija į rizikos valdymą	36
1.6. Siekiamo sprendimo apibrėžimas	36
1.7. Analizės išvados	37
2. Rizikų valdymo metodikos „Agile“ projektuose aprašas	39
2.1. Rizikų valdymo sistemos reikalavimai ir panaudojimo atvejai.....	39
2.2. Rizikų valdymo metodikos procesas.....	42
2.2.1. Dalykinės srities eksperto procesas.....	42
2.2.2. „1. Apibrėžti <i>Scrum</i> projektų artefaktų tipus“ žingsnis	43
2.2.3. „2. Apibrėžti <i>Scrum</i> projektų atributus“ žingsnis.....	44

2.2.4.	„3. Apibrėžti rizikų kategorijas“ žingsnis	45
2.2.5.	„4. Priskirti <i>Scrum</i> projektų rizikas rizikų kategorijai“ žingsnis	46
2.2.6.	„5. Sukurti šabloninį projektą“ žingsnis	46
2.2.7.	„6. Susieti projekto artefaktą su atributu“ žingsnis	47
2.2.8.	„7. Priskirti rizikas artefakto atributui“ žingsnis	48
2.2.9.	Projekto dalyvio procesas	49
2.2.10.	„1. Apibrėžti projekto specifiką“ žingsnis	50
2.2.11.	„2. Pasirinkti standartizuotą projektą“ žingsnis	52
2.2.12.	„3. Suformuoti naują projektą“ žingsnis	52
2.2.13.	„4. Priskirti rizikos reikšmę artefaktų atributams“ žingsnis	53
2.2.14.	„5. Priskirti stulpelių reikšmes apskaičiuojamiems atributams“ žingsnis	53
2.2.15.	„5. Importuoti projekto duomenis“ žingsnis	54
2.2.16.	„6. Pridėti reikšmes artefaktų atributams“ žingsnis	55
2.2.17.	„7. Valdyti rizikos objektą“ žingsnis	55
2.2.18.	„7.1. Identifikuoti rizikos objektą“ žingsnis	56
2.2.19.	„7.2. Fiksuoti rizikos objektą“ žingsnis	56
2.2.20.	„7.3. Atnaujinti rizikos objekto atributų reikšmes“ žingsnis	57
2.3.	Reikalavimų apibendrinimas	57
3.	Rizikų valdymo metodikos eksperimentinės realizacijos projektas	58
3.1.	Esybės ryšių diagrama	58
3.2.	Esybės „Rizika“ būsenų modelis	61
3.3.	Sistemos loginė architektūra	62
3.4.	Naudotojo sąsajos modelis	64
3.5.	Realizacijos modelis	66
3.5.1.	Komponentų diagrama	66
4.	Eksperimentinis tyrimas	68
4.1.	Projektas „DVS“: Darbuotojų atrankos ir vertinimo platforma	68
4.1.1.	Žingsnis „1. Apibrėžti projekto specifiką“	68
4.1.2.	Žingsnis „2. Pasirinkti standartizuotą projektą“	70
4.1.3.	Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“	70
4.1.4.	Žingsnis „5. Priskirti rizikos reikšmę artefaktų atributams“	71
4.1.5.	Žingsnis „5. Importuoti projekto duomenis“	71
4.1.6.	Žingsnis „6. Pridėti reikšmes artefaktų atributams“	71
4.1.7.	Žingsnis „7.1. Identifikuoti rizikos objektą“	72
4.1.8.	Žingsnis „7.2. Fiksuoti rizikos objektą“	74
4.1.9.	Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“	76
4.2.	Projektas „EBS“: Elektroninės ir mobilios bankininkystės platforma	78
4.2.1.	Žingsnis „1. Apibrėžti projekto specifiką“	78
4.2.2.	Žingsnis „2. Pasirinkti standartizuotą projektą“	79
4.2.3.	Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“	80
4.2.4.	Žingsnis „7.1. Identifikuoti rizikos objektą“	81
4.2.5.	Žingsnis „7.2. Fiksuoti rizikos objektą“	82
4.2.6.	Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“	83
4.3.	Projektas „HRS“: Darbo užmokesčio apskaitos platforma	85

4.3.1. Žingsnis „1. Apibrėžti projekto specifiką“	86
4.3.2. Žingsnis „2. Pasirinkti standartizuotą projektą“	87
4.3.3. Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“	88
4.3.4. Žingsnis „7.1. Identifikuoti rizikos objektą“	88
4.3.5. Žingsnis „7.2. Fiksuoti rizikos objektą“	90
4.3.6. Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“	92
4.4. Eksperimento išvados.....	94
4.4.1. Ekspertų nuomonė apie metodikos naudingumą.....	97
5. Išvados	99
Literatūros sąrašas	100
Priedai	102
1 priedas. Artefaktų sąrašas	102
2 priedas. Atributų sąrašas.....	102
3 priedas. Rizikų kategorijos	107
4 priedas. Rizikos pagal rizikos kategoriją.....	108
5 priedas. Naudotojo sąsajos modelis	111
6 priedas. Projekto „DVS“ artefaktų ir atributų su priskirtomis rizikomis sąrašas	115
7 priedas. Interviu su projekto „DVS“ vystymo komanda	124
8 priedas. Straipsnis, pateiktas į ICIST 2024 konferenciją	127

Lentelių sąrašas

1 lentelė. <i>Scrum</i> ir <i>Kanban</i> metodikų palyginimas	20
2 lentelė. <i>Scrum</i> rizikų valdymo metodikų palyginimas.....	35
3 lentelė. Rizikų valdymo metodikų palyginimas pagal kriterijų	35
4 lentelė. <i>Scrum</i> projekto artefaktų pavyzdžiai	44
5 lentelė. <i>Scrum</i> projektų atributų pavyzdžiai.....	45
6 lentelė. <i>Scrum</i> projektų rizikų kategorijų pavyzdžiai	46
7 lentelė. <i>Scrum</i> projektų rizikų pavyzdžiai.....	46
8 lentelė. Šabloninio <i>Scrum</i> projekto charakteristikos	47
9 lentelė. "Projekto artefakto tipas" lentelės pavyzdys.....	47
10 lentelė. "Projekto artefakto atributas" lentelės pavyzdys	48
11 lentelė. Projekto rizikos lentelės pavyzdys	49
12 lentelė. Atributų išraiškos priskyrus įkeliamų duomenų failo stulpelius	51
13 lentelė. Atributų išraiškos priskyrus įkeliamų duomenų failo stulpelius	54
14 lentelė. Rizikų valdymo sistemos esybių lentelė	59
15 lentelė. Rizikos objekto gyvavimo ciklas.....	61
16 lentelė Sistemos valdiklių aprašas	62
17 lentelė. Projekto „DVS“ aprašymas.....	69
18 lentelė. Projekto "DVS" artefaktai ir jų įkeliamų duomenų failų šablonai	69
19 lentelė. Projekto „DVS“ naudojamų komponentų kiekybiniai rodikliai	70
20 lentelė. Artefaktų įkeliamų duomenų failai.....	70
21 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui.....	72
22 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui.....	72
23 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui	73
24 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	74
25 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	74
26 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui	75
27 lentelė. Užfiksuotų rizikų atnaujinimo rodikliai.....	76
28 lentelė. Projekto „EBS“ aprašymas	78
29 lentelė. Projekto "EBS" artefaktai ir jų įkeliamų duomenų failų šablonai.....	79
30 lentelė. Projekto „EBS“ naudojamų komponentų kiekybiniai rodikliai	80
31 lentelė. Artefaktų įkeliamų duomenų failai.....	80
32 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui.....	81
33 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui.....	81
34 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui	82
35 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	82
36 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	83
37 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui	83
38 lentelė. Užfiksuotų rizikų atnaujinimo rodikliai.....	84
39 lentelė. Projekto „HRS“ aprašymas	86
40 lentelė. Projekto "HRS" artefaktai ir jų įkeliamų duomenų failų šablonai	87
41 lentelė. Projekto „HRS“ naudojamų komponentų kiekybiniai rodikliai.....	87
42 lentelė. Artefaktų įkeliamų duomenų failai.....	88

43 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui	88
44 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui	89
45 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui	90
46 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	91
47 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui.....	91
48 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui	92
49 lentelė. Užfiksuotų rizikų atnaujinimo rodikliai.....	92
50 lentelė. Apklausoje dalyvavę asmenys.....	97

Paveikslų sąrašas

1 pav. <i>Agile</i> yra bendras terminas daugeliui metodikų	17
2 pav. <i>Kanban</i> lentos pavyzdys	19
3 pav. <i>Scrum</i> gyvavimo ciklas Pritaikyta pagal Ettinger (2016)	24
4 pav. Neapibrėžtumo ir sudėtingumo modelis pagal Stacey sudėtingumo modelį	27
5 pav. Rizikos gyvavimo ciklo procesas	39
6 pav. Rizikų dalykinės srities eksperto metodikos panaudos atvejai	40
7 pav. Projekto dalyvio rizikų valdymo modulio panaudojimo atvejai	41
8 pav. Rizikų dalykinės srities eksperto procesas	43
9 pav. „7. Priskirti riziką artefakto atributui“ žingsnis	49
10 pav. Rizikų valdymo modulio veiklos diagrama	50
11 pav. „1. Apsibrėžti projekto specifiką“ žingsnis	51
12 pav. „3. Suformuoti naują projektą“ žingsnis	52
13 pav. „4. Priskirti stulpelių reikšmes importuojamiems atributams“ žingsnis	54
14 pav. „7. Valdyti rizikos objektą“ žingsnis	55
15 pav. „7.1 Identifikuoti rizikos objektą“ žingsnis	56
16 pav. „7.2 Fiksuoti rizikos objektą“ žingsnis	57
17 pav. Esių ryšių diagrama	58
18 pav. Rizikų valdymo modulio klasių diagrama	59
19 pav. Projekto artefakto rizikos būsenų diagrama	61
20 pav. Sistemos loginė architektūra	64
21 pav. Rizikų valdymo sistemos navigacijos planas	65
22 pav. Rizikų valdymo sistemos komponentų diagrama	67
23 pav. Eksperimente naudojamų ir identifikuotų komponentų palyginimas	95
24 pav. Eksperimente atpažintų ir užfiksuotų rizikų kiekis	96
25 pav. Rizikos objektų atnaujinimo palyginimas	97

Santrumpų ir terminų sąrašas

Santrumpos:

UML – Vieninga modeliavimo kalba (angl. *Unified Modelling Language*)

BPMN – Verslo procesų modeliavimo žymėjimas (angl. *Business Process Modelling Notation*)

Terminai:

Projektas – pradžia, pabaigą ir baigtinius išteklius turinti laikina veikla, kuria siekiama unikalios tikslo (pvz., produkto, paslaugos ar rezultato sukūrimo).

Komponentas – informacinės sistemos, informacijos ar duomenų apdorojimo sistemos, techninės bei programinės įrangos, procesų, programos arba procedūros sudėtinė dalis, galinti atlikti tam tikrą veiksmą.

Rizika – galimi faktinių rezultatų nuokrypiai nuo laukiamų. Dažniausiai siejama su nepageidaujamo, beverčio ar žalingo rezultato, klaidos ar nuostolių tikimybe. Tai neapibrėžtumo poveikis pokyčio, sprendimo ar įmonei. Projekto komanda bendradarbiauja su kitomis suinteresuotosiomis šalimis, kad nustatytų, vertinti, nustatyti riziką, nustatyti jos prioritetus ir spręsti šios rizikos problemas, keičiant tikimybę, kad sąlygas ar įvykius, dėl kurių kyla neapibrėžtumas: mažinti pasekmes, pašalinant rizikos šaltinį, visiškai išvengiant rizikos nepradėti ar tęsti veiklą, kuri sukelia riziką [17].

Ivadas

Agile projektų vystymo ir valdymo filosofija tampa vis populiareesnė ir dažniau pritaikoma vis daugiau Lietuvos bei užsienio įmonių. Projektų komandos pradėjo vadovautis *Agile* principais ne tik programinės įrangos kūrimo, o taip pat ir marketingo, švietimo, finansų bei kitose srityse. Pagrindiniai šios projektų valdymo filosofijos principai yra atvira komunikacija, noras lanksčiai vertinti produktų ir rezultatų pateikimo rinkai būdus [1].

Agile projektų valdymo metodikos, tokios kaip *Scrum* ar *Kanban*, nuo tradicinės krioklio (angl. *waterfall*) projekto valdymo metodikos skiriasi tuo, kad projekto eiga yra planuojama iteracijomis. Projektas yra išskaidomas į smulkesnius etapus su mažesniais uždaviniais, kuriuos įgyvendinus yra vykdomos demonstracijos suinteresuotiems asmenims siekiant gauti grįžtamąjį ryšį bei atlikti reikiamus pakeitimus. Krioklio valdymo metodika projekto etapus, apimtį ir planuojamą biudžetą apibrėžia ankstyvoje projekto stadijoje. Etapai atliekami vienas po kito, vėlesnis projekto vystymo etapas nepradedamas nepabaigus ankstesnio. Dažnai toks projektų planavimo metodas apriboja lankstumą bei poreikių ar prioritetų pasikeitimus vėlesniuose projekto vystymo etapuose.

Iteraciniu projekto valdymo būdu, kuris yra vienas iš esminių *Agile* skirtumų nuo tradicinės krioklio valdymo metodikos, bandoma sumažinti galimas rizikas dėl projekto realizavimo vizijos pasikeitimo, terminų ar biudžeto pokyčių, tačiau *Agile* projekto valdymo metodika pati savaime nėra pakankama rizikoms eliminuoti. Nors iteracinė metodika suteikia galimybę sekti ir reguliariai tikrinti projekto progresą, laiku pastebėti reikalavimų neatitikimus ar poreikių pasikeitimus, tačiau to gali neužtenka rizikoms valdyti. *Agile* projektų valdymo metodika nesukuria veiksmingo mechanizmo projekto problemoms spręsti, kurios atsiranda vėlesniuose projekto valdymo etapuose, taip pat nėra apibrėžiamos vieningos metodikos, kaip valdyti rizikas skirtingoms projekto sudedamosioms dalims. Šiuo metu pagrindiniai įrankiai, pritaikyti *Agile* projektų valdymui, nesuteikia galimybės sekti ir kontroliuoti rizikas įvairiuose projekto etapuose.

Darbo problematika ir aktualumas

Rizikų identifikavimas ir valdymas yra vienas iš svarbiausių projekto sėkmės faktorių [23], tačiau šiuolaikiniuose projektų valdymo įrankiuose nėra vieningo sprendimo, kuris galėtų apibrėžti, laiku identifikuoti ir valdyti galimas rizikas, kurios gali iškilti projekto eigoje. Viena pagrindinių to priežasčių – *Agile* projektuose pagrindiniai faktoriai, lemiantys projekto sėkmę yra projekto komanda (skirtinga kultūrinė aplinka, darbo proceso ir technologijų standartai bei skirtingi interesai ir tikslai), o žmogiškojo faktoriaus rizikos kategorija gali būti sunkiai apčiuopiama ir valdoma [3]. Taip pat, nors *Agile* projektų valdymo struktūra pati savaime yra eliminuojanti nemažai rizikų, lyginant su krioklio valdymo principais [25], tačiau tai gali sukelti papildomų architektūros valdymo rizikų. Ypač pabrėžiamas informacijos ir reikalavimų decentralizavimas, kurie yra susiję su dokumentacijos bei funkcionalumo procesų aprašymų perkėlimu į modelius – dažniausiai UML/BPMN diagramas. Taip pat verta išskirti rizikų kategoriją – aplinkos rizikos, kurios kyla dėl užsakovų ir kitų suinteresuotų asmenų nepakankamo įsitraukimo į projektą [4].

Šiuolaikinės priemonės nenumato rizikos identifikavimo ir klasifikavimo skirtinguose *Agile* projektų procesuose, tokiuose kaip reikalavimų apibrėžimas ir analizė, procesų ir dizainų modeliavimas, darbų sąrašo sudarymas ir prioretizavimas, darbų skaidymas ir priėmimo kriterijų aprašymas, testavimas, dokumentacija. Kadangi skirtinguose *Agile* projektų komponentuose yra identifikuojamos skirtingos rizikų kategorijos, yra reikalinga metodika, kaip šias rizikas apibrėžti ir valdyti kiekvienam iš šių komponentų. Metodika turi užtikrinti sklandų rizikų identifikavimo procesą kiekviename iš projekto vystymo etapų bei išlaikyti *Agile* principus, neapsunkinant komandos narių perteklinėmis, laikui imliomis veiklomis.

Tyrimo tikslas ir uždaviniai

Tyrimo tikslas – sudaryti prielaidas identifikuoti bei valdyti projekto rizikas sudarant reikalingą metodiką rizikų apibrėžimui ir valdymui skirtingiems artefaktams *Agile* projektuose.

Tyrimo uždaviniai:

1. Atlikti *Agile* projekto sudedamųjų dalių bei jų įtakos projekto vykdymui analizę;
2. Atlikti rizikų valdymo metodų bei jų taikymo projektų valdyme analizę;
3. Parengti rizikų nustatymo metodiką *Agile* projekto artefaktams integruojant artefaktų atributus, susietus su rizikų klasifikatoriumi;
4. Sukurti prototipą, kurio pagalba būtų kuriamos, integruojamos bei sekamos rizikos *Agile* projektuose ir jį ištestuoti;
5. Suformuluoti išvadas.

Tyrimo problema, objektas ir sritis

Tyrimo problema – *Agile* projektuose nepakankamai dėmesio skiriama įvertinti ir kontroliuoti rizikas, susijusias su skirtingomis projekto sudedamosiomis dalimis. Projekto valdymo sistemose nėra vieningos metodikos, apibrėžiančios rizikos klasifikatorius bei jų įtaką kiekvienam iš projekto komponentų. Nėra vieningo sprendimo veiksmingam rizikų sekimui kiekvienai pagrindinių projekto sudedamųjų dalių.

Tyrimo objektas – rizikų valdymas *Agile* projektuose.

Tyrimo objekto naudotojai yra komanda, dirbanti vadovaudamasi *Agile* projektų valdymo principais

Tyrimo sritis – informacinių sistemų projektų įgyvendinimas pagal *Agile* principus įtraukiant rizikų valdymo procesą.

Darbo rezultatai ir jų svarba

Darbo rezultatas yra sukurti bei apibrėžti metodiką, kuri efektyvintų *Agile* projektų planavimo bei vystymo valdymą sumažindama neidentifikuotų rizikų tikimybę skirtingiems projekto komponentams.

Agile projektų valdymo principai padeda eliminuoti dalį rizikų, kurios galėtų kilti vystant projektą kriklo valdymo principu, kadangi iteracinė metodika yra lankstesnė ir lengviau

prisitaikanti prie pokyčių. Tačiau *Agile* principai neužtikrina visiško rizikų eliminavimo, kurios gali atsirasti įvairiuose projekto valdymo komponentuose ilgalaikėje perspektyvoje. Skirtinguose projekto valdymo komponentuose yra išskiriami skirtingi rizikų rinkiniai, kurie turi būti skirtingai identifikuojami ir valdomi. Laiku neidentifikuotos ir nesuvaldytos rizikos gali sukelti neigiamų pasekmių tolimesnei projekto eigai, tokių kaip viršijamas biudžetas, vėluojantys terminai, persidirbantys žmogiškieji resursai.

Vieninga rizikų valdymo metodika įvairiems projekto vystymo procesams identifikuotų priemonės, kaip atpažinti ir valdyti galimas rizikas bei padėtų minimizuoti nenumatytų rizikų tikimybę.

Darbo struktūra:

- 1 skyrius. Aprašoma rašto darbo paskirtis, tikslas ir uždaviniai, darbo problematika, rezultatai ir jų svarba;
- 2 skyrius. Aprašoma rašto darbo probleminės srities analizė, tyrimo objektas, sritis ir problema. Atliekama *Agile* projektų struktūrų, jų planavimo metodų, rizikų valdymo metodų bei jų taikymo projektų valdyme analizė;
- 3 skyrius. Aprašoma rizikų nustatymo metodika pagal rizikų klasifikatorius, jų nustatymo ir susiejimo su *Agile* projekto komponentais;
- 4 skyrius. Sukuriamas prototipas, kurio pagalba būtų kuriamos, integruojamos bei sekamos metrikos *Agile* projektuose;
- 5 skyrius. Eksperimentiškai patikrinamas prototipo veikimas.

1. Probleminės srities analizė

Probleminės srities analizė turi padėti įvertinti, kaip turi būti įvertinamos rizikos kiekvienam projekto valdymo etapui ir paruošti tinkamą metodiką rizikoms kontroliuoti bei eliminuoti. Analizėje apibrėžiami pagrindinės *Agile* projektų valdymo metodikos, šių metodikų pagrindiniai komponentai (procesai), juose dažniausiai kylandčios rizikos, taip pat šios rizikos suskirstomos pagal rizikų klasifikatorių.

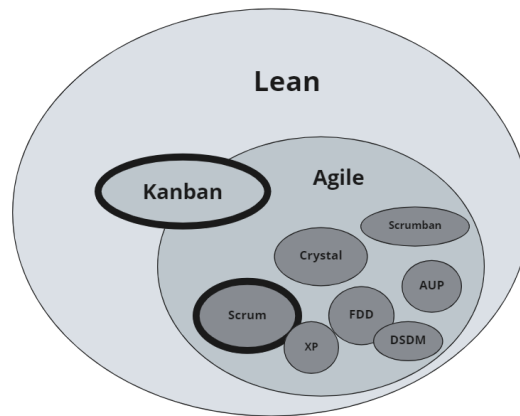
1.1. „Agile“ projektų tipai

Agile projektų valdymo manifestas susidaro iš keturių pagrindinių vertybių, kurios yra išskaidomos į dvylika pagrindinių principų, kurie yra kiekvienos iš *Agile* projekto valdymo metodikos pagrindas [5]:

1. Didžiausias prioritetas - patenkinti kliento poreikius laiku ir nuolat pristatyti programinę įrangą;
2. Palankiai vertinti besikeičiančius reikalavimus, net ir vėlyvuju kūrimo etapu. Lankstūs procesai leidžia pokyčius panaudoti kliento konkurenciniam pranašumui. Šis punktas nurodo lanksčiai vertinti besikeičiančias aplinkybes, kurios yra apibrėžiamos kaip vienas iš pagrindinių rizikos faktorių krioklio metodikoje. Tai gali lemti reikalavimų pasikeitimą, kurie yra apibrėžiami pradinėje projekto vystymo stadijoje. Reikalavimai gali būti apibrėžti neteisingai, nepakankamai išsamiai, taip pat gali atsirasti papildomų reikalavimų vėlesnėse projekto vykdymo stadijose. Šie veiksliai yra susiję su didesniu defektų kiekiu projekte, taip pat tai pareikalauja daugiau pastangų siekiant sėkmingai užbaigti projektą. [7] Šiuo *Agile* principu pabrėžiama, jog metodika neturėtų prisirišti prie išankstinių reikalavimų ir komanda turi gebėti lanksčiai prisitaikyti prie pokyčių be didesnių pastangų bei papildomų kaštų;
3. Dažnai (nuo poros savaitių iki poros mėnesių) pristatyti veikiančią programinę įrangą, pirmenybę teikiant trumpesniems terminams;
4. Verslo atstovai ir programuotojai turi kasdien dirbti kartu viso projekto metu;
5. Kurti projektus, kuriuose dalyvautų motyvuoti asmenys. Suteikti jiems reikiamą aplinką ir paramą ir pasitikėti, kad jie atliks darbą;
6. Veiksmingiausias būdas perduoti informaciją kūrimo komandai ir jos nariams yra pokalbis akis į akį;
7. Pagrindinė pažangos priemonė - veikianti programinė įranga;
8. Lankstūs procesai skatina tvarų vystymąsi. Užsakovai, kūrėjai ir naudotojai turėtų sugebėti išlaikyti pastovų tempą neribotą laiką;
9. Nuolatinis dėmesys techniniam meistriškumui ir geram dizainui didina lankstumą;
10. Paprastumas - menas maksimizuoti neatlikto darbo kiekį - yra labai svarbus;
11. Geriausios architektūros, reikalavimai ir projektai kuriami save organizuojančiose komandose;
12. Komanda reguliariai apmąsto, kaip tapti veiksmingesne, tada atitinkamai koreguoja ir pritaiko savo elgesį.

Šie pagrindiniai *Agile* principai tapo pamatu ne vienai iš projektų valdymo metodologijų. Žemiau pavaizduotame paveikslėlyje 1 pav. *Agile* nurodomas kitų projektų valdymų principų bei metodikų kontekste. *Agile* vizualizuojamas kaip bendra sąvoka, apibrėžianti metodikas,

karkasus ar praktikas, naudojančias Agile Manifesto požiūrį [5]. Taip pat, Agile nurodoma kaip Lean metodologijos poaibis, kuri yra grindžiamas dviem pagrindiniais principais: nuolatinio tobulėjimu ir pagarba žmonėms.



1 pav. Agile yra bendras terminas daugeliui metodikų

Pagrindinės Agile metodologijos, pritaikomos šiuolaikinių projektų vystymui yra Scrum ir Kanban bei šių metodikų kombinacijos [8].

1.1.1. „Scrum“ metodika

Scrum yra viena iš daugelio dažniausiai naudojamų iš daugelio *Agile* metodikų. *Scrum* yra iteracinė projektų valdymo metodika, skirta kontroliuoti riziką ir optimizuoti projekto nuspėjamumą. *Scrum* pasižymi trimis kartinėmis savybėmis [13]:

1. skaidrumas: procesas turi būti matomas visiems, kurie dalyvauja projekte;
2. tikrinimas: nuolatinė komponentų patikra siekiant aptikti problemas ankstyvose stadijose;
3. pritaikymas: jeigu kai kurie projekto komponentai yra nepriimtini ir neatitinka projekto apimties, procesas turi būti koreguojamas, kad būtų išvengta tolesnių problemų.

Scrum sudaro trys pagrindiniai komponentai: komandos rolės, įvykiai bei artefaktai. [13]

Scrum metodiką įgyvendina projekto komandos, kurios yra sudarytos iš produkto savininko (angl. *Product Owner*), *Scrum* meistro (angl. *Scrum Master*) ir inžinierių komandos. Kiekviena komanda yra savi organizuojanti, todėl sprendimų priėmimas yra atliekamas komandos viduje be išorinių žmonių įsikišimo.

Projekto vykdymo procesas yra vykdomas iteracijomis – sprintais. Kiekvieną sprintą galima laikyti kaip 1-4 savaitių trukmės projektą, kuriame yra suplanuota, kokie darbai turi būti atlikti. Sprinto planas bei tikslai neturėtų būti keičiami sprinto eigos metu, tačiau produkto savininkas ir inžinierių komanda prireikus gali keisti ir iš naujo apibrėžti sprinto apimtį bei numatomus darbus [12]. Kiekvieną sprintą apima pagrindiniai sprinto įvykiai – sprinto planavimas, kasdienis sprinto eigos aptarimas, sprinto peržiūra, sprinto retrospektyva. Šie įvykiai yra skirti skaidrumui užtikrinti ir projekto būklei tikrinti. Jei kuris nors įvykis yra neįvykdomas, sumažės projekto eigos skaidrumas ir bus prarasta galimybė tikrinti projekto būklę, identifikuoti numatomas rizikas.

Pasak K. Švėberio ir J. Suterlando [13], *Scrum* naudojamas iteracinis metodas siekia optimizuoti rizikos kontrolę. Autoriai teigia, kad sprintų naudojimas taip pat padeda valdyti riziką, nes riboja riziką iki vieno kalendorinio mėnesio sąnaudų. Tačiau yra teigiančių, kad nors ši metodika ir padeda lengviau identifikuoti rizikas, tačiau jų neeliminuoja. Pagrindiniai rizikų valdymo būdai įvardijami *Scrum* projektuose yra šie [12]:

1. Nuolatinis grįžimas prie projektų komponentų bei galimų rizikų kiekvieno sprinto metu - kad nuolatinis grįžtamasis ryšys yra pagrindinis projekto sėkmės veiksnys;
2. Pakartotinis žinių apie rizikas bei išmuktų pamokų panaudojimas - išmuktos pamokos paprastai naudojamos *Scrum* gyvavimo cikluose ir dažniausiai nustatomos bei aptariamose retrospektyviuose susitikimuose;
3. Pripažinimas, kad rizikos valdymas yra daugiau nei metodologinis procesas. Individualus elgesys ir bendradarbiavimas yra svarbus *Scrum* projekto vystymui - *Scrum* yra labiau susijęs su žmonėmis, elgesiu ir kultūra nei su procesais.

1.1.2. „Kanban“ metodika

Kanban išsivystė iš *Scrum*. Tiek *Kanban*, tiek *Scrum* vizualizuoja komandos darbo eigą ir planuojamas, vykdomas ir užbaigtas projekto užduotis naudojant fizinę arba skaitmeninę lentą [10]. *Kanban* lenta paprastai yra sudaroma iš šių dalių, kurios yra įvardijamos kaip užduoties statusas, tačiau ši statusų aibė *Kanban* lentoje gali būti modifikuojama pagal komandos poreikius:

1. Pateiktų darbų sąrašas (angl. *Backlog*) – šioje dalyje pateikiami kliento poreikiai, kurie turi būti atliekami;
2. Atlikimui paruošti darbai (angl. *Ready to start*) – kliento poreikiai, kurių reikalavimai yra vienareikšmiškai apibrėžti, suderinti su klientu ir patvirtinti kliento;
3. Atliekami darbai (angl. *Work in Progress* arba *WIP*) – darbai, kurie yra deleguoti ir atliekami šiuo metu;
4. Peržiūra (angl. *Review*) – darbai, kurie yra atlikti ir perduoti pakartotinei peržiūrai kitam komandos nariui;
5. Testavimas (angl. *Testing*) - darbai, kurie yra testuojami įvertinti, ar atliktas darbas tenkina užduočiai išskeltus reikalavimus. Šis veiksmas atliekamas darbo kokybei užtikrinti;
6. Dokumentavimas (angl. *Documentation*) – pabaigto darbo dokumentavimas siekiant išsaugoti informaciją apie atliktus pokyčius;
7. Atlikta (angl. *Done*) – atliktas darbas yra uždaromas.

Šie statusai yra išdėstomi *Kanban* lentoje iš kairės į dešinę (2 pav.), aktyvūs darbai yra perkeliama iš kairės į dešinę per visus statusus nepraleidžiant nei vienos lentos dalies. Galiausiai užbaigtos užduotys dedamos į galutinį stulpelį "Atlikta".

Pagrindiniai *Kanban* principai [14]:

1. Pereinamojo proceso greičio kontrolė;
2. Efektyvus specializuotų įgūdžių, srities žinių ar žinių apie senąją kodą paskirstymas organizacijoje. Specializuotos ir santykinai statiškos komandos, nors ir pageidautinos, šiuo požiūriu paprastai yra nepraktiškos;

3. Komandų aprūpinimas pagrindiniais principais, ypač taupumo principais ir apribojimų teorija;
4. Suteikti geresnį būdą mokytis, kaip tobulėti. Iteracijos pabaigos retrospektyvos gali būti per siauros ir per vėlyvos, kad būtų vertingos ilguoju laikotarpiu;
5. Suteikiama galimybė komandoms dirbti su nepertraukiamomis užduotimis. Naudojant laiko ribas, darbas suspaudžiamas į iš anksto nustatytą laikotarpį ir nesusijusios darbo dalys sugrupuojamos į vieną sprintą.



2 pav. Kanban lentos pavyzdys

Rizikos *Kanban* metodikoje gali būti valdomos įprastą *Kanban* lentą pavertus rizikoms valdyti modifikuota *Kanban* lenta. Šioje lentoje derinamos vertę kuriančios užduotys, tiesiogiai susijusios su galutiniam klientui skirtu produkto kūrimu, kartu su rizikos užduotimis, kurias reikia atlikti, kad būtų galima nuolat kurti vertę – rizika šiuo atveju įvertinama kaip papildoma užduotis prie kitų komandos įsipareigojimų. Tai leidžia komandoms vizualizuoti, kiek laiko skiriama rizikos valdymui. Rizikos apdorojimo metu komandos gali suformuluoti keletą užduočių, kurios padės joms valdyti grėsmes ir išnaudoti galimybes [15].

1.1.3. „Scrum“ ir „Kanban“ metodikų palyginimas

Vienas iš ryškiausių skirtumų tarp *Scrum* ir *Kanban* metodikų yra laiko apribojimas – *Scrum* naudoja laiko iteracijas, kuriomis yra planuojamas ir matuojamas atliktas darbas, kai *Kanban* WIP nėra apribota laiko rėmų. *Kanban* gali susidoroti su daugybe projekto trikdžių, palaikyti skirtingų įgūdžių turinčius darbuotojus ir pasižymėti pasikartojančiais darbais. *Kanban* taip pat puikiai tinka didesnėms komandoms nes bendravimo ir planavimo išlaidos yra mažesnės [12]. Priešingai, *Scrum* geriau tinka projektams, kuriuose reikia glaudaus bendradarbiavimo ir naujovių. *Scrum* geriausiai tinka mažoms tarp funkcinėms komandoms.

Eksperimentiniu būdu šias dvi metodikas palygino M. Alkvdahas [26], A. Granulo ir A. Tanovičius [11] mokymosi valdymo sistemos projekto inicijavimo ir valdymo procese. Mokymosi valdymo sistema - programinės įrangos sistema, kuri palaiko mokymosi procesą ir palengvina bendravimą, dalijimąsi medžiaga ir informacija, realus tokio pobūdžio sistemos pavyzdys – Moodle. Sistema sukurta kaip bandomasis projektas tarp studentų universiteto kurso metu. Pagrindinis tikslas buvo palyginti ir išanalizuoti abi metodikas – *Scrum* ir *Kanban*.

Šis projektas buvo išskaidytas į 15 etapų, kiekvieno etapo pabaigoje sukuriama vienas sistemos modulis. Kiekvienas etapas išskaidomas į užduotis bei įvertinama kiekvienos užduoties numatoma atliekamo darbo apimtis. Kiekvienos iš užduočių atlikimo laikas suplanuojama *Scrum* ir *Kanban* metodikų pagrindu. Atlikti skaičiavimai parodė, jog turint to paties dydžio projekto komandą bei vienodus išteklius projektas būtų pabaigtas anksčiau naudojant *Kanban* metodiką. Pagal *Scrum* metodiką projektui užbaigti reikėtų 14 mėnesių, o *Kanban* – 12 mėnesių. Darbo autoriai išvadose teigė, jog šis skirtumas atsiranda dėl kaštų, kuriuos *Scrum* komanda skiria darbų planavimui bei kitoms *Scrum* ceremonijoms, kai tuo metu *Kanban* komanda laiką skiria darbui. Tačiau taip pat autoriai pabrėžia, jog nėra vienareikšmiško sprendimo, kuri metodika yra geresnė, kadangi tai stipriai priklauso nuo paties projekto specifikos – ar yra nurodyti tikslūs terminai, ar reikalingas detalus darbų stebėjimas bei apžvalga, kadangi tokiu atveju geresnis pasirinkimas būtų *Scrum*. *Kanban* metodika tinkamesnė tuo atveju, jei orientuojamasi į galutinį rezultatą.

Pagrindinius *Scrum* ir *Kanban* metodikų skirtumus tikslingai apibrėžia M. Alkvaldis [6] žemiau pateiktoje 1 lentelėje.

1 lentelė. *Scrum* ir *Kanban* metodikų palyginimas

Kriterijus	<i>Scrum</i>	<i>Kanban</i>
Metodo nusistatymas	Išsamiai ir skaidriai nustatytas	Nenustatytas ir pasižymi didesniu lankstumu palyginus su " <i>Scrum</i> "
Vaidmenys ir atsakomybė	Iš anksto nustatyti kiekvieno komandos nario vaidmenys ir pareigos, įskaitant produkto savininką, <i>Scrum</i> meistrą ir kitus	Nėra iš anksto nustatytų vaidmenų ir pareigų
Priėmimo laikas	Gali kilti pereinamojo laikotarpio iššūkių	Lengvesnis perėjimas, ypač kai komandos, kurios palaiko ir valdo projektus, nori pereiti nuo tradicinių metodų
Komandos dydis	Komanda, kurią sudaro 5-11 kūrimo komandos narių, turinčių įtraukiant produkto savininką ir <i>Scrum</i> meistrą. Jei komanda didelė, bus sunku deleguoti darbus bei valdyti procesą visiems <i>Agile</i> komandos nariams. Taigi, didelei komandai <i>Kanban</i> yra geriau nei <i>Scrum</i> .	<i>Kanban</i> yra lankstesnis, palyginti su <i>Scrum</i> , komandą gali sudaryti mažiau nei 5 komandos nariai, gali būti daugiau nei 11 (iki 14 komandos narių).
Atiduodamo darbo dydis	Mažiau nei vieno mėnesio sprintas laikomas dideliu, ypač kai susiduriama su nuolatiniais pokyčiais	<i>Kanban</i> neturi vienos, dviejų ar keturių savaitinių sprinto, kaip <i>Scrum</i> . Tai rodo, kad kasdien / kas valandą galima pristatyti svarbius elementus. Išipareigojimai nėra privalomas
Reikalavimų prioritetų nustatymas	Reikalavimams reikia nustatyti prioritetus pagal sprinto trukmę, kas dvi, tris ar keturias savaites	Reikalavimų prioretizavimas atliekamas nuolat, kuris gali būti kasdien / kas valandą
Atliktos funkcijos dydis	Būdingas mažas funkcijos dydis	Funkciją galima suskirstyti į mažesnes dalis, kurios gali būti

Kriterijus	<i>Scrum</i>	<i>Kanban</i>
		naudinga, kai patyrusi komanda nori greičiau grįžtamąjį ryšį
Parengiamojo darbo laikas	Skirtingai nei <i>Kanban</i> , <i>Scrum</i> vengia trumpinti parengiamojo darbo laiką	<i>Kanban</i> trumpina parengiamojo darbo laiką, nes vengia atlikti daug užduočių
Išlaidos	Skirtingai nei <i>Kanban</i> , <i>Scrum</i> vengia taupyti sąnaudas, tačiau daugiau dėmesio skiria žinioms, patirčiai ir sprendimų priėmimui remiantis tuo, kas žinoma	Lyginant su <i>Scrum</i> , daugiausia dėmesio skiriama sąnaudų mažinimui, ypač operacijoms
Kokybė	Sprinto peržiūros susitikimas yra pagrindinė praktika, skirta tobulinti kokybę taikant <i>Scrum</i> metodą	<i>Kanban</i> daugiau dėmesio skiriama kokybei gerinti, ypač palyginti su <i>Scrum</i>

1.2. „Agile“ projektų artefaktai

Šiame skyriuje identifikuojami pagrindiniai *Agile* principais grindžiamos *Scrum* metodikos artefaktai. *Scrum* artefaktai - tai galutiniai rezultatai, kuriuos projekto komanda sukuria užbaigusi sprintą.

Jie padeda užtikrinti, kad ir *Scrum* komanda, ir suinteresuotieji asmenys suprastų, kokia yra kūrimo proceso stadija. Taip pat *Scrum* artefaktai padeda komandai lengvai identifikuoti sprinto projekto metu vykstančius procesus ir jų būklę. Pagal artefaktus yra tikrinama projekto būklė, identifikuojamos rizikos kiekvienam artefaktui.

1.2.1. Produkto tikslas

Produkto tikslas (*angl. Product Goal*) apibūdina būsimą produkto būseną, kuri gali būti *Scrum* komandos planavimo tikslas. Produkto tikslas yra įtrauktas į produkto darbų sąrašą. Likusi produkto darbų sąrašo dalis sukurama siekiant apibrėžti, kas įgyvendins produkto tikslą.

1.2.2. Produkto atliekamų darbų sąrašas

Produkto darbų sąrašas (*angl. Product Backlog*) - tai naujai sudaromas, tvarkingas sąrašas to, ko reikia produktui patobulinti. Tai vienintelis *Scrum* komandos atliekamo darbo šaltinis.

Produktų sąrašo tikslinimas - tai produktų sąrašo elementų skaidymas ir tolesnis apibrėžimas į mažesnius tikslesnius elementus. Tai yra nuolatinė veikla, kurios metu pridedamos detalės, pavyzdžiui, aprašymas, eiliškumas ir dydis. Atributai dažnai skiriasi priklausomai nuo darbo srities.

Už dydžio nustatymą atsakingi programuotojai, kurie atliks darbą. Produkto savininkas gali daryti įtaką programuotojams, padėdamas jiems suprasti ir pasirinkti kompromisus.

Produkto atliekamų darbų sąrašas atitinka paruoštų darbų apibrėžimą, kai projekto darbai – naudotojo pasakojimai - atitinka šiuos kriterijus:

1. Darbas turi trumpą santrauką;
2. Darbe pateikta aiškiai suformuluota verslo vertė;

3. Priimtino kriterijai yra aiškūs ir patikrinami;
4. Prie naudotojo sąsajos istorijų yra pridėti patvirtinti dizainai;
5. Pateikti testavimo atvejai ir pavyzdiniai duomenys testavimo atvejams patikrinti;
6. Jei reikia, nustatyti veiklos kriterijai;
7. Susieti blokuojantys veiksniai ir priklausomybės;
8. Nustatyta darbo apimtis.

1.2.3. Srinto tikslas

Srinto tikslas (angl. *Sprint Goal*) yra programuotojų įsipareigojimas atlikti tam tikrus darbus, kurie priartintų komandą prie projekto tikslo. Tačiau srinto tikslas suteikia lankstumo nustatant tikslus darbus, kurių reikia jam pasiekti. Srinto tikslas taip pat užtikrina nuoseklumą, skatindamas *Scrum* komandą dirbti kartu, o ne prie atskirų veiklų.

Srinto tikslas sukuriamas per srinto planavimą ir tada įtraukiamas į srinto darbų sąrašą. Jei paaiškėja, kad darbas yra kitoks, nei jie tikėjosi, jie bendradarbiauja su produkto savininku, kad srinto metu susitartų dėl srinto darbų sąrašo apimties, nedarydami įtakos srinto tikslui.

1.2.4. Srinto atliekamų darbų sąrašas

Srinto darbų sąrašą (angl. *Sprint Backlog*) sudaro srinto tikslas, srintui atrinktų produktų sąrašo elementų rinkinys ir įgyvendinamas planas, kaip pasiekti produkto prieaugį.

Srinto darbų sąrašas yra programuotojams skirtas planas. Tai gerai matomas, realiuoju laiku matomas darbų, kuriuos programuotojai planuoja atlikti per srintą, kad pasiektų srinto tikslą, vaizdas. Todėl srinto darbų sąrašas atnaujinamas visą srintą, kai sužinoma daugiau. Jame turėtų būti pakankamai išsamiai aprašyta, kad jie galėtų patikrinti savo pažangą kasdienio *Scrum* susitikimo metu. [16].

Srinto atliekamų darbų sąrašas atitinka paruoštų darbų apibrėžimą, kai srinto suplanuoti darbai – naudotojo istorijos - atitinka šiuos kriterijus:

1. Visi srinto darbai atitinka paruoštų darbų apibrėžimą;
2. Visiems srinto darbams nustatyti prioritetai;
3. Darbai yra pakankamai maži, kad tilptų į vieną srintą;
4. Srinte nėra paslėptų darbų;
5. Pašalintos visos kliūtys;
6. Srinto darbus galima parodyti klientui;
7. Komanda peržiūrėjo visus srinto darbus;
8. Komandos nariai apskaičiavo savo pajėgumą srinto laikotarpiui;
9. Komanda sutinka su srinto apimtimi;

1.2.5. Atlikto darbo apibrėžimas

Atlikto darbo apibrėžimas (angl. *Definition of Done*) yra oficialus prieaugio būsenos, kai jis atitinka produktui keliamus kokybės reikalavimus, aprašymas. Kai projekto darbų sąrašo elementas atitinka apibrėžtį "Atlikta", sukuriama prieaugio dalis. Baigtumo apibrėžimas užtikrina skaidrumą, nes visiems suteikia bendrą supratimą apie tai, koks darbas buvo atliktas

kaip priaugio dalis. Jei produktų sąrašo elementas neatitinka „Atlikta“ apibrėžties, jis negali būti išleistas ar net pristatytas per sprinto peržiūrą. Vietoj to, jis grįžta į produktų sąrašą, kad būtų svarstomas ateityje. Jei priaugio baigtumo apibrėžtis yra organizacijos standartų dalis, visos *Scrum* komandos privalo jos laikytis kaip minimumo. Jei tai nėra organizacijos standartas, *Scrum* komanda turi sukurti produktui tinkamą baigtumo apibrėžtį.

Programuotojai privalo laikytis šio apibrėžimo. Jei prie produkto kartu dirba kelios *Scrum* komandos, jos turi tarpusavyje apibrėžti tą patį „Atlikta“ apibrėžimą ir jo laikytis [16]. Projekto komponentai atitinka atlikto darbo apibrėžimą, kai yra tenkinami žemiau nurodyti reikalavimai.

Kodas atitinka šiuos reikalavimus:

1. Kode nebeidentifikuojama klaidų;
2. Kodas yra peržiūrėtas kitų komandos narių
3. Kodas yra užkomentuotas
4. Kodas tikrinamas ir paleidžiamas pagal dabartinę sistemos versiją;
5. Kodas atitinka stiliaus standartus;
6. Kodas atitinka visus kliento nurodytus kūrimo standartus (OWASP, WCAG ir kt.).

Testai atitinka šiuos reikalavimus:

1. Vienetų testai yra parašyti ir perduoti;
2. Parašyti integraciniai testai, kurie yra sėkmingi;
3. Parašyti palaikomų įrenginių ir (arba) naršyklių naudotojo sąsajos testai (BDD testai), kurie yra sėkmingi;
4. Kodas kuriamas be klaidų;
5. Laikomasi kodo aprėpties reikalavimų;
6. Kodas dislokuojamas testavimo aplinkoje.

Naudotojo pasakojimai atitinka šiuos reikalavimus:

1. Visi nurodyti priėmimo kriterijai yra tenkinami;
2. Naudotojo pasakojimai yra parodomi ir priimti produkto savininko.

Architektūra atitinka šį reikalavimą – parengti ir (arba) atnaujinti atitinkami dokumentai ir (arba) diagramos. Dokumentacija atitinka šį reikalavimą – atnaujinti atitinkami mokymosi ir naudotojo vadovai. Sprintas atitinka šį reikalavimą – visos sprinto užduotys pristatytos ir priimtos kliento.

1.2.6. Produkto priaugis

Produkto priaugis (angl. *Product Increment*) yra konkretus žingsnis produkto tikslo link. Tam, kad suteiktų vertę, priaugis turi būti tinkamas naudoti.

Per sprintą gali būti kuriami keli priaugiai. Priaugių suma pristatoma sprinto demonstracijos peržiūroje. Tačiau padidinimas gali būti pristatytas suinteresuotosioms šalims iki sprinto pabaigos. Darbas negali būti laikomas priaugio dalimi, jei jis neatitinka baigto darbo apibrėžties.

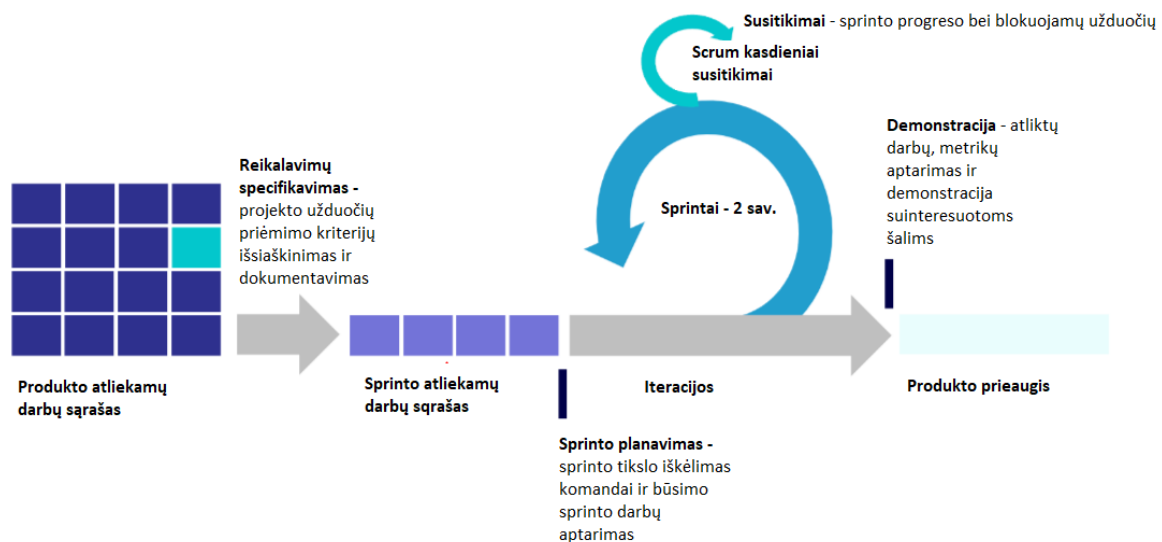
1.2.7. Sprinto eigos diagrama

Sprinto eigos diagrama (angl. *Burndown chart*) ne visada laikoma esminių *Scrum* artefaktų dalimi. Tai grafikas, rodantis, kaip greitai komanda įgyvendina naudotojų istorijas arba produkto plano elementus. Todėl sprinto eigos diagrama parodo bendras pastangas, palyginti su sprinto darbų kiekiu.

Sprinto eigos diagramos tikslas – įsitikinti, kad projektas vyksta pagal planą ir kad rezultatas atitiks lūkesčius bei bus pristatytas pagal grafiką.

Scrum komandos pažangos tempas vadinamas greičiu (angl. *velocity*) - tai per sprintą užbaigtų naudotojo istorijos punktų skaičius. Iš dalies baigtas darbas į greitį neįskaičiuojamas.

Scrum gyvavimo ciklas pateiktas 3 pav.



3 pav. *Scrum* gyvavimo ciklas Pritaikyta pagal Ettinger (2016)

1.3. Rizikų gyvavimo ciklas

Rizika gali būti apibrėžiama kaip galimi faktinių rezultatų nuokrypiai nuo laukiamų. Dažniausiai siejama su nepageidaujamo, beverčio ar žalingo rezultato, klaidos ar nuostolių tikimybe. Tai neapibrėžtumo poveikis pokyčiui, sprendimui ar įmonei. Projekto komanda bendradarbiauja su kitomis suinteresuotosiomis šalimis, kad nustatytų ir įvertintų riziką, nustatytų jos prioritetus ir spręstų šios rizikos problemas, keičiant sąlygas ar įvykius, dėl kurių kyla neapibrėžtumas: mažinti pasekmes, pašalinant rizikos šaltinį, visiškai išvengiant rizikos nepradėti ar tęsti veiklą, kuri sukelia riziką [17].

Riziką sudaro trys esminiai komponentai: neapibrėžtumas, nuostoliai ir ribota trukmė. Visada yra neapibrėžtumas, ar rizika įvyks. Kadangi rizika gali ir neįvykti, galima valdyti ir eliminuoti jos atsiradimą. Neigiama rizika dažniausiai yra susijusi su nuostoliais: laiko ar pinigų praradimu, rinkos dalies praradimu arba kitu nuostoliu, kurį projektas gali patirti, jei rizika įvyks [18].

Šiuolaikiniuose programinės įrangos projektuose saugumas ir rizikos valdymas yra ne tik tai, ką galima daryti, jei yra laiko ir išteklių. Saugumas tapo svarbia galutinio produkto dalimi. Tai reiškia, kad rizikos valdymas turi būti įdiegtas projekto pradžioje, o rizika turi būti vertinama ir nustatoma viso kūrimo ciklo metu [19].

Veiksmingas rizikos valdymas apima:

1. Rizikos nustatymą;
2. Kiekvienos rizikos analizę, siekiant nustatyti jos rimtumą;
3. Nustatytos rizikos prioritetų nustatymą pagal jos rimtumą;
4. Veiksmų planų, skirtų rizikai įveikti, kūrimą;
5. Nuolatinę stebėseną ir tolesnius veiksmus, siekiant užtikrinti, kad veiksmų planai mažina riziką.

1.3.1. Rizikos nustatymas

Rizikos nustatymas yra reikalingas tam, kad būtų galima ją kontroliuoti. Yra įvairių metodų, kurie gali būti naudojami nustatant riziką. Keletas populiariausių rizikos identifikavimo metodų yra šie: išsamūs rizikos kontroliniai sąrašai, dokumentų peržiūra, prielaidų ir apribojimų analizė. Dažniausiai su rizikos valdymu susijusius procesus inicijuoja projekto vadovas bei suteikia komandai informaciją, kaip atpažinti bei valdyti atsiradusias rizikas. Pagrindiniai rizikos nustatymo žingsniai yra šie[20]:

1. Rizikų nustatymas apžvelgiant projektui išskeltus reikalavimus. Pradiniame projekto vystymo etape komanda turėtų susipažinti su reikalavimais ir nustatyti, kokios yra galimos rizikos reikiamo sprendimo įgyvendinimo procese. Viena iš pagrindinių rizikų, įvardijama *Agile* tipo projektuose yra rizika, susijusi su neapibrėžtais ar netinkamai apibrėžtais reikalavimais [5];
2. Naudotojų istorijų įvertinimo procesuose (angl. *estimation*) komanda turėtų įvertinti darbų trukmę bei kiek pastangų tam tikra užduotis pareikalaus;
3. Artėjančio sprinto planavimo susitikimuose į sprinto atliekamų darbų sąrašą turėtų būti pridėtos užduotys su atpažintomis rizikomis bei galimais problemų sprendimo būdais rizikos pasiteisinimo atveju;
4. Kasdieniai *Scrum* susitikimai turėtų būti panaudojami rizikoms iškelti. Komanda turėtų aptarti rizikos poveikį bei tikimybę prieš imdamiesi rizikos mažinimo priemonių. Taip pat kasdieniauose susitikimuose turėtų būti aptariamas sprinto statusas, projekto būklė bei tolimesni veiksmai.

Riziką galima nustatyti visuose susitikimuose, išskyrus retrospektyvinius. Nustačius riziką, reikėtų aptarti, kokį poveikį ir tikimybę ji gali turėti. Net jei rizika greitai identifikuojama per sprinto planavimą, tai vis tiek reiškia, kad tolesnį vertinimą turėtų atlikti asmuo atsakingas už šią riziką. Nustatyti ir įvertinti naujas rizikas galima ir esamo sprinto peržiūros metu, tačiau įprastai šios rizikos bus valdomos ir kontroliuojamos kito sprinto metu [21].

1.3.2. Rizikos analizė ir valdymas

Atpažinus riziką reikalinga strategija, kaip šią riziką kontroliuoti bei valdyti. Yra metodų, kurie siūlo, kaip apibrėžti riziką įvairiems projektų tipams. Taikant tradicinius projektų valdymo

metodus, rekomenduojama sudaryti rizikos registrą, kad būtų galima stebėti, kaip valdyti ir kontroliuoti riziką [18, 29]. Šis būdas apibrėžia pagrindinius rizikos atributus, tačiau tai reikalauja viso komandos įsitraukimo. Taip pat gali būti naudojamos rizikos mažinimo diagramos (angl. *risk burndown charts*), kuriose atvaizduojama rizika, rizikos tikimybė, nuostolio dydis (prarasto laiko suma, jei rizika įvyktų) bei poveikis.

Vienas iš dažniausių *Agile* komandų naudojamų rizikos analizės ir valdymo metodų yra „Kas ir kodėl“ metodas. Komanda suorganizuoja susitikimą, kurio metu siekiama išsiaiškinti, kas galėtų įvykti projekte, po to analizuojama įvykio priežastys. Pirmuoju atveju nustatomi rizikos įvykimo padariniai, o antruoju yra identifikuojami rizikos faktoriai bei aplinkybės. Pagrindinis šio metodo privalumas yra jo paprastumas, ypač komandoms, kurios kitaip gali būti nesusipažinusios su specializuotomis rizikos valdymo praktikomis [21].

Kai kurios metodikos [21] išskiria, jog rizika gali būti valdoma įvertinus ją kaip atskirą komponentą sprinto metu, arba kaip atskirą užduotį *Kanban* lentoje, jei dirbama pagal *Kanban* modelį. Taip pat galima identifiкуoti rizikų užduotis su tam tikrais spalvų indikatoriais, padedančiais nustatyti rizikos prioritetą bei statusą. Šie indikatoriai taip pat gali būti susiejami su esamomis užduotimis.

1.3.3. Rizikos sekimas

Rizika turėtų būti sekama ir dokumentuojama viso proceso metu. Dažniausiai šioms praktikoms yra panaudojami kasdieniai *Scrum* susitikimai.

Rizikos dokumentavimas įprastai vykdomas rizikos registruose arba įvertinus riziką kaip papildomą sprinto užduotį.

Rizikos sekimas suteikia informacijos, kuri yra reikalinga siekiant nustatyti požiūrį į riziką bei tolimesnius veiksmus (rizikos vengti, priimti ar išnaudoti). Nors kai kurių rizikų valdymas ir sekimas apima tokius veiksmus kaip papildomos užduotys rizikai valdyti sprinto metu, kai kurių rizikų valdymo modeliuose reikia atsižvelgti į tam tikras projekto valdymo bei vykdymo procedūras bei jas koreguoti. Pavyzdžiui, reikalavimų neatitikimo rizikai programavimo metu komandos naudoja porinį programavimą – sprinto užduočių realizacijos kodą turi patikrinti du programuotojai.

Rizikos atžvilgiu modifikuota *Kanban* lentoje su rizikos susijusios užduotys yra žymimos spalviniais kodais, kad būtų lengviau vizualizuoti riziką. Tokią praktiką taip pat galima taikyti kitiems *Agile* artefaktams, įskaitant *Agile* istorijų žemėlapius, kuriuose aprašomas ryšys tarp funkcionalumo modulių ir juos sudarančių naudotojų istorijų. Tai leidžia vizualizuoti rizikos pasiskirstymą ir leidžia nustatyti, kur rizikos analizė galėjo būti nepakankama (pavyzdžiui, naudotojų istorijos, kuriose nėra jokios akivaizdžios teigiamos ar neigiamos rizikos).

1.4. Pagrindinės rizikos „Agile“ projektuose

Rizikos valdymas yra vienas iš pagrindinių procesų programinės įrangos kūrimo srityje. *Agile* principais grindžiamos metodikos naudoja iteracinius projekto vystymo etapus, kuriais siekiama sumažinti neapibrėžtumo riziką, nustatyti prioritetus bei juos pasitvirtinti su klientu

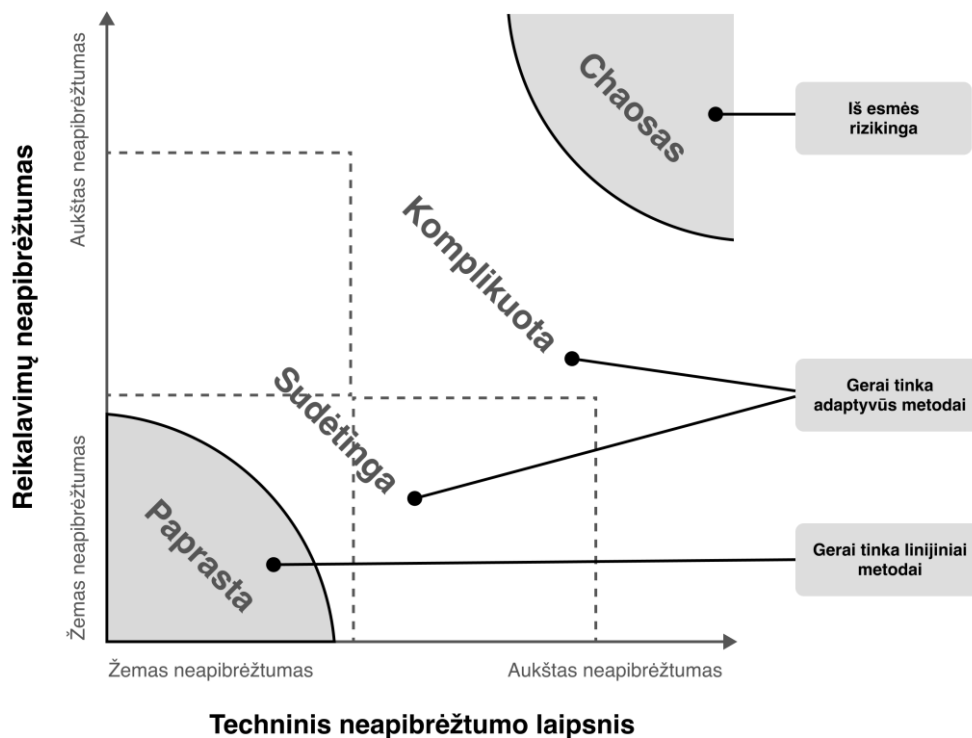
kiekvienos iteracijos metu, įtraukti klientą į produkto kūrimo eigą. Šios praktikos sumažina arba visai eliminuoja rizikas, kurios gali kilti projektą vystant tradiciniais metodais, tačiau reikia nepamiršti ir rizikų, kurios gali kilti *Agile* principais valdomuose projektuose [27, 28]. Toliau pateikiamos pagrindinės rizikos, išskiriamos pagal *Scrum* metodiką vystomuose projektuose.

1.4.1. Neapibrėžti reikalavimai

Kai kuriuose projektuose yra daug neaiškumų, susijusių su projekto reikalavimais ir tuo, kaip juos įvykdyti naudojant dabartines žinias ir technologijas. Šis neapibrėžtumas gali lemti didelį pokyčių skaičių ir projekto sudėtingumą (4 pav.).

Didėjant projekto neapibrėžtumui, didėja rizika, jog didžiąją dalį darbų teks perdaryti, taip pat atsiranda poreikis taikyti kitokią projekto valdymo metodą. Siekdamas sumažinti šių rizikų poveikį, komandos pasirenka tokius gyvavimo ciklus, kurie leidžia mažais darbo etapais spręsti projektus, kuriuose yra didelis neapibrėžtumas.

Naudodamos nedidelius etapus komandos gali patikrinti savo darbą ir keisti tolesnius veiksmus. Kai komandos dirba trumpais sprintais, jos gali greičiau ir tiksliau suprasti tikruosius kliento reikalavimus [5].



4 pav. Neapibrėžtumo ir sudėtingumo modelis pagal Stacey sudėtingumo modelį

Komandos gali nesunkiai planuoti ir valdyti projektus su aiškiais ir stabiliais reikalavimais bei aiškiais techniniais iššūkiais. Tačiau didėjant projekto neapibrėžtumui, didėja ir pakeitimų, papildomo darbo ir atlikto darbo perdarymo tikimybė, o tai brangiai kainuoja ir atima daug laiko.

Kai kurios komandos patobulino projekto gyvavimo ciklus ir taiko iteracinius metodus. Daugelis komandų pastebi, kad periodiškai aiškinantis reikalavimus ir dažniau pristatant atliktus darbus galima lengviau prisitaikyti prie pokyčių. Šie iteraciniai metodai sumažina papildomo darbo nei darbo perdarymo tikimybę, nes komandos gauna grįžtamąjį ryšį.

1.4.2. Su projekto komanda susijusios rizikos

Naudodamas eksperimentinį metodą J. J. Cho [23] straipsnyje *An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum* klasifikavo pagrindines rizikas, kylančias *Scrum* projektuose projekto inicijavimo bei vykdymo stadijose. Buvo atliktas išsamus atvejo tyrimas dviejose organizacijose, kuriose *Scrum* metodas buvo integruotas į organizacijų programinės įrangos kūrimo procesus. Viena organizacija teikia didelės apimties ir kritinės svarbos taikomąsias programas, o kita – mažos ir vidutinės apimties taikomąsias programas. Išanalizavus duomenis pateiktos pagrindinės *Scrum* projekto rizikos bei jų poveikis kiekvienai iš organizacijų.

Pasak J. J. Cho, pagrindinės su žmogiškaisiais ištekliais susijusios rizikos yra klasifikuojamos taip:

1. Komandos valdymo – organizacijoms nusprendus pereiti prie *Scrum* metodikos projekto vystymui, turi būti pertvarkoma esamų komandų struktūra. Išskiriamos rizikos, jog šioje struktūroje gali būti neatsižvelgta į komandos narių turimas žinias bei įgūdžius, išmoktą verslo logiką, programavimo įrankius bei kalbas. Šioms rizikoms eliminuoti yra reikalingas *Scrum* meistras – komandos vadovas. Nors *Scrum* ir yra save organizuojanti komanda, *Scrum* meistras matydamas bendrą poveikslą gali tinkamai paskirstyti resursus bei nukreipti komandą tinkama linkme. Šios rizikos pasitvirtino ir neigiamai paveikė didelės apimties programinę įrangą teikiančios įmonės perėjimo procesą;
2. Bendradarbiavimo – išskiriamos bendradarbiavimo problemos tarp komandos narių, kai tam tikri reikalavimų arba funkcionalumo įgyvendinimo pakeitimai iš programuotojų nebuvo perduodami testuotojams – tokiais atvejais turėtų būti betarpiškai komunikuojama su testuotojais arba testuotojams neperduodami pakeitimai testuojama savarankiškai. Išskiriamos problemos dėl komunikacijos trūkumo darbuotojo persidirbimo atveju, kai vienas asmuo atlieka kelių rolių funkcijas bei apie tai neinformuoja komandos. Tai gali kelti grėsmę darbo kokybei bei darbuotojo pasiekiamumui. Su bendradarbiavimo trūkumo problemomis bei įvykusios rizikos padariniais susidūrė abi eksperimentinės organizacijos;
3. Apmokymo – darbuotojų apmokymo problema buvo viena pagrindinių problemų didelės apimties taikomąsias programas teikiančioje organizacijoje. Dėl sistemoje įdiegto sudėtingumo nauji darbuotojai turi praleisti daug laiko susipažinti su sistema. Naujų darbuotojų apmokymo problema dar labiau paaštrėja, kai darbuotojas, turintis patirties tam tikroje srityje, dirba kitoje vietoje nei naujasis darbuotojas. Būdas kaip išvengti ir valdyti su apmokymu susijusią riziką – paskirti mentorių, apmokyti visus naujus darbuotojus pagrindinės įmonės praktikos, o kiekvienai komandai paskirti konkrečius mokymus. Apmokymo problemos pastebėtos ir mažos apimties taikomąsias programas teikiančioje organizacijoje, tačiau problemos kildavo ne dėl sistemos apmokymo trikdžių, o dėl ribotų žinių apie *Scrum* valdymo procesus bei jų svarbą. Tokiu atveju rekomenduojama, jog visi nauji programuotojai pereitų išsamius oficialius mokymus;

4. Atsakomybės stokos – ši problema pastebima didelėse projekto komandose, kai sprinto darbai yra nepabaigiami laiku ir perkeliama į kitą sprintą, kadangi komandoje nariai neprisiima atsakomybės už projekto eigą. Jei savi organizuojančios komandos veikia netinkamai, jos skatina priežiūros stoką, o tai gali lemti atskaitomybės trūkumą ir vėluojančius darbus. Su atsakomybės stokos problemomis nesusidūrė mažos apimties taikomas programos teikianti organizacija;
5. Pasitikėjimo stokos – ši problema pastebima didelėse projekto komandose tarp programuotojų bei projekto vadovo. Ši problema kilo iš bendradarbiavimo ir informacijos dalijimosi stokos tarp komandos narių ir projekto vadovo. Su pasitikėjimo stokos problemomis nesusidūrė mažos apimties taikomas programos teikianti organizacija.

Taip pat naudinga išnagrinėti, kaip šios rizikos yra klasifikuojamos ir valdomos daugiakultūrinėje organizacijoje, kurios plėtra apima ne vieną valstybę. Su projekto komanda susijusių rizikų valdymas tarptautinėse kompanijose įvertina tokius faktorius, kaip lokacija, kalbos, kultūriniai bei politiniai skirtumai [23]. Šių įmonių rizikų valdymas apima:

1. Lokalizacijos įgūdžių naudojimą siekiant patenkinti vietinius projekto komandos poreikius – šie įgūdžiai apima valdymo struktūros, darbų eigos ir programavimo platformos koregavimą ją lokalizuojant taip atitinkant vietos komandos poreikius [23] (pavyzdžiui, meniu teksto, naudotojo sąsajos komponentų vertimas, įrankių juostų perkėlimas skaitantiems iš dešinės į kairę ir pan.);
2. Partnerystės valdymo įgūdžių naudojimą, kad būtų koordinuojama daugybė komandų įveikiant prieštarigus daugiašalius santykius bei siekiant bendrų tikslų ir abipusės naudos tarp paskirstytų komandų. Šiuo įgūdžių rinkiniu siekiama pasiūlyti abiem pusėms naudingą projekto strategiją ir valdymo procesą. Įvairių suinteresuotųjų šalių išitraukimas kuriant partnerystę su jomis, padeda projekto komandai sukurti bendrą atsakomybės jausmą [22]. Dėl plačios dalyvaujančių suinteresuotųjų šalių įvairovės, yra didelė rizika, kad tik kelios suinteresuotosios šalys gali tinkamai įvertinti pagrindinį tikslą, strateginį požiūrį, hierarchinę struktūrą ir projekto verslo apribojimus [23];
3. Įvairių kultūrų valdymo įgūdžių naudojimą, kad būtų įveikti kultūriniai skirtumai. Siekiama motyvuoti projekto komandą bei padėti jiems suprasti savo užduotis ir bendrą projekto vaizdą. jų vaidmenis ir santykius globaliai pasiskirsčiusioje projekto komandoje. Taip pat apimami galimų ar kylančių konfliktų dėl kultūrų skirtumų sprendimai [23].

1.4.3. Su kūrimo procesu susijusios rizikos

Pasak J. J. Cho, pagrindinės su žmogiškaisiais ištekliais susijusios rizikos yra klasifikuojamos taip:

1. *Scrum* procesai (angl. *Scrum Framework*) – abiejų organizacijų darbuotojai teigiamai priėmė *Scrum* procesų integraciją į jų kasdienes užduotis, tačiau pagrindinė iškeliamą problema buvo per daug laiko atimantys *Scrum* susitikimai. Kasdienis *Scrum* susirinkimas truko per ilgai, o jo vertė nebuvo pakankama. Įvairūs *Scrum* susirinkimai atimdavo per daug programuotojų laiko iš programavimo, nors *Scrum* susirinkimai ir padėjo komandos nariams patikslinti kiekvieno sprinto tikslus ir pagerino produktų kokybę. Metodas, kaip valdyti komandos laiko sąnaudas susitikimams – nustatyti tikslią susitikimo apimtį bei glaustai nurodyti darbų statusus;

2. Vienetų ir integraciniai testavimai (angl. *Unit and Integration Testing*) – didelės apimties taikomas programoms teikiančioje organizacijoje pagrindinė išskiriama problema susijusi su produkto testavimu yra testuotojų darbo atsilikimas nuo programuotojų darbo sprinto metu. Ši problema buvo sprendžiama sukūrus tik testuotojų *Scrum* komandą, kuri apėmė visą kitų *Scrum* komandų sukurtą kodą. Šis sprendimas sukėlė dar vieną problemą, nes kai programuotojai padarė pakeitimą ir perdavė kodą testuotojų komandai, testuotojai kartais nežinodavo kitų sričių, kurių pokyčiai turėjo įtakos kuriamoms sistemoms – tai buvo sprendžiama dalį testavimo darbų atidavus programuotojui. Mažos apimties taikomas programoms teikiančioje organizacijoje testavimo darbu atsilikimo problema buvo sprendžiama pasitelkus papildomą įrankį vienetų testavimui (angl. *Unit tests*) – kiekvienas programuotojas gali testuoti savo kodą. Tačiau, kai programuotojai patys testavo savo kodą teko susidurti su dviem problemomis. Pirma, programuotojai paprastai mano, kad jų kodas visada veikia. Antra, programuotojai negalėjo taip kruopščiai išbandyti savo kodo kaip trečiųjų šalių testuotojai. Kita problema buvo susijusi su dideliu paveldėtu kodu, neskirtu vienetams ar integracijai, dėl ko gali tekti perrašyti paveldėtą kodą ai kitaip jį pritaikyti vienetų testavimui;
3. Kodo standartai – didelės apimties taikomas programoms teikiančioje organizacijoje dauguma programuotojų sutinka, kad turėdami formalų kodavimo standartą jie gali suprasti kitų programuotojų kodą, tačiau kai kurie nerimauja dėl to, kad per daug kodavimo standartų apsunkina užduočių įgyvendinimą. Didelė prievarta atitikti standartą gali trukdyti jų darbui, ir prideda papildomo darbo kodo dokumentacijos peržiūrai įsitikinti, ar jų kodas atitinka standartą. Mažos apimties taikomas programoms teikiančioje organizacijoje dauguma projektų vadovų manė, kad aklas kodavimo standarto laikymasis gali riboti kūrybiškumą ir sumažinti našumą;
4. Dokumentacija – abiejose organizacijose pradėjus taikyti *Scrum* metodiką pajuntamas didelis dokumentacijos, tokios kaip klasių, sekų, panaudojimo atvejų ar veiklos diagramos, sumažėjimas. Vienas pagrindinių paveiktų procesų buvo testavimas, nes testuotojai dažnai rėmėsi dokumentacija tikrinami veikimo bei alternatyviuosius scenarijus. Kita problema, susijusi su išsamios dokumentacijos trūkumu, buvo tendencija rašyti neskiriant laiko pagalvoti apie tai, kokį poveikį kodas gali turėti kitoms sistemos dalims. Dėl to padaugėjo klaidų, o tada reikėjo daug programuotojo darbo valandų joms ištaisyti. Taip pat išlieka rizika, jog žinios apie produktą yra laikomos darbuotojų atmintyje ir projektas tampa priklausomas nuo tam tikrų asmenų – darbuotojui palikus organizaciją išsinešamos iš žinios. Visų komandos narių lygiateisiškumo užtikrinimas dalijantis įgūdžiais ir žiniomis apie kuriamą sistemą nėra lengvas uždavinys, todėl dažnai ir neįgyvendinamas. Dokumentacijos kiekį reikėtų nustatyti atsižvelgiant į kūrimo kontekstą bei komandos lūkesčius;
5. Kodo peržiūra – didelės apimties taikomas programoms teikiančioje organizacijoje kodo peržiūra buvo laikomas svarbiu ir nuo produkto kūrimo neatsiejamu procesu siekiant užtikrinti produkto kokybę, todėl didelių problemų organizacijoje nebuvo užfiksuota. Mažos apimties taikomas programoms teikiančioje organizacijoje programuotojai neturėjo formalios kodo peržiūros, tačiau kartais vykdavo neformalias. Neturint formalios kodo peržiūros iškilo tam tikrų problemų, tokių kaip neatidumas kodo kokybei žinant, jog jis nebus peržiūrimas. Taip pat, programuotojai prarado galimybes pagerinti savo kodo

kokybę ir patobulinti savo kodavimo įgūdžius dėl grįžtamojo ryšio iš kitų programuotojų. Be to, buvo didelė tikimybė, kad programuotojai švaistys laiką ir pinigus bandydami iš naujo sukurti tai, kas jau sukurta. Pagrindinis šių problemų sprendimo būdas – organizuoti formalias kodo peržiūras;

6. Projekto apimties vertinimas ir planavimo pokeris (angl. *Project Estimation and Planning Poker*) – didelės apimties taikomas programoms teikiančioje organizacijoje problemų, susijusių su projekto apimties vertinimu neiškilo. Mažos apimties taikomas programoms teikiančioje organizacijoje programuotojams buvo sunku įvertinti projekto trukmę ir kompleksiskumą. Problema buvo sprendžiama planavimo pokeriu – projekto vertinimo metodu, kurio metu pagal pasirinktą skalę yra vertinama užduoties kompleksiskumo lygis. Šis metodas suteikia visiems komandos nariams galimybę balsuoti lygiaverčiai bei išsakyti savo nuogastavimus.

Su kūrimo procesu susijusių rizikų valdymas tarptautinėse kompanijose apima [23]:

1. Modulinio požiūrio taikymą pakartotinai panaudoti sukurtus komponentus, kad būtų išvengta pasikartojančių užduočių kelių komandų projekte. Programinės įrangos kūrimo vadovuose aptariamos modulinis metodas, leidžiantis surinkti ir suderinti skirtingas programinės įrangos dalis. Šis požiūris padeda sumažinti riziką, susijusią su besikeičiančiais naudotojo ar kliento reikalavimais. Modulinis požiūris suteikia galimybę greitai ir efektyviai perprojektuoti produktą pasikeitus reikalavimams. Modulinis požiūris yra aktualesnis globalių sistemų diegimo kontekste, kai sistemos turi būti perkomplektuojamos, kad atitiktų vietinius specifinius naudotojų reikalavimus;
2. Decentralizuotą žinių valdymo naudojimą. Decentralizuotas žinių valdymas didina techninio įgyvendinimo, produkto bei procesų koordinavimo įgūdžius. Decentralizacija suteikia lankstumo atskiroms komandos nariams papildyti vietinę žinių bazę savo žiniomis apie kintančias užduotis, kylančias problemas. Dėl to atsiranda mažiau klaidų, tikslesnių ir ekonomiškai efektyvių projekto sprendimų. Be to, decentralizuotas žinių valdymo metodas gerina transakcinę atmintį (t. y. komandos žinias apie kitų komandos narių žinias);
3. Lanksčių užduočių planavimo naudojimą, kuriuo atsižvelgiama į atskirų komandų interesus, kompetenciją ir tikslus. Atsižvelgiant į didelį vidinį neapibrėžtumą ir išorinės aplinkos elementus, projekto planas negali būti galutinis pačioje projekto pradžioje, todėl yra reikalingas lankstus užduočių planavimas. Lankstus užduočių planavimas gali būti taikomas siekiant valdyti dinaminę riziką, susijusią su užduočių planavimu.

1.4.4. Aplinkos veiksniai

Pasak J. J. Cho, pagrindinės su aplinkos veiksniais susijusios rizikos yra klasifikuojamos taip:

1. Kliento įsitraukimas – pagrindinės problemos susijusios su kliento įsitraukimu pastebimos tada, kai kliento lokacija yra toli nuo programuotojų komandos. Kai kliento atstovus nuo programavimo komandos skiria laiko juostos, klientas negali dalyvauti kasdienių sprendimų priėmime bei nėra tiek įtraukiamas į projekto vystymo eigą. Taip pat akcentuojamas kliento nenoras įsitraukti į kasdienes projekto veiklas dėl kitų prioritetų. Prastas klientų dalyvavimas projektuose kėlė problemų įmonei, nes projekto vystymo komandai reikėjo kurti konkretų funkcionalumą nesikalbant su klientais, kas užtrukdavo

papildomai laiko ir lėtindavo produkto kūrimą. Taip pat atsiranda produkto perdarymo rizika dėl kliento nepasitenkinimo galutiniu produktu;

2. Darbo aplinka – komanda buvo nevienareikšmiškai nusiteikusi dėl atviros darbo aplinkos. Kai kuriems programuotojams patiko kabinetų aplinka, kadangi tai padidino komandos narių bendravimo skaičių ir skatino bendradarbiavimą bei komandinį darbą. Tačiau daugumai komandos narių nepatiko atviros erdvės darbo aplinka, nes jie negalėjo susikcentruoti į darbą taip pat įvardinama, kad tai mažina produktyvumą dėl „foninio triukšmo“ darbo aplinkoje;
3. Modulių nepriklausomumas – sistemai didėjant ir sudėtingėjant augo projekto apimtis, didėjo priklausomybė ir tarpusavio ryšiai tarp skirtingų projekto užduočių. Tačiau kūrėjai negalėjo iki galo atsižvelgti į visas priklausomybes ir modulių tarpusavio ryšius, nes jų planavimas ir projektavimas buvo siaurai orientuotas sprinto planavimo susitikime. Programuotojai taip pat buvo linkę viską atlikti greitai negalvodami, ar kodas bus pakankamai lankstus, kad būtų galima jį naudoti ateityje. Šiai rizikai sumažinti ir eliminuoti rekomenduojama, jog visos organizacijos turėtų remti ir skatinti programuotojus skirti daugiau laiko modulių priklausomybėms ir tarpusavio ryšiams apsvastyti;
4. Bendri įrankiai bei problemos tarp skirtingų projektų komandų – kyla rizika, kad vienos *Scrum* komandos nariai gali praleisti daug valandų ieškodami tinkamų įrankių ar technologijų, tinkamų jų projektui, nežinodami, kad kitos *Scrum* komandos jau naudoja panašias priemones ar technologijas. Tai didelis programuotojų laiko švaistymas jei dvi *Scrum* komandos gali naudoti tas pačias arba panašias priemones ar technologijas. Rekomenduojama, jog įmonė paskirtų asmenį, kuris informuotų komandas, jei yra panašios technologijos, kuriomis jau pasinaudojo kitos komandos, ir ar yra panašios problemos, su kuriomis susidūrė kitos komandos ir kurias jos sėkmingai išsprendė;
5. Socialinė atskirtis – organizacijose nustatyta socialinio palaidumo problema, nes įmonės neturėjo būdų, kaip įvertinti individualią veiklą – komanda vertinama kaip vienetą. Tai gali sumažinti darbuotojų motyvaciją. Jei įmonė nenumato būdo, kaip tiksliai įvertinti individualų veiklos rezultatą ir vertina tik padalinio arba komandos pasiekimus, gali atsirasti socialinė atskirtis. Tam pašalinti, kiekvieno komandos nario indėlis turi būti įvertinamas atskirai. Siūloma idėja pakviesti visus komandos narius į sprinto peržiūros susitikimą, kuriame programuotojai pristatytų, ką jie įgyvendino.

Su aplinka susijusių rizikų valdymas tarptautinėse kompanijose apima [23]:

1. Analitinius įgūdžius, leidžiančius pajusti rinkos ir konkurencinius pokyčius, apimančius turimos informacijos apie vietinių ir pasaulinių konkurentų elgseną, pvz. strategijos inicijavimą, naujų produktų kūrimą ir naujų rinkodaros skatinimo priemones;
2. Tiekėjų valdymą siekiant sukurti strateginius tinklus su stipriais pramonės atstovais. Tai leidžia projekto komandai reaguoti į konkurentų elgseną. Tai taip pat gali padėti sušvelninti dinaminę riziką, susijusią su produkto pozicionavimu, ir sutrumpinti patekimo į rinką laiką;
3. Centralizuotą avarijos padarinių šalinimo praktiką, taikant iš anksto nustatytą politiką ir atsarginių kopijų sistemas. Tai leidžia sumažinti dinaminę riziką, susijusią su projekto tęstinumu.

1.4.5. Taikomos technologijos

Pasak J. J. Cho, pagrindinės technologijomis susijusios rizikos yra klasifikuojamos taip [22]:

1. Komunikacija - neefektyvus bendravimas yra daugumos programinės įrangos kūrimo projektų nesėkmių priežastis. Pagrindinės komunikacijos priemonės abiejose eksperimentinėse organizacijose buvo vaizdo konferencijų sistemos, telefoninių konferencijų sistemos, internetinių demonstracijų sistema, žinučių sistema ir elektroninio pašto sistema. Taip pat buvo naudojama virtualaus privataus tinklo sistema. Tačiau nė viena iš sistemų neveikė taip optimaliai, kaip tiesioginis susitikimas. Būtent tokį bendravimo būdą ir skatina *Scrum* metodika kasdienių susitikimų principu;
2. Žinių dalijimasis – abiejose organizacijose pastebima, jog naujai įdarbinti programuotojai darė daugiau klaidų, nes jie neturėjo tinkamų žinių apie programinę įrangą, su kuria dirbo. Kai kurios programos dalys buvo labai jautrios pakeitimams, todėl nedidelis šių dalių pakeitimas turėjo daug įtakos kitoms sritims. Svarbu turėti gerai struktūrizuotas informacijos ir žinių dalijimosi sistemas tarp patyrusių programinės įrangos kūrėjų ir visiškai naujų programinės įrangos kūrėjų;
3. Klaidų sekimo procesas – pagrindinė problema, su kuria susidūrė projektų komandos yra nevieningas įrankis sprinto eigos sekimui bei klaidų registravimui. Naudojant skirtingas platformas šiems procesams sekti, gali kilti rizika, jog užfiksuota klaida nebus laiku adresuota. Tam spręsti rekomenduojama naudoti vieną platformą šiems procesams apjungti (pvz. „Jira“), kuri galėtų palaikyti abiejų procesų eigą ir užtikrintų sekimą;
4. Versijų sekimo procesas – tinkamam produkto versijų sekimui rekomenduojama naudoti specialius įrankius, suteikiančius galimybę atsekti produkto istoriją bei produkto pokyčius kiekvienos versijos leidybos metu.

Su taikomomis technologijomis susijusių rizikų valdymas tarptautinėse kompanijose apima [23]:

1. Laisvai centralizuotos IT strategijos naudojimą, kuri padeda sumažinti pasipriešinimą pokyčiams ir užtikrinti sklandų technologijų diegimą. Laisvai centralizuota strategija išnaudoja vietines galimybes organizacijos ribose. Laisvai centralizuota strategija taip pat svarbi, nes griežta (arba griežtai centralizuota) IT strategija greičiausiai nesugebės išspręsti vietinių verslo prioritetų, todėl sukelia vidinį pasipriešinimą tarp kelių vietovių grupių arba narių. Laisvai centralizuota strategija skatina dalijimąsi organizacijos geriausiomis praktikomis;
2. Standartizuotų IT platformų naudojimą nevienodumo problemos daugelyje komandų sumažinimui. Standartizuota IT platforma leidžia organizacijoms eksperimentuoti su įvairiomis technologinėmis alternatyvomis ir taip padeda organizacijoms išbandyti įvairias konfigūracijas mažomis sąnaudomis ir per trumpą laiką;
3. Bendradarbiavimo struktūras palengvinančias komunikaciją tarp skirtingų komandų. Technologijomis paremtas bendravimas ir bendradarbiavimo infrastruktūra tampa labai svarbi, nes užduotys planuojamos ir keičiamos atsižvelgiant į vienos komandos atsiliepimus kitai komandai. Bendradarbiavimo procesai yra dar labiau sustiprinami naudojant elektroninius bendravimo kanalus, pavyzdžiui, greitųjų žinučių programas ir grupines bendravimo programas (pvz., „SharePoint“).

1.5. Rizikos valdymas „Agile“ projektuose ir komandos reakcija

1.5.1. Pagrindinės rizikos valdymo metodikos

Vis dažniau kylantį klausimą – kaip atpažinti, analizuoti ir tinkamai kontroliuoti rizikas *Agile* tipo projektuose kelia vis daugiau projektų komandų. Ieškoma tinkamo būdo kaip efektyviai reaguoti į galimas projekto grėsmes. Toliau aprašomos kelios pagrindinės metodikos, kuriomis rekomenduojama vadovautis vertinant ir kontroliuojant rizikas *Agile* metodikos *Scrum* projektuose.

B. G. Tavaresas, C. E. Sančasas ir A. D. de Souza [9] pristatė rizikos valdymo metodikas bei jų populiarumą projektų komandose. Metodikos yra pagrindžiamos eksperimentiniu metodu įvertinant 10 *Scrum* projektų komandų. Penki projektai buvo sukurti vidutinio dydžio bendrovėms, o kiti penki – didelėms bendrovėms. Projektų *Scrum* meistrai atliko NTCP klausimyną, kurį sudaro keturios analizuojamos sritys – Naujumas (angl. *Novelty*), Technologijos (angl. *Technology*), Kompleksiškumas (angl. *Complexity*) ir Tempas (angl. *Pace*). Šios klasifikacijos leidžia nustatyti tyrimo apribojimus bei tikslingai palyginti projektus, nes taikant tą patį tyrimo protokolą kitoje aplinkoje gali būti gauti kitokie rezultatai. Pagal klausimyno rezultatus identifikuojamos pagrindinės rizikų valdymo praktikos kiekviename iš projektų. Populiariausia praktika – nuolatinis grįžtamasis ryšys leidžiantis laiku įvertinti nukrypimus bei reikalavimų ar kūrimo neatitikimus ir kiek galima labiau sumažinti kaštus. Šiam tikslui naudojamos *Scrum* ceremonijos, kur daug dėmesio skiriama grįžtamajam ryšiui susirinkti (demonstracijos, retrospektyvos, sprinto planavimas). Antroje vietoje esanti praktika susijusi su pakartotiniu rizikos žinių arba išmoktų pamokų panaudojimu. Trečioji - tai praktika, susijusi su organizacijos valdymo ir individualios elgsenos pokyčiais. Individuali elgsena yra esminis reikalavimas siekiant veiksmingai valdyti riziką *Scrum* projektuose. Mažiausiai priimtina rizikos valdymo praktika yra rizikos valdymo perdavimas į *Scrum* meistro rankas tik ją identifikavus. Taip pat pabrėžiama, jog *Scrum* projektuose pasigendama formalaus rizikų valdymo proceso. Keliamą problema dėl rizikos dokumentavimo naudojant scenarijus, o ne formalius dokumentus – ši problema kyla iš požiūrio *Scrum* metodikoje kuo labiau minimizuoti dokumentacijos kieki.

E. E. Odzalis, D. Gryras ir D. Stiuartas[4] pristatė rizikos valdymo metodą pasitelkiant dirbtinio intelekto įrankį – programinės įrangos agentus (angl. *Software agents*) optimaliam rizikos identifikavimui bei kontrolei. Šis įrankis konstruojamas tam, jog rizika būtų suvaldoma nepakenkiant *Agile* projektų lankstumui bei būtų palengvinamas sprendimų priėmimo procesas. Įrankio modelis yra naudojamas projekto duomenų įvedimui, kuriuos sudaro rizikos rūšis ir rizikos rodikliai, taip pat aplinkos duomenys, kurie gali būti naudojami rizikai nustatyti. Rizikos rodikliai ir taisyklės yra susijusios su iš anksto nustatytais rizikos veiksniais, kuriuos komanda parengė ankstyvuojų projekto vystymo etapu ir yra užkoduojamos kaip taisyklės. Sprinto eigoje įrankis valdo riziką, remdamasis anksčiau apibrėžtais įvesties duomenimis. Pagrindinis privalumas yra tas, jog šis rizikos procesas yra autonominis. Suveikus bet kokiai rizikai, rizikos duomenys rodomi rizikos registre. Tačiau, autorių teigimu, sprendimas visiškai kompiuterizuoti rizikos valdymą iš savęs yra rizikingas, todėl yra reikalinga *Scrum* meistro ir komandos priežiūra. Taip pat, įrankis yra tinkamiausias rizikos identifikacijos žingsniui, o šio

proceso valdymą ir kontrolę rekomenduojama perduoti *Scrum* meistriui. Taip pat, išlieka grėsmė, jog su žmogiškuoju faktoriumi susijusios rizikos nebus užfiksuojamos.

M. Singas ir R. Saxena [18] pasiūlė riziką fiksuoti rizikos registruose. Šis metodas gali būti pritaikomas ne tik *Scrum*, bet ir kitose projektų valdymo metodikose. Rizikos registro pagrindiniai komponentai yra šie: rizikos aprašymas, rizikos nustatymo data, tikimybė, poveikis, prioritetas, atsakingas asmuo, veiksmas, kaip kontroliuoti riziką bei rizikos statusas. Registrą rekomenduojama peržiūrėti kiekvienos sprinto ceremonijos metu – registro vertės koreguojamos atsižvelgiant į esamas ir naujai atsiradusias rizikas. Taip pat siūloma naudoti rizikos pašalinimo tempo diagramas (analogiškai plano atlikimo diagramoms – vienam iš *Scrum* artefaktų). Tačiau nerimaujama, jog šis metodas gali kainuoti daug komandos laiko, kadangi reikalingas pastovus rizikos registro sekimas tam, kad šis metodas būtų veiksmingas. Skirtingos rizikų valdymo metodikos *Scrum* projektuose palyginamos 2 lentelėje.

2 lentelė. *Scrum* rizikų valdymo metodikų palyginimas

Metodika	Privalumai	Trūkumai
Nuolatinis grįžtamasis ryšys iš kliento – <i>Scrum</i> ceremonijos	Leidžia tinkamai įvertinti nukrypimus bei reikalavimų ar kūrimo neatitikimus	Reikalauja didelio kliento įsitraukimo (ne visuose projektuose tai lengvai pasiekama), rizika gali būti nustatyta per vėlai
Pakartotinis rizikos žinių arba išmoktų pamokų panaudojimas	Egzistuojančios rizikos yra lengvai užfiksuojamos be didelių laiko sąnaudų	Naujos rizikos gali būti praleistos ir neužfiksuotos
Programinės įrangos agentai	Automatizuojamas rizikos valdymo procesas – mažai laiko sąnaudų rizikų kontrolei	Dėl žmogaus neįsitraukimo naujos rizikos gali būti praleistos ir neužfiksuotos
Rizikos registras	Maža tikimybė, jog naujos rizikos bus praleistos, nes tam skiriamas didelis dėmesys	Nesutarimai dėl perteklinės dokumentacijos Tam, kad procesas vyktų sėkmingai, reikalaujama daug laiko sąnaudų

3 lentelėje pateikiama rizikų valdymo metodikų palyginimas pagal iškeliamus kriterijus.

3 lentelė. Rizikų valdymo metodikų palyginimas pagal kriterijų

Kriterijus	Metodika			
	<i>Scrum</i> ceremonijos	Pakartotinis išmoktų pamokų panaudojimas	Programinės įrangos agentai	Rizikos registras
Automatizuotas rizikų identifikavimas	-	-	+	-
Rizikos objektų agregavimas	-	+	+	-
Informacijos auditas	-	+	+	+
Rizikos parametrų nustatymas	+	+	+	+

Kriterijus	Metodika			
	<i>Scrum</i> ceremonijos	Pakartotinis išmoktų pamokų panaudojimas	Programinės įrangos agentai	Rizikos registras
Rizikos parametrų atnaujinimas	-	-	-	+
Rizikos tipų pridėjimas	+	-	+	+
Komandos įtraukimas	+	+	-	+

1.5.2. Komandos reakcija į rizikos valdymą

Agile projektuose rizikos kontrolei šiuo metu nėra skiriamas didelis dėmesys, kadangi daugelis projekto vystytojų grindžia, jog metodika iš savęs eliminuoja rizikas. M. Tomanek ir J. Juricek rašto darbe „Project risk management model based on Prince2 and *Scrum* frameworks“ [19] atliko apklausą, koks dėmesys yra skiriamas rizikai atpažinti ir valdyti *Scrum* projektuose.

Apklausa buvo atlikta Čekijoje ir Slovakijoje su 64 dalyviais, apklausoje buvo akcentuojami šie rizikos atributai: rizikos poveikis, rizikos mažinimo priemonės, rizikos tikimybė, už rizikos valdymą atsakingas asmuo, rizikos tipas, finansinė įtaka bei rizikos rezultatas. Pagrindiniai išskirti rizikos faktoriai, kurie atkreipia komandos dėmesį yra rizikos poveikis, mažinimo priemonės bei tikimybė. Taip pat iš apklausoje dalyvavusių apklaustųjų, 67% teigia, jog rizika yra peržiūrima ir apžvelgiama kiekvieną savaitę. 87% iš apklaustųjų nurodė, jog už rizikos valdymą yra atsakingas vienas žmogus – projekto vadovas.

Apklausoje taip pat buvo užduota, kas komandoje identifikuoja riziką ir 80% teigia, jog tai yra komandos nariai, o 67% - jog tai yra projekto vadovas. 73% respondentų naudoja rizikos registrą rizikai dokumentuoti ir tvarkyti. Likusieji naudoja kitus metodus arba visai nedokumentuoja rizikos. Taip pat buvo identifiukuota, kada skiriamas laikas rizikai identifiukuoti ir valdyti – didžioji dalis respondentų teigė, jog tam jie naudoja sprinto planavimo susitikimus.

Paskutinis klausimas buvo suformuluotas taip: „Ar manote, kad *Agile* metodikos turėtų būti praturtinti projektų valdymo sistemos rizikos valdymo metodais?“ - dauguma respondentų (60%) mano, kad praturtinimas būtų naudingas, o likusieji mano, kad toks sprendimas nėra naudingas.

1.6. Siekiamo sprendimo apibrėžimas

Vis aktyviau naudojant *Agile* metodikas projektų valdymui ieškoma būdų, kaip efektyviai valdyti bei kontroliuoti rizikas. Nors ir yra nemažai teigiančių, jog *Agile* metodikos *Scrum* principai iš esmės eliminuoja riziką ir sumažina ją iki vieno mėnesio (maksimali rekomenduojama sprinto trukmė) kaštų, tačiau atsiranda vis daugiau metodikų, kaip papildomai vykdyti rizikos valdymo procesą.

Egzistuojančios metodikos pagrindinį dėmesį telkia į rizikas, kurios gali būti apibrėžiamos projekto inicijavimo etape, tačiau pasigendama rizikų kontrolės vėlesnėse projekto fazėse bei detalesnio proceso kiekvienam projekto artefaktų – *Scrum* procesas aprašomas kaip nedalomas vienetas.

Siekiamo sprendimo apibrėžimas – rizikų klasifikavimas pagal atskirus projekto artefaktus, atkreipiant dėmesį į galimas rizikas kiekviename iš projekto vykdymo etapų. Taip pat siekiama apibrėžti metodiką rizikų analizei bei kontrolei kiekvienam iš artefaktų.

1.7. Analizės išvados

Lankstūs programinės įrangos kūrimo metodai plačiai taikomi įvairiose pramonės šakose ir įvairiose organizacijose, o *Agile* filosofija paremti metodai įgauna vis didesnį populiarumą. Analizės dalyje apibrėžiama *Agile* filosofija bei pagrindinės, dažniausiai naudojamos metodikos – *Kanban* ir *Scrum* bei pagrindiniai rizikų valdymo procesai, pritaikyti šioms metodikoms. *Kanban* metodikoje išskiriamas rizikos kontroliavimo procesas rizikas vertinant kaip papildomas užduotis *Kanban* lentoje ar kuriant specializuotas rizikų lentas. *Scrum* metodikoje skatinama naudotis kasdienėmis ceremonijomis rizikų identifikavimui ir sekimui. Atlikus šių metodikų palyginimą, išvada lieka tokia, jog sprendimas, kurią metodiką pasirinkti priklauso nuo paties projekto specifikos – ar yra nurodyti tikslūs terminai, ar reikalingas detalus darbų stebėjimas bei apžvalga, kadangi tokiu atveju geresnis pasirinkimas būtų *Scrum*. *Kanban* metodika tinkamesnė tuo atveju, jei orientuojamasi į galutinį rezultatą. Taip pat *Kanban* dažnai naudojama priežiūros projektams vykdyti.

Apibrėžtos pagrindinės *Agile* rizikų rūšys, suskirstytos į 4 kategorijas: su žmonėmis susijusios rizikos, su procesais susijusios rizikos, aplinkos rizikos ir su technologijomis susijusios rizikos. Šių rizikų poveikis yra palyginamas dviejose kompanijose – didelės ir mažos apimties taikomąsias sistemas teikiančiose organizacijose. Taip pat pateikiamos pagrindinės rizikų suvaldymo metodikos globaliose kompanijose. Išryškinami pagrindiniai kontrastai skirtingo profilio kompanijose, tokie kaip atsakomybės stoka (didelėje kompanijoje) ar daugialypė atsakomybė (mažoje kompanijoje), naujų komandos narių įdarbinimo bei apmokymo ypatumai, įmonėse vykdomi procesai, tokie kaip testavimas, kodo peržiūra, dokumentacija bei skirtingas su tuo susijusių problemų spektras. Taip pat apžvelgiamos su komunikacija, žinių dalijimusi bei technologine puse susijusios rizikos.

Aprašytos egzistuojančios rizikų valdymo metodikos, pritaikytos *Scrum* projektams, tokios kaip rizikų registras (gana lankstus įrankis, kuris gali būti panaudojamas ne tik *Agile* tipo projektų rizikoms sekti), grįžtamasis ryšys, kuris yra labai akcentuojamas *Scrum* metodikoje, pakartotinis žinių pritaikymas. Taip pat aprašytas rizikos identifikavimą ir kontrolę automatizuojantis įrankis pasitelkiant programinės įrangos agentus. Tačiau šios metodikos sulaukia prieštaringų reakcijų, kadangi rizikos valdymo procesas nėra formalizuojamas ar kitaip dokumentuojamas – *Agile* filosofijoje skatinama atsisakyti perteklinės dokumentacijos, tačiau yra teigiančių, jog rizikoms valdyti dokumentacija būtų naudingas įrankis. Naudojant automatizuotus sprendimus keliamos kitos rizikos, jog sumažinus žmogaus įsikišimą, kai kurios išankstinėje projekto fazėje neidentifikuotos problemos gali būti laiku nepastebėtos ir nesukontroliuotos.

Analizės išvados:

1. Išanalizavus *Agile* principais grindžiamas metodikas nustatyta, kad pagrindiniai skirtumai tarp *Scrum* ir *Kanban* metodikų yra rolės ir atsakomybės, dėmesys rizikoms bei

pritaikomumas. *Kanban* turi formalizuotą rizikų valdymo metodiką, todėl rašto darbe pagrindinis dėmesys skiriamas *Scrum* metodikos rizikų valdymui;

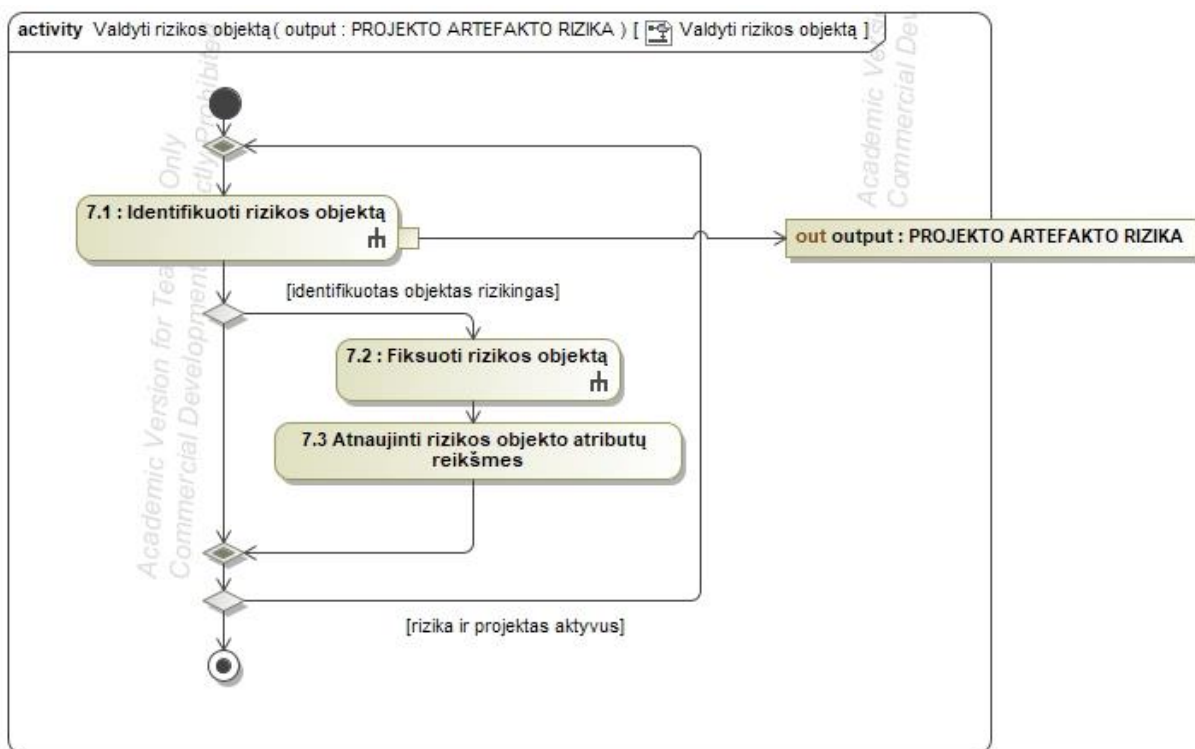
2. Išanalizavus *Scrum* artefaktus nustatyta, kuriuos artefaktus yra tikslinga naudoti rizikos valdymo metodikoje siekiant efektyvesnių rezultatų: produkto darbų sąrašas, sprinto darbų sąrašas ir pan;
3. Išanalizavus rizikų gyvavimo ciklą nustatyta, kad pagrindinis dėmesys yra teikiamas rizikos identifikavimo procesui, o tolimesnis rizikų gyvavimo ciklas nėra stebimas;
4. Išanalizavus rizikų valdymo metodikas nustatyti palyginami esamų metodikų pagrindiniai privalumai ir trūkumai. Pagrindiniai esamų metodikų trūkumai: dėl žmogaus neįsitraukimo naujos rizikos gali būti praleistos ir neužfiksuotos, nesutarimai dėl perteklinės dokumentacijos bei tam, kad procesas vyktų sėkmingai, reikalaujama daug laiko sąnaudų.

2. Rizikų valdymo metodikos „Agile“ projektuose aprašas

Šiame skyriuje aprašoma rizikų valdymo metodika, reikalinga efektyviam rizikų valdymo procesui vykdyti. Metodikos struktūrą sudaro dvi dalys – konfigūracija bei darbas su sistema ir konkrečiais projekto objektais. Konfigūracijos metu paruošiamas projektas, kuriame yra apjungiamos projekto sudedamosios dalys – artefaktai – su rizikomis, kurios yra labiausiai tikėtinos būtent tam projekto artefaktui. Taip yra paruošiama aplinka rizikų identifikavimui, analizei bei sekimui vykdyti. Toliau yra vykdomas darbas su sistema – konkrečiuose projekto artefaktuose identifikuoti rizikos objektai yra užfiksuojami ir sekami.

2.1. Rizikų valdymo sistemos reikalavimai ir panaudojimo atvejai

Rizikų valdymo metodika apima visą rizikos gyvavimo ciklą – rizikos identifikavimą (detalizuota 2.2.18 skyrelyje), analizę (detalizuota 2.2.19 skyrelyje) bei sekimą (detalizuota 2.2.20 skyrelyje). Rizikų gyvavimo ciklo procesas pateikiamas 5 pav.



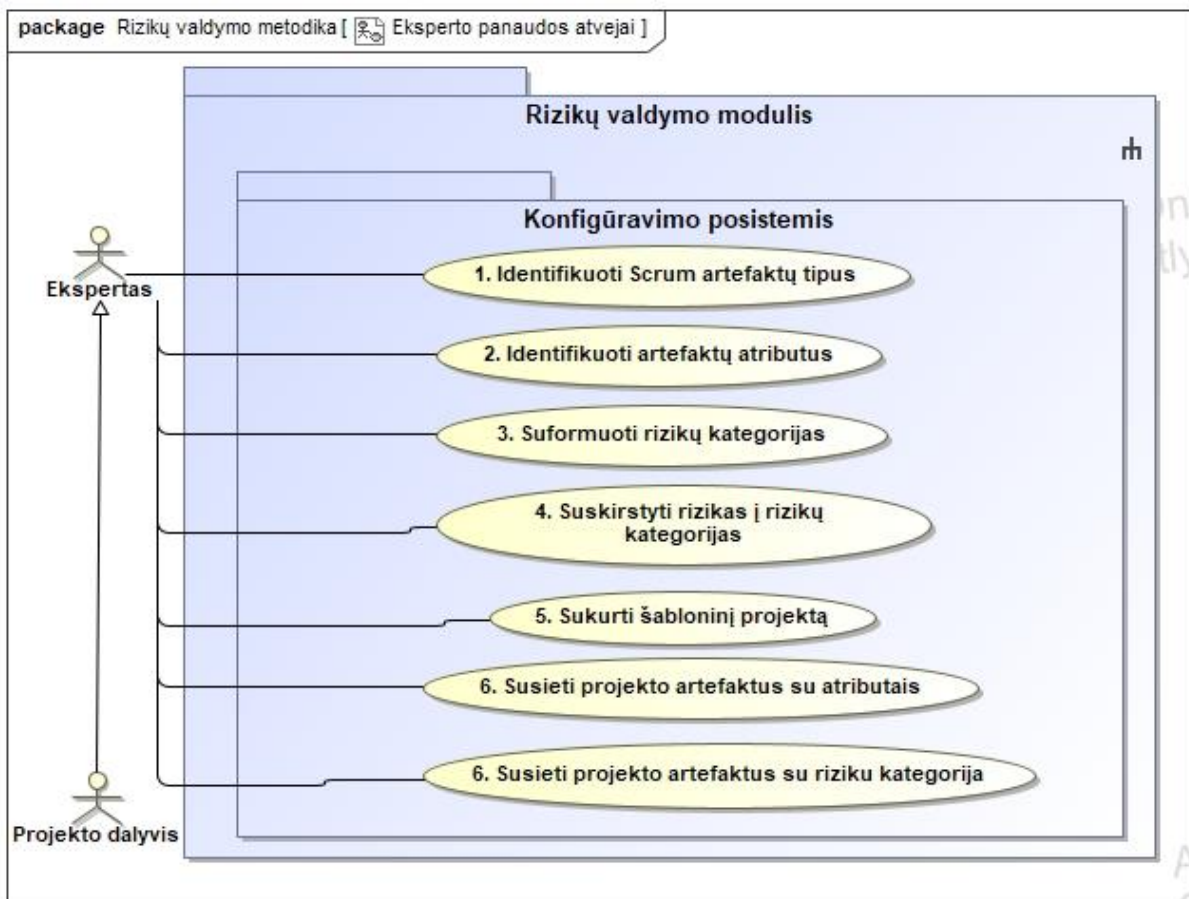
5 pav. Rizikos gyvavimo ciklo procesas

Efektyvus naujų rizikų identifikavimas yra apibrėžiamas kaip minimalus laiku neidentifikuotų rizikų skaičius. Vienas iš pagrindinių metodikos tikslų yra kuo labiau sumažinti žmogiškųjų resursų indėlį vykdant rizikų analizę ir valdymą– tiek komandos, tiek užsakovo.

Rizikų valdymo metodika *Scrum* projektuose rizikų dalykinės srities ekspertui apima šiuos panaudojimo atvejus (6 pav.):

1. PA1. Identifikuoti *Scrum* artefaktų tipus - nustatomi pagrindiniai *Scrum* projektų artefaktai, kurių būseną turi būti sekama siekiant įvertinti, ar šis artefaktas yra rizikingas;

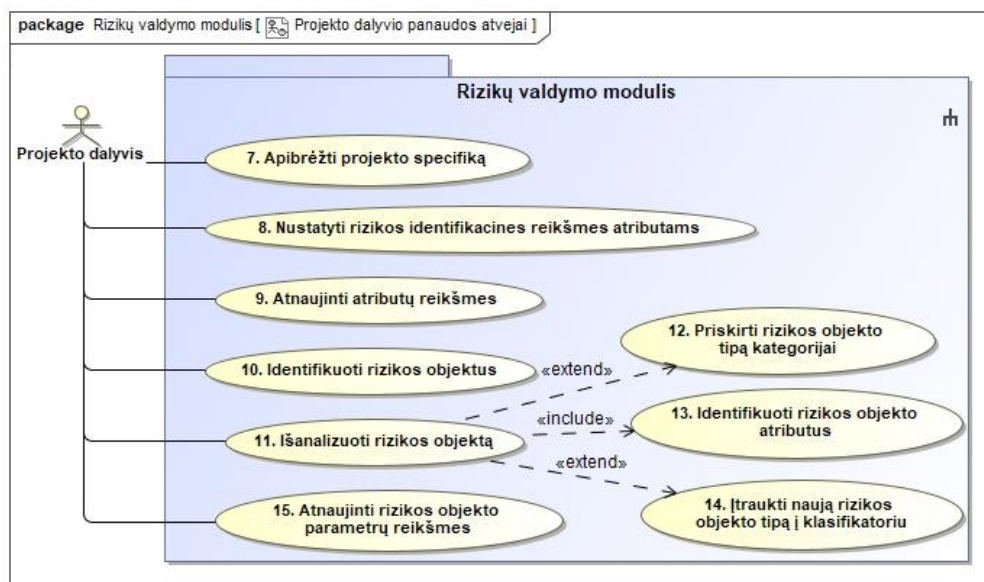
2. PA2 Identifikuoti artefaktų atributus – nustatomi pagrindiniai *Scrum* projektų atributai, kurie yra priskiriami konkrečiam artefaktui. Pagal atributo reikšmę yra sekama artefakto būsena ir rizikingumas (skyrelis 2.2.3);
3. PA3 Suformuoti rizikų kategorijas – tam, kad rizika būtų identifikuojama be didelių papildomų kaštų, reikia apibrėžti pagrindines rizikų kategorijas, pagal kurias bus klasifikuojamos rizikos (skyrelis 2.2.4);
4. PA4 Suskirstyti rizikas į rizikų kategorijas – nustatomos pagrindinės rizikos *Scrum* projektuose, kurios yra suklasifikuojamos į PA3 panaudojimo atvejuje sudarytas kategorijas (skyrelis 2.2.5);
5. PA5 Sukurti šabloninį projektą – sukuriamas projektas, nesusietas su jokių konkrečiu domenu, kuriame vėliau yra sukuriami ryšiai tarp projekto artefaktų bei rizikų kategorijų (skyrelis 2.2.6);
6. PA6 Susieti projekto artefaktus su atributais – kiekvienam artefaktui priskiriamas bent vienas atributas efektyviam rizikų sekimui vykdyti pagal artefakto atributo reikšmes bei jų pokytį;
7. PA7 Susieti projekto artefaktus su rizikų kategorija – kiekviena rizikos kategorija turi būti priskiriama tam tikram (vienam ar keliems) projekto artefaktui, kuriam yra būdingas šios kategorijos rizikų atsiradimas (skyrelis 2.2.8).



6 pav. Rizikų dalykinės srities eksperto metodikos panaudos atvejai

Rizikų valdymo metodika *Scrum* projektuose rizikų dalykinės srities ekspertui apima šiuos panaudojimo atvejus (7 pav.):

1. PA7 Apibrėžti projekto specifiką – sukuriamas projektas, kuriame nurodomos konkretaus projekto, su kuriuo dirbama, atributai, tokie kaip tikslas, kaina, apimtis, trukmė. Šie atributai nurodo, kokias savybes ir kokius apribojimus iš užsakovo turi projektas. Šis panaudojimo atvejis gali būti pakeičiamas PA5 Sukurti šabloninį projektą panaudojimo atveju, jei projekto komandai projekto atributų informacija nėra reikalinga ir šabloninio projekto charakteristikos atitinka nustatymus reikalavimus projektui;
2. PA8 Nustatyti rizikos identifikacines reikšmes atributams - nustatyti reikšmė, kada atributas laikomas rizikingu. Be pačios reikšmės, naudotojas turi pasirinkti operatorių (didesnė už, mažesnė už, lygi, nelygi, intervale nuo). Šį veiksma gali atlikti projekto komanda, nes kiekviename projekte ji gali nustatyti skirtingas tos pačios rizikos reikšmes;
3. PA9 Atnaujinti atributų reikšmes - atlikus konfigūravimo dalį, naudotojas turi atlikti kitą veiksma – įkelti duomenis. Duomenys įkeliami periodiškai. Dažniausiai kas du mėnesius, nes sprintas trunka dvi savaites, tačiau komanda gali nuspręsti kitaip ir duomenis sekti dažniau. Atlikti tai rečiau nerekomenduojama *Agile* projektuose, nes dėl to gali dingti projekto statuso identifikacija;
4. PA10 Identifikuoti rizikos objektus – sukurtame projekte identifikuojami rizikingi artefaktai, kurių atributų reikšmės atitinka rizikingo objekto apibrėžimą;
5. PA11 Išanalizuoti rizikos objektą - Identifikavus riziką, reikia ją aprašyti, nustatyti tikimybę, statusą, poveikį, prioritetą, identifikuoti atsakingą asmenį parengti tolimesnį veiksma planą. Taip pat, išanalizuota rizika turi būti priskiriama rizikų kategorijai pagal projekto rizikų klasifikatorių. Jei rizika yra neidentifikuota ir neaprašyta pradinėje projekto parengimo stadijoje, ji turi būti išnagrinėjama ir priskiriama rizikų kategorijai;
6. PA12 Atnaujinti rizikos objekto atributų reikšmes - metodika apima ir trečiąją rizikų valdymo etapą – rizikos sekimą ir kontrolę. Išanalizuoti rizikingi objektai turi būti periodiškai patikrinami, atnaujinamos rizikos objekto atributų reikšmės, sekama istorija. Kol rizika nėra eliminuota, ji turi būti kontroliuojama komandos.



7 pav. Projekto dalyvio rizikų valdymo modulio panaudojimo atvejai

2.2. Rizikų valdymo metodikos procesas

Rizikų valdymo procesas yra sudaromas iš dviejų etapų: rizikų valdymo modulio paruošiamieji darbai bei rizikos valdymas modulyje.

Paruošiamieji darbai yra atliekami dalykinės srities eksperto (angl. *subject matter expert* – *SME*). Modulyje turi būti aprašomi pagrindiniai *Scrum* projektuose egzistuojantys artefaktai (aprašyti skyrelyje 2.2.2) ir atributai (aprašyti skyrelyje 2.2.3) bei šiems artefaktams apibrėžtos pagrindinės rizikų kategorijos (aprašytos skyrelyje 2.2.4), kurių metu konkrečios kategorijos rizikų atsiradimo tikimybė yra didžiausia – tai reikalinga efektyviam rizikų identifikavimo procesui. Toliau kiekvienai rizikos kategorijai yra nustatomos rizikos (aprašytos skyrelyje 2.2.5).

Rizikos valdymo procesas yra atliekamas *Scrum* projekto komandos. Nėra išskiriama konkreti *Scrum* komandos rolė, kuri turėtų dirbti su šiuo moduliu – taip skatinamas projekto būklės skaidrumas tarp komandos narių bei kiekvieno nario įsitraukimas.

Projekto dalyvis atlieka visus pagrindinius rizikų valdymo veiksmus:

1. rizikos identifikavimą;
2. analizę – analizės metu identifikuojami pagrindiniai rizikos atributai bei išsiaiškinama, ar nustatyta rizika jau yra įtraukta į žinomų rizikų sąrašą klasifikatoriuje. Jei tai nebuvo padaryta, naujai rizikai yra priskiriama kategorija ar sukuriama nauja ir rizika yra įtraukiama į klasifikatorių;
3. tolimesnę rizikos kontrolę – rizika yra susiejama su konkrečiais aktualiais projekto artefaktais efektyviam sekimui. Susiejant riziką su artefaktu galima iš anksto numatyti, kada rizika bus aktuali bei be didelių laiko sąnaudų efektyviai ją kontroliuoti. Rizikai eliminavus reikia paruošti rizikos panaikinimo pašalinimo ataskaitą istorijos sekimui.

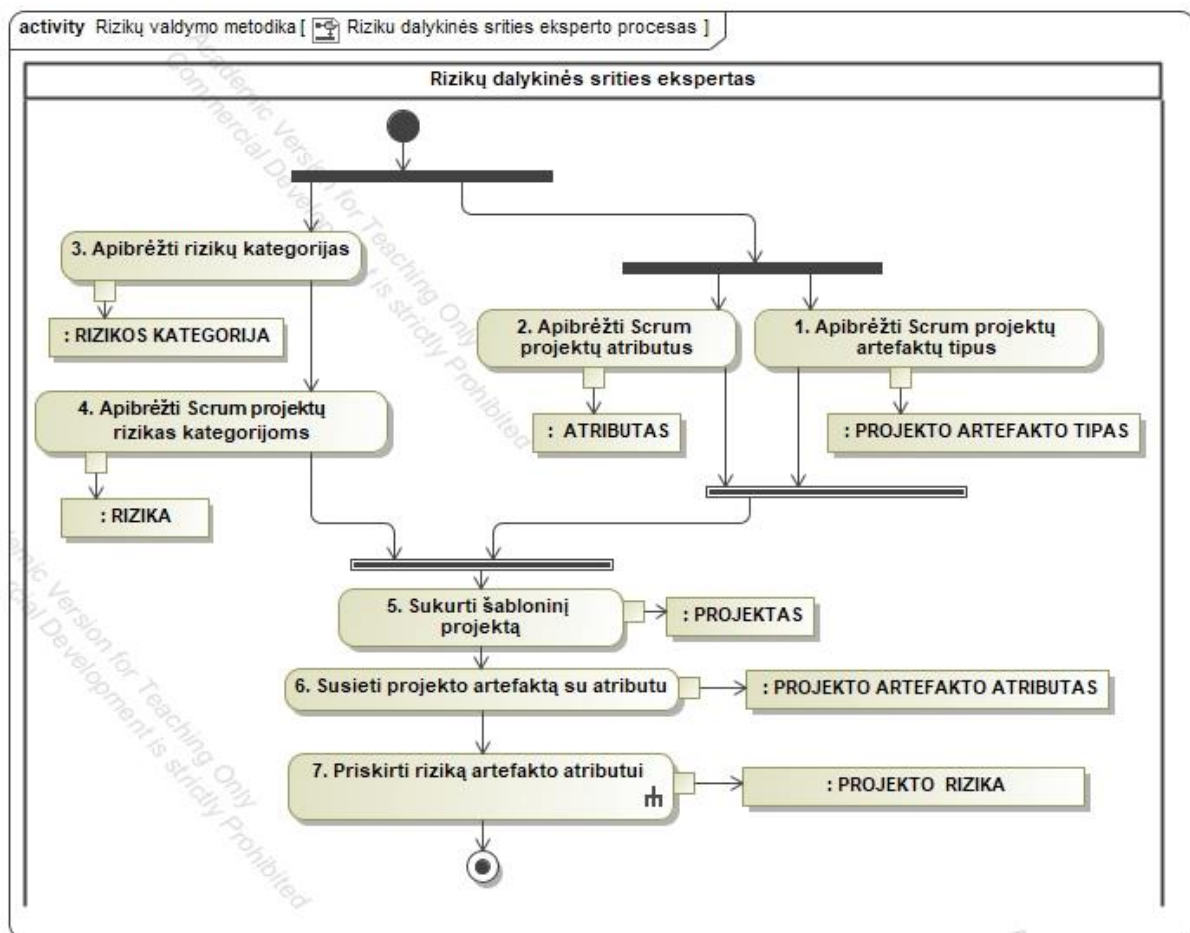
2.2.1. Dalykinės srities eksperto procesas

Šiame etape sukuriamas šabloninis rizikų klasifikatorius pagal projekto artefaktus, kuris vėliau gali būti koreguojamas ir pritaikomas pagal komandos poreikius konkrečiam projektui.

Rizikų dalykinės srities ekspertas sukuria *Scrum* projekto šabloną, kuris vėliau gali būti pritaikomas konkrečiam projektui arba modifikuojamas pagal projekto specifiką – šis procesas atliekamas iš tokių žingsnių (8 pav.):

1. Žingsnis „1. Apibrėžti *Scrum* artefaktų tipus“ – nurodomi pagrindiniai *Scrum* projektų valdymo metodikoje egzistuojantys artefaktai, standartinis artefaktų sąrašas pateikiamas 2.2.2 skyrelyje;
2. Žingsnis „2. Apibrėžti *Scrum* projektų atributus“ – nurodomi pagrindiniai *Scrum* projektų valdymo metodikoje egzistuojantys artefaktams priskiriami atributai, standartinis atributų sąrašas pateikiamas 2.2.3. skyrelyje;
3. Žingsnis „3. Apibrėžti rizikų kategorijas“ - nurodomos pagrindinės *Scrum* projektų valdymo metodikoje egzistuojančios rizikų kategorijos, standartinis kategorijų sąrašas pateikiamas 2.2.4. skyrelyje;

4. Žingsnis „4. Apibrėžti rizikas kategorijoms“ – kiekvienai kategorijai nurodomos jai priklausančios rizikos – sukuriamas rizikų klasifikatorius;
5. Žingsnis „5. Sukurti šabloninį projektą“ – suformavus artefaktų ir atributų sąrašus projekto kūrimo žingsniu yra paruošiama aplinka šių komponentų susiejimui;
6. Žingsnis „6. Susieti projekto artefaktą su atributu“ – šabloniniame projekte yra parenkami artefaktai iš artefaktų sąrašo (sąrašas pateikiamas skyrelyje 2.2.2.) Kiekvienam iš artefaktų priskiriami atributai, kurių duomenys bus sekami šiame projekte. Visi parinkti artefaktai turi turėti bent vieną priskirtą atributą, nes kitu atveju nebus galimybės sekti šio artefakto būsenos;
7. Žingsnis „7. Priskirti riziką artefakto atributui“ – pasirinkus projekto artefaktus bei jiems priklausančius atributus, kiekvienam iš atributų galima priskirti riziką iš ketvirtajame žingsnyje suformuoto sąrašo bei nurodyti tokią atributo reikšmę, kada atributas yra laikomas rizikingu – atitinka rizikos sąlygą.



8 pav. Rizikų dalykinės srities eksperto procesas

2.2.2. „1. Apibrėžti Scrum projektų artefaktų tipus“ žingsnis

Rizikų dalykinės srities ekspertas suformuoja Scrum projektų artefaktų sąrašą (4 lentelė.). Artefaktai yra Scrum projekto sudedamosios dalys pagal kuriuos projektas yra suskaidomas į etapus ir iš kurių yra kuriamas projekto valdymo procesas. Projekto valdymo metu reikia identifikuoti, kada artefakto būseną yra nukrypusi nuo normos ir kada šis artefaktas gali būti

laikomas rizikingu. Žinant, kada artefaktas pasiekia nestabilią būseną, galima identifikuoti rizikas bei jas kontroliuoti artefakto ribose. Išplėstas rekomenduojamų artefaktų sąrašas pateikiamas prieduose – Priedas 1.

4 lentelė. Scrum projekto artefaktų pavyzdžiai

Nr.	Artefaktas	Artefakto aprašymas
AR1.	Produkto atliekamų darbų sąrašas (angl. <i>product backlog</i>)	Naujai sudaromas, tvarkingas sąrašas to, ko reikia produktui patobulinti.
AR5.	Naudotojo pasakojimo įgyvendinimas (angl. <i>user story implementation</i>)	Procesas, kurį sprinto metu atlieka Scrum projekto komanda, kad naudotojo pasakojimas atitiktų atlikto darbo apibrėžimą
AR8.	Funkcionalumo modulio įgyvendinimas (angl. <i>epic implementation</i>)	Procesas, kurį sprinto ar kelių sprintų metu atlieka Scrum projekto komanda, kad visi funkcionalumo modulyje esantys naudotojo pasakojimai atitiktų atlikto darbo apibrėžimą

2.2.3. „2. Apibrėžti Scrum projektų atributus“ žingsnis

Šiame žingsnyje aprašomi pagrindiniai Scrum projektų artefaktų atributai. Atributai identifikuoja, kokią informaciją apie konkretų artefaktą reikia sekti ir saugoti. Artefakto būklė yra sekama pagal tai, kokia yra jam priskirtų atributų reikšmė ir ar ši reikšmė yra nukrypusi nuo normos – artefaktas tampa rizikingu. Kadangi tas pats atributas gali būti naudojamas keliems artefaktams, jie identifikuojami atskirai ir vėliau susiejami su artefaktais. Pavyzdys galėtų būti atributas „Pakartotinai atidarytų užduočių skaičius“, kuris gali būti susietas ir su „Funkcionalumo modulio įgyvendinimas“ artefaktu, ir su „Sprinto atliekamų darbų sąrašas“ artefaktu, kad būtų galima nustatyti pagrindinę reikalavimų rinkimo problemą, atsakomybių paskirstymą tarp skirtingo rango žmonių, bendravimo su klientu trūkumą.

Atributo tipas apibrėžia, kaip atributas gauna savo vertę, ir jį galima pasirinkti iš šių variantų:

1. Enumeratorius – atributas gauna vieną iš anksto nustatytų reikšmių. Dažniausiai naudojamas atributams, apibrėžiantiems pasirengimo, EHR, projekto sveikatos rodiklių būsenas;
2. Skaičius – atributas gauna skaičiaus vertę, kurią naudotojas įveda rankiniu būdu;
3. Apskaičiuota reikšmė – atributas gauna reikšmę, kuri importuojama iš išorinių šaltinių (pvz., JIRA) naudojant eksportuotas „Excel“ skaičiuokles arba apskaičiuojama naudojant kitų atributų reikšmes. Pvz. išeksportuojami visi projekto atliekamų darbų sąrašo elementai, apskaičiuojamas šių elementų kiekis ir taip gaunamas atributas „Užduočių kiekis atliekamų darbų sąraše“.

Toliau pateikiami keli atributų, kurių reikšmės gali būti sekamos siekiant įvertinti artefakto statusą pavyzdžiai. Pilnas rekomendacinis atributų sąrašas pateikiamas prieduose – Priedas 2.

Išraiška naudojama tik apskaičiuotiems atributams apibrėžtiems formulėmis, pagal kurias bus nustatoma atributo vertė. Išraiškai apibrėžti naudojamos standartinės „Excel“ [30] formulės, kurios būtų pajėgios apdoroti iš projektų valdymo sistemos atsisiųstą duomenų failą. Šių formulių naudojimo privalumas yra toks, kad jos yra plačiai pritaikomos, pažįstamos ir lengvai

suprantamos projekto komandos, taip pat yra suteikiama galimybė sukurti papildomus atributus su jau žinoma sintakse. Pavyzdžiui, toliau 5 lentelėje pateikiamas artefakto „Sprinto atliekamų darbų sąrašas“ rekomenduojamų atributų sąrašas. Atributas „A6 Pajėgumo standartinis nuokrypis“ apskaičiuojamas pagal atributo „A5 Pajėgumas“ reikšmes taikant standartinio nuokrypio formulę ir pasirenkant trijų reikšmių intervalą, o tai reiškia, kad nuokrypis apskaičiuojamas pagal tris paskutines greičio reikšmes. Atributas „A7 Paruoštų užduočių skaičius“, kuris nurodo užduočių, atitinkančių parengtos užduoties apibrėžtį, skaičių, apskaičiuojamas iš susieto įkeliamų duomenų failo, skaičiuojant duomenų eilutes, kurių sąlyga tenkinama, kai stulpelio „Label“ reikšmė yra „Patvirtinta“, nurodanti, kad užduotis yra peržiūrėta ir patvirtinta projekto komandos ir kliento. Vienas atributas gali būti susietas su keliomis rizikomis. Tuo pačiu metu rizika gali būti susijusi su keliais požymiais.

Kuriant šabloninį projektą atributo išraiška nurodo, kaip atributas bus apskaičiuotas iš įkeliamų duomenų failo arba iš kitų atributų reikšmių, tačiau atributams, kurie išskaičiuojami iš failo reikšmių nenurodomos konkrečios stulpelio reikšmės – tiek įkeliamų duomenų failas, tiek failo stulpeliai gali skirtis kiekvieno projekto lygmenyje ir failas bei failo stulpeliai priskiriami atributui turi būti apibrėžiami projekto konfigūravimo metu („1. Apibrėžti projekto specifiką“ žingsnis ir „5. Priskirti stulpelių reikšmes apskaičiuojamiems atributams“ žingsnis). Šiame žingsnyje išraiškos formulėse yra paliekama „stulpelis“ ir įgauna tikslią reikšmę formulėje žingsnyje „5. Priskirti stulpelių reikšmes apskaičiuojamiems atributams“ žingsnis, kai atributams yra priskiriama tikslus šabloninio failo stulpelis/stulpeliai.

5 lentelė. Scrum projektų atributų pavyzdžiai

Nr.	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas (Atributas“ klasė – „Tipas“)	Išraiška
A2	Klaidų skaičius	Apskaičiuota reikšmė	COUNTIF('file!'“stulpelis“, "Bug")
A3	Laikas, skiriamas klaidų taisymui	Skaičius	-
A4	Standartinio nuokrypio apimtis	Skaičius	-
A5	Pajėgumas (angl. <i>velocity</i>)	Apskaičiuota reikšmė	SUM('file!'“stulpelis“)
A6	Pajėgumo standartinis nuokrypis	Apskaičiuota reikšmė	ST.DEV(A5,3)

2.2.4. „3. Apibrėžti rizikų kategorijas“ žingsnis

Šiame proceso žingsnyje yra sukuriamas rizikų klasifikatorius – sukuriamos rizikų grupės, kurios yra susiejamos su projekto artefaktais. Rizikos kategorijų ir artefaktų kūrimo procesas gali būti vykdoma lygiagrečiai, kadangi tiek rizikų klasifikatorius, tiek artefaktų tipų aibės yra nepriklausomi sąrašai, kurie yra susiejami tik konkretaus projekto ribose. Projekto komanda atrenka, kurios kategorijos ir kurie artefaktai yra aktualūs pagal projekto charakteristikas.

6 lentelėje pateikiami keli rizikų kategorijų Scrum projektuose pavyzdžiai. Pilnas rekomendacinis rizikų kategorijų sąrašas Scrum projektuose, kuris vėliau gali būti papildomas

ar iš kurio galima atrikti poaibį kategorijų, kuriomis bus manipuluojama yra prieduose – Priedas 4.

6 lentelė. *Scrum* projektų rizikų kategorijų pavyzdžiai

Nr.	Rizikos kategorija	Aprašymas
K1.	Reikalavimų tenkinimo rizikos	Rizikos, susijusios su suinteresuotų asmenų keliamų reikalavimų tenkinimu kuriamam ar tobulinamam produktui
K2.	Komandos narių kompetencijų rizikos	Rizikos, susijusios su projekto komandos turimomis kompetencijomis, resursų paskirstymas ne pagal žinomą sritį, gabumus, kompetencijų nuvertinimu ar pervertinimu
K3.	Bendradarbiavimo rizikos	Rizikos, susijusios su projekto komandos tarpusavio bendradarbiavimu bei pasikeitimų eskalavimu visiems suinteresuotiems komandos nariams – skaidri komunikacija tarp produkto vadovų, programuotojų, testuotojų.

2.2.5. „4. Priskirti *Scrum* projektų rizikas rizikų kategorijai“ žingsnis

Šiame etape rizikų rinkinys, būdingas pagal *Scrum* metodiką valdomiems projektams. Kiekviena rizika turi būti priskiriama rizikos kategorijai, kadangi rizikos kategorija, o ne konkreti rizika yra susiejama su projekto artefaktu – projekto artefaktui būdingos visos tam tikrai kategorijai priskirtos rizikos. Toliau pateikiami keli rizikų, kurios bus sekamos projekte, pavyzdžiai (7 lentelė). Pilnas rekomendacinis rizikų sąrašas pateikiamas prieduose – Priedas 5.

7 lentelė. *Scrum* projektų rizikų pavyzdžiai

Id.	Rizika	Rizikos kategorija
R1	Neapibrėžti reikalavimai	K1 Reikalavimų tenkinimo rizikos
R5	Neatsižvelgiama į komandos narių turimas žinias - neteisingas pareigų paskirstymas	K2 Komandos narių kompetencijų rizikos
R8	Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius	K3 Bendradarbiavimo rizikos
R10	Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus	K4 Apmokymo rizikos
R16	Testuotojų darbo atsilikimas nuo programuotojų sprinto metu	K9 Kokybės užtikrinimo rizikos
R17	Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams	K9 Kokybės užtikrinimo rizikos
R18	Vienetų testavimai - neatidus programuotojų testavimas	K9 Kokybės užtikrinimo rizikos

2.2.6. „5. Sukurti šabloninį projektą“ žingsnis

Sukūrus artefaktų ir atributų sąrašą reikia sukurti aplinką, kurioje šie komponentai yra susiejami. Tam naudojamas šabloninis projektas – tai projektas, kuris nėra sukurtas konkrečiam domeniui vystyti, šiame projekte yra sukuriami ryšiai tarp artefaktų ir rizikų kategorijų, kurių rekomenduojama laikytis valdant rizikų procesą. Sukūrus šabloninį projektą

yra suformuojamas 3.4 skyrelyje aprašytos klasės „Projektas“ egzempliorius (8 lentelė). Šabloninio projekto individas neturi tokių atributų, kaip užsakovas, kaina ar terminas, kadangi tai nėra realaus gyvenimo objektas, o dirbtinai sukuriama aplinka su rekomendacinėmis gairėmis, kaip reikėtų sieti skirtingų klasių individus.

8 lentelė. Šabloninio *Scrum* projekto charakteristikos

Nr.	Klasės atributas	Reikšmė
1.	id	001
2.	Pavadinimas	<i>Scrum</i> šablonas
3.	Projekto pradžia	Sistemos data
4.	Aprašymas	Projektas, skirtas apibrėžti numatytąsias priklausomybės tarp sistemos komponentų: artefaktų ir atributų bei rizikos sąsajos su projekto atributais
5.	Tikslas	Sukurti rizikų valdymo modulio šablona

Šabloniniam projektui aprašomos pagrindinės projekto charakteristikos bei iš 3.2.2. skyrelyje pateiktų artefaktų atrenkami tie, kurie bus aktualūs konkrečiau projekto rizikoms sekti – sukuriama esybė „Projekto artefakto tipas“ (9 lentelė).

9 lentelė. "Projekto artefakto tipas" lentelės pavyzdys

Id.	Projektas_id (esybės „Projektas“ reikšmė „Id“)	Artefakto_tipas_id (esybės „Artefaktas“ reikšmė „Id“)
1	001	AR1
2	001	AR2
3	001	AR3
4	001	AR4

8 lentelėje pateikiamas esybės „Projekto artefakto tipas“ pavyzdys, kuriame nurodoma, jog šabloniniame projekte bus nagrinėjami ir sekami šie artefaktai: AR1 – Produkto atliekamų darbų sąrašas, AR2 – Produkto tikslas, AR3 – Naudotojo pasakojimo aukšto lygio poreikių surinkimas ir AR4 – Naudotojo pasakojimo reikalavimų surinkimas. Jeigu artefaktų aibė projektui yra nepilna, reikia grįžti į artefaktų kūrimo žingsnį (2.2.2 skyrelis) ir papildyti sąrašą trūkstamais artefaktais, kurie paskui yra priskiriami projektui.

2.2.7. „6. Susieti projekto artefaktą su atributu“ žingsnis

Sukūrus šabloninį projektą šiame projekte reikia susieti artefaktus su jiems priklausančiais atributais. Tai atliekama inicijavus atributo pridėjimą prie projekto artefakto, kuris buvo priskirtas projektui žingsnyje (2.2.6 skyrelis). Atributai nėra pridėjami visai artefaktų aibei, o tik projektui atrinktiems artefaktams, šiuo atveju šabloniniam projektui atrinktiems artefaktams, kadangi tik projekto artefaktų atributų informaciją yra prasminga saugoti ir sekti.

Tas pats atributas gali būti priskiriamas keliems artefaktams, tačiau atributas negali būti pridodamas tam pačiam artefaktui kelis kartus – kiekvienas atributas prie artefakto yra pridodamas po vieną kartą.

Projekto artefaktą susiejus su atributu yra sukuriama esybė „Projekto artefakto atributas“. Šioje lentelėje yra sukuriama sąsaja tarp projektui priskirtų artefaktų bei jiems aktualių atributų. Taip pat lentelėje „Projekto artefakto atributas“ yra saugoma artefakto atributo reikšmė – „Atributo reikšmė“, kurią įgyja konkretus atributas duomenų įkeliamų duomenų ar rankinio įvedimo į sistemą metu. Pagal šią reikšmę yra nustatoma, ar atributas gali būti laikomas rizikingu. Šabloniniam projektui ši reikšmė nėra pildoma, kadangi šabloninis projektas nėra realaus gyvenimo projektas ir tokio tipo duomenų neturi, tačiau lentelės struktūra leidžia sekti duomenis konkrečiam projektui. Taip pat lentelėje nurodoma atributo reikšmės fiksavimo data istorijai sekti ir atributo pokyčiui matuoti.

10 lentelė. "Projekto artefakto atributas" lentelės pavyzdys

Id.	Projekto_artefakto_id (esybės „Projekto artefakto tipas“ reikšmė „Id“)	Atributas_id (esybės „Atributas“ reikšmė „Id“)
AA1	AR1	A1
AA2	AR1	A7
AA3	AR1	A8
AA4	AR2	A1

10 lentelė. pateikiamas esybės „Projekto artefakto atributas“ pavyzdys, kuriame nurodoma, jog šabloniniame projekte artefaktui 1 – „001 – Produkto atliekamų darbų sąrašas“ bus nagrinėjami ir sekami šie atributai: A1 – „Nuoroda – sąsaja su artefaktu projektų valdymo platformoje“, A7 – Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius ir A8 – Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius, o projekto artefaktui 1 – „001 – Produkto tikslas“ bus sekamas šis atributas A49 – Paruoštų sprintų skaičius. Jeigu atributų aibė projekto artefaktams yra nepilna, reikia grįžti į atributų kūrimo žingsnį (2.2.3 skyrelis) ir papildyti sąrašą trūkstamais atributais, kurie paskui yra priskiriami projekto artefaktams.

Pagal šios lentelės sąryšius vėliau yra pildoma „Atributo reikšmė“ lentelė su realiai projekto atributų duomenimis. Šis procesas yra atliekamas projekto komandos rankiniu būdu arba importuojant duomenis iš projektų valdymo sistemos ir yra aprašomas skyreliuose 2.2.15, 2.2.16.

2.2.8. „7. Priskirti rizikas artefakto atributui“ žingsnis

Šabloniniame projekte sukūrus artefaktų ir jiems priklausančių atributų sąrašą (atlikus žingsnį, aprašytą skyrelyje 2.2.6), kiekvienam iš jų galima priskirti riziką.

Rizikos priskyrimo atributui procesas (9 pav.) atliekamas tokia eiga:

1. Pasirenkamas projekto artefaktas (reikšmę iš sąrašo, aprašyto skyrelyje, pavyzdžiui „Sprinto atliekamų darbų sąrašas“);

2. Pasirinktam projekto artefakto atributams (susietoms reikšmėms iš 3.2.7. skyrelio, pavyzdžiui „Sprinto atliekamų darbų sąrašas“ artefaktui priskirtam atributui „Pajėgumas“) yra priskiriama rizika inicijavus rizikos pridėjimą mygtuko paspaudimu. Rizika pasirenkama iš 3.2.5. skyrelyje aprašyto klasifikatoriaus (atributui „Pajėgumo standartinis nuokrypis“ gali būti priskiriama rizika „Sprinto atliekamų darbų skaičiaus prognozės neatitikimas“).



9 pav. „7. Priskirti riziką artefakto atributui“ žingsnis

Rekomendacinis atributų sąsajos su rizikomis šablonas pateikiamas prieduose (Priedas 6).

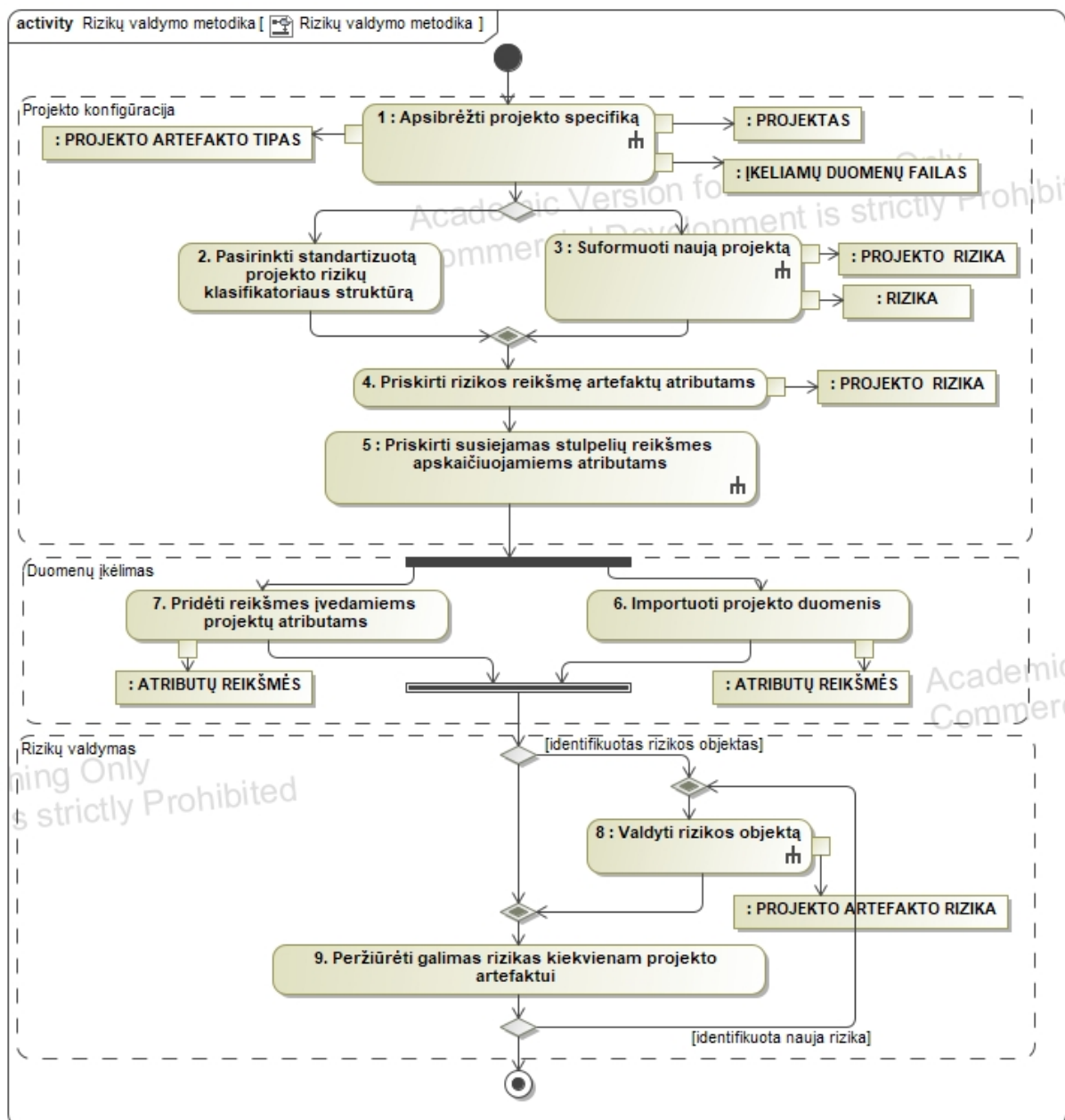
11 lentelė. Projekto rizikos lentelės pavyzdys

Id.	Projekto_artefakto_atributas (esybės „Projekto artefakto atributas“ reikšmė „Id“)	Rizika_id (esybės „Rizika“ reikšmė „Id“)
AAR1	AA1	R7
AAR2	AA1	R8
AAR3	AA1	R17

11 lentelė. Projekto rizikos lentelės pavyzdys pateikiamas esybės „Projekto rizika“ pavyzdys, kuriame nurodoma, jog šabloniniame projekto artefakto „AR1- Produkto atliekamų darbų sąrašas“ atributui bus nagrinėjamos ir sekamos šios rizikos: „R7 – Neatsižvelgiama į komandos narių turimas žinias – komandoje nėra tinkamų specialistų“, „R8 – Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius“, „R17 – Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams“. Jeigu rizikų aibė projekto artefaktams yra nepilna, reikia grįžti į rizikų kūrimo žingsnį ir papildyti sąrašą trūkstamomis rizikomis, kurios paskui yra priskiriamos projekto artefaktų atributams.

2.2.9. Projekto dalyvio procesas

Projekto dalyvio procesas pradamas nuo projekto konfigūracijos. Rizikų valdymo metodikos veiklos diagrama pateikta 10 pav.



10 pav. Rizikų valdymo modulio veiklos diagrama

2.2.10. „1. Apibrėžti projekto specifiką“ žingsnis

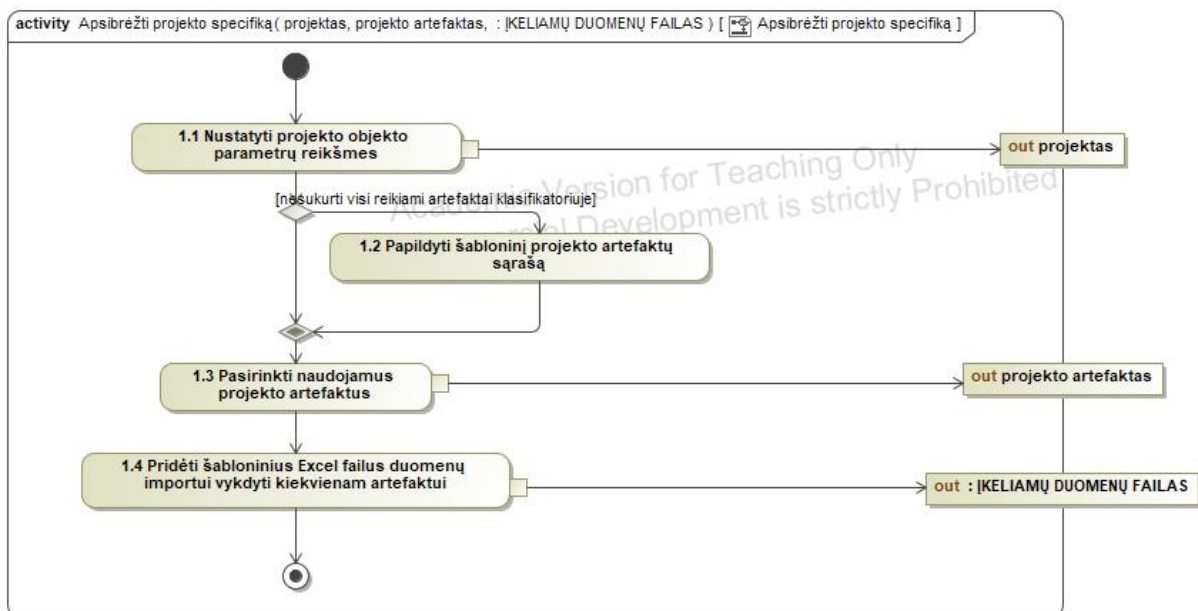
Scrum komanda, dirbdama konkrečiame projekte, pirmiausia turi apibrėžti projekto specifiką, kuri padėtų identifikuoti, kokios rizikos yra būdingos šiam projektui. Projekto specifikos apibrėžimas sudarytas iš šių pagrindinių žingsnių:

1. Inicijuoti projekto kūrimą – sukuriama aplinka rizikų procesui valdyti;
2. Nustatyti projekto parametrų reikšmes – apibrėžiama projekto esybė rizikų valdymo modelyje. Identifikuojami šie projekto atributai – tikslas, kaina, apimtis, trukmė. Šie atributai nurodo, kokias savybes ir kokius apribojimus iš užsakovo turi projektas;
3. Pasirinkti naudojamus projekto artefaktus – pasirenkami artefaktai iš rizikų dalykinės srities eksperto suformuotos artefaktų aibės („1. Apibrėžti Scrum projektų artefaktų tipus“ žingsnis), kurie yra aktualūs konkrečiam projektui;

4. Pridėti šabloninius Excel failus duomenų importui vykdyti – pridedami šabloniniai eksportuojamų duomenų failai iš projektų valdymo sistemos, kurie bus naudojami projektų artefaktų atributų reikšmėms pildyti.

Kiekvienam artefaktui yra įkeliamas įkeliamų duomenų failo šablonas – failas gali neturėti jokių duomenų, šis žingsnis yra skirtas tam, kad įkeliant realius failus su duomenimis sistema gebėtų atpažinti, iš kurių importuoto failo stulpelių turi būti apskaičiuojamos atributų reikšmės. Pvz., projekto artefaktui „AR16 – Produkto prieaugis“ komandai, naudojančiai projektų valdymo sistemą JIRA gali būti naudojamas „Sprint report“ eksportuojamų duomenų failas, kuriame yra išeksportuojamos visos atlikto sprinto užduotys bei jų metrikos.

Šiame žingsnyje išraiškos formulėse yra paliekama „stulpelis“ ir įgauna tikslią reikšmę formulėje žingsnyje „5. Priskirti stulpelių reikšmes apskaičiuojamiems atributams“ žingsnis, kai atributams yra priskiriama tikslus šabloninio failo stulpelis/stulpeliai.



11 pav. „1. Apsibrėžti projekto specifiką“ žingsnis

Lentelėje pavaizduojama, kaip keičiasi skyriuje 2.2.3 pavyzdyje pateiktų apskaičiuojamų atributų išraiškos po pirmojo projekto konfiguravimo žingsnio – įkeliamų duomenų failų pridėjimo.

12 lentelė. Atributų išraiškos priskyrus įkeliamų duomenų failo stulpelius

Nr.	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas (Atributas“ klasė – „Tipas“)	Išraiška
A2	Klaidų skaičius	Apskaičiuota reikšmė	COUNTIF('failas_artefaktui!'!„stulpelis“, "Bug")
A5	Pajėgumas (angl. <i>velocity</i>)	Apskaičiuota reikšmė	SUM('failas_artefaktui!'!„stulpelis“)
A6	Pajėgumo standartinis nuokrypis	Apskaičiuota reikšmė	ST.DEV(A5,3)

2.2.11. „2. Pasirinkti standartizuotą projektą“ žingsnis

Projekto komandai identifikavus projekto specifiką bei sekamus artefaktus reikia nustatyti, ar konkrečiam projektui gali būti panaudojamas rizikų dalykinės srities eksperto sudarytas šabloninis *Scrum* projektas rizikoms valdyti, t. y. projektas, kuriame yra nurodytos sąsajos tarp:

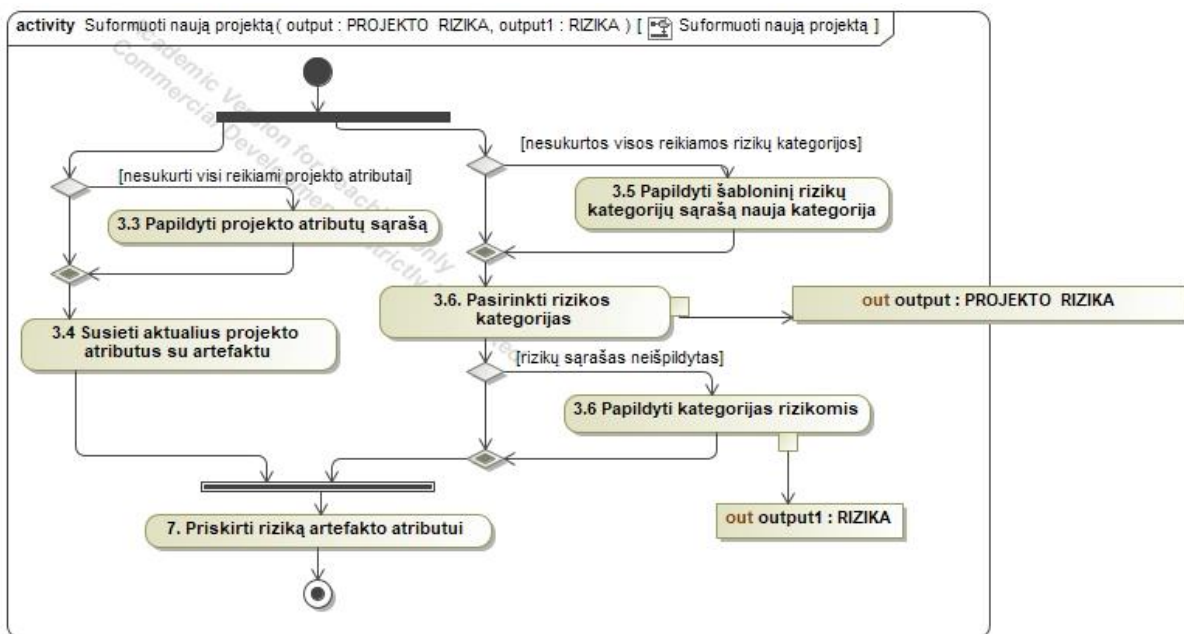
1. Projekto artefaktų bei atributų – ši sąsaja atsiranda projekto artefaktams priskyrus atributus (atliekant žingsnį aprašytą 3.2.7. skyrelyje);
2. Projekto artefaktų bei rizikų kategorijų – ši sąsaja atsiranda priskyrus riziką iš klasifikatoriaus artefakto atributui (atliekant žingsnį aprašytą 3.2.8. skyrelyje).

Projekto komandai pasirinkus standartizuoto projekto galimybę ir ankstesniame žingsnyje nustatčius kokie artefaktai bus naudojami šiame projekte, jų sąsaja su atributais ir atributų rizika yra pridama automatiškai pagal prieduose pateikiamą susiejimo žemėlapią.

2.2.12. „3. Suformuoti naują projektą“ žingsnis

Projekto komanda gali sudaryti rizikų klasifikatorių skirtą apibrėžti rizikas bei rizikų kategorijas konkrečiam projektui pagal turimą projekto specifiką.

Pirmiausia apibrėžiamos rizikos, būdingos konkrečiame projekte pagal turimus projekto atributus (12 pav.). Toliau šioms rizikoms yra apibrėžiamos naujos rizikos kategorijos arba rizikos yra priskiriamos rizikų dalykinės srities eksperto apibrėžtomis kategorijoms.



12 pav. „3. Suformuoti naują projektą“ žingsnis

Naujos rizikos bei rizikų kategorijos gali būti pridamos prie jau turimo rizikų dalykinės srities eksperto apibrėžto klasifikatoriaus.

Kurdamas naują projektą projekto dalyvis turi sukurti sąsajas tarp dalykinės srities eksperto aprašytų artefaktų bei atributų, taip pat, jei šie sąrašai yra neišbaigti, dalyvis gali juos papildyti.

Komandos dalyvio projekto konfigūravimo procesas nuo dalykinės srities eksperto skiriasi tuo, jog projekto dalyviui yra suformuoti šie sąrašai: *Scrum* projektų artefaktai (4 lentelė.), projektų artefaktų atributai (5 lentelė. *Scrum* projektų atributų), rizikų kategorijos (6 lentelė. *Scrum* projektų rizikų kategorijų) bei rizikos, suskirstytos pagal kategorijas (7 lentelė. *Scrum* projektų rizikų). Atsižvelgdamas į projekto specifiką projekto dalyvis papildo šiuos sąrašus trūkstamais atributais arba pasirenka reikšmes iš sistemos turimos aibės ir savo projekte atlieka žingsnius, kurie buvo atliekami šabloniniame projekte:

1. Atributų priskyrimas projekto atrinktiems artefaktams („6. Susieti projekto artefaktą su atributu“ žingsnis).
2. Rizikingos atributo reikšmės identifikavimas konkrečiam artefakto atributui – rizikos kategorijos ir projekto artefakto apjungimas („7. Priskirti rizikas artefakto atributui“ žingsnis).

2.2.13. „4. Priskirti rizikos reikšmę artefaktų atributams“ žingsnis

Sukurtame projekte su pasirinktais artefaktais, su jais susietais atributais bei atributų rizikomis, reikia rizikoms priskirti rizikos indikatorių, kada laikoma, jog atributo būseną yra rizikinga.

Pasirinkus riziką reikia nurodyti atributo reikšmę, kurią įgijęs atributas tampa rizikingu (pavyzdžiui atributui „Pajėgumo standartinis nuokrypis“ įgijus reikšmę 10 naudotojo taškų galima teigti, jog toks faktinis atliktų darbų nukrypimas nuo plano yra rizikingas, kadangi negalime aiškiai prognozuoti komandos pajėgumo bei numatyti darbų kiekio viename sprints). Reikšmei taip pat parenkamas operatorius: „daugiau nei“, „mažiau nei“, „lygu“, „nelygu“, „intervale“, „ne intervale“, kuriuo yra įvertinama, kokia atributo reikšmė yra laikoma rizikinga, kadangi vieniems atributams rizikinga reikšmė yra mažesnė nei nurodytas režis, o kitoms – didesnė. Pvz., atributui „Procentas apimties skiriamas techninių užduočių atlikimui“ rizikingas dydis gali būti 15 ir operatorius „mažiau nei“, reiškiantis, jog skiriant mažiau nei 15 procentų bendro laiko techninėms užduotims vykdyti artefaktas tampa rizikingu dėl nepakankamo dėmesio techninei projekto daliai. Tačiau atributui „Apimties skirtumas nuo pajėgumo“ rizikinga reikšmė yra didesnė nei nustatyta, nes esant dideliame skirtumui tarp numatytų darbų ir komandos pajėgumo kyla grėsmė dėl prasto projekto darbų planavimo.

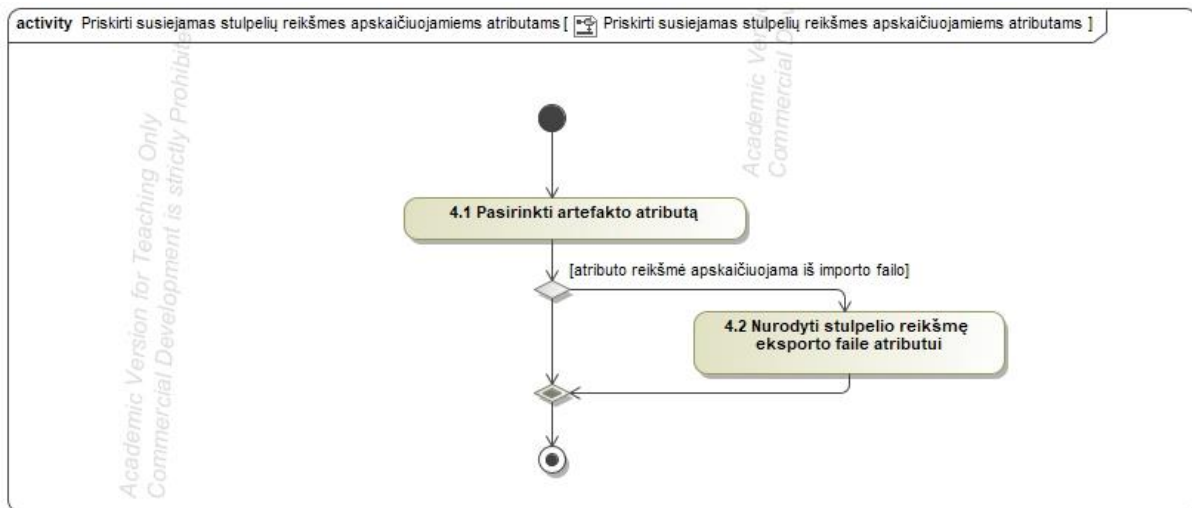
Priskyrus riziką atributui sukuriama sąsaja tarp projekto artefakto atributo (šiuo atveju „Pajėgumo standartinis nuokrypis“) ir rizikos (šiuo atveju „Srinto atliekamų darbų skaičiaus prognozės neatitikimas“) – sukuriama 3.1 skyriuje aprašomos esybės „Projekto rizika“ įrašas. Šiuo procesu projekto artefakto atributui sukuriama rizikinga reikšmė, pagal kurią galima identifikuoti artefakto būklę bei nurodoma, kokia rizika iškyla pasiekus šią būseną.

2.2.14. „5. Priskirti stulpelių reikšmes apskaičiuojamiems atributams“ žingsnis

Atlikus projekto konfigūraciją į rizikų valdymo sistemą turi būti sukonfigūruojami atributai, kurių „Atributo tipas“ klasės „Tipas“ reikšmė yra „Apskaičiuota reikšmė“ bei atributo išraiškoje yra įtraukiama įkeliamų duomenų failo aibė (nenaudojamos kitų atributų reikšmės) – tai reiškia, jog šios reikšmės yra importuojamos iš projekto valdymo sistemos eksportuojamų duomenų failo. Šiame žingsnyje kiekvienam apskaičiuojamam artefakto atributui reikia

nurodyti, kokie įkeliamų duomenų failo stulpeliai bus naudojami atributo rezultato skaičiavimui – išraiškos „stulpelis“ įgauna tikslią reikšmę.

Šiuo žingsniu (13 pav.) eksportuojamų duomenų failo stulpelio reikšmė yra susiejama su atributu – taip identifikuojama, kokią įkeliamų duomenų failo reikšmę priskirti konkrečiam atributui.



13 pav. „4. Priskirti stulpelių reikšmes importuojamiems atributams“ žingsnis

13 lentelėje pavaizduojama, kaip keičiasi 2.2.3 skyriuje pavyzdyje pateiktų apskaičiuojamų atributų išraiškos po paskutinio projekto konfigūravimo žingsnio – artefakto atributas turi tikslią failo bei stulpelio faile reikšmę, todėl formulė yra išbaigta ir galima vykdyti tolimesnį darbą su sistema – duomenų įkėlimą.

13 lentelė. Atributų išraiškos priskyrus įkeliamų duomenų failo stulpelius

Nr.	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas (Atributas“ klasė – „Tipas“)	Išraiška
A2	Klaidų skaičius	Apskaičiuota reikšmė	COUNTIF('failas_artefaktui'!“Issue Type“, "Bug")
A5	Pajėgumas (angl. <i>velocity</i>)	Apskaičiuota reikšmė	SUM('failas_artefaktui'!“Velocity“)
A6	Pajėgumo standartinis nuokrypis	Apskaičiuota reikšmė	ST.DEV(A5,3)

2.2.15. „5. Importuoti projekto duomenis“ žingsnis

Pradėjus dirbti su rizikų valdymo moduli, pvz. „JIRA“, reikia surinkti duomenis apie projektą ir importuoti juos į rizikų valdymo sistemą, kad būtų galima apskaičiuoti atributų, kurių reikšmės nėra tiesiogiai įvedamos į sistemą, rezultatus ir patikrinti jų būklę – ar atributas ir tuo pačiu, artefaktas, turintis šį atributą, gali būti laikomas rizikingu.

Eksportuojamų duomenų failui yra keliami šie reikalavimai:

1. Eilučių reikšmės atitinka konkrečius artefaktų objektus projektų valdymo sistemoje;

2. Stulpelių reikšmės identifikuoja konkretaus artefakto objekto atributų reikšmes, kurios yra susiejamos su rizikų valdymo sistemos atributais žingsnyje.

Tokį duomenų eksportuojamų duomenų šabloną šiuo metu teikia projektų valdymo sistema „Jira“.

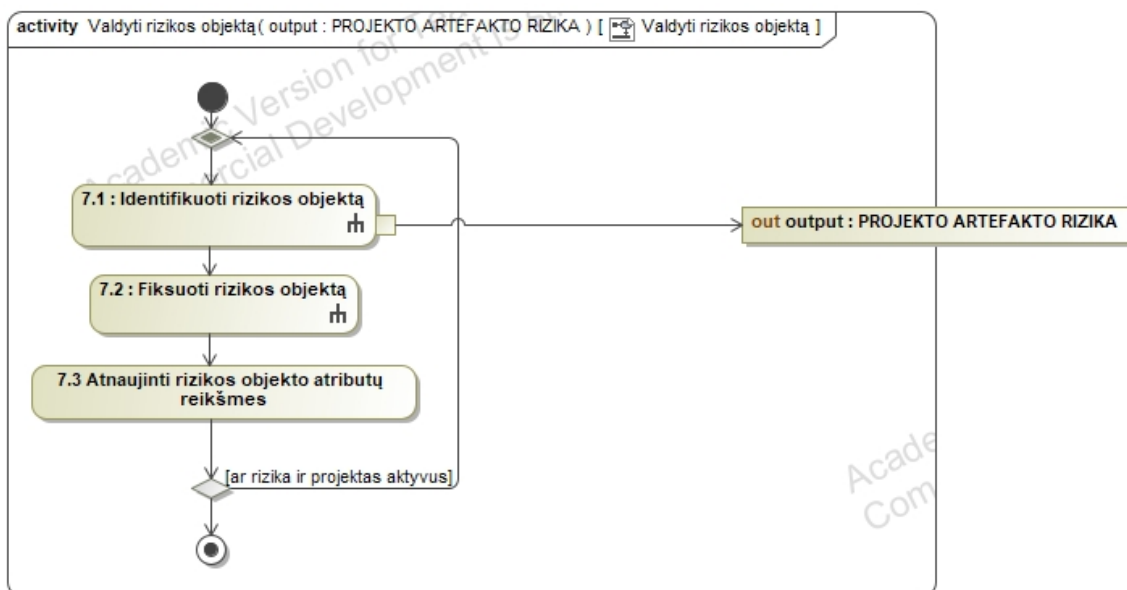
Pagal importuotus, įvedamus duomenis ir iš importuotų duomenų išskaičiuojamas reikšmės sistemos duomenų bazėje yra pildoma lentelė „Atributų reikšmės“. Ši lentelė pasipildo naujais įrašais, kai yra atliekamas veiksmas su atributais. Lentelėje užfiksuojamas konkretus projekto atributas, pavyzdžiui „Sprinto pajėgumas“ ir nurodoma jo reikšmė bei data, kada šio atributo duomenys buvo įkelti į sistemą. Turint informaciją apie projekto atributų reikšmes konkrečiu laiko momentu galima identifikuoti, kada ši reikšmė tampa rizikinga, t. y. atributo reikšmė lentelėje „Atributų reikšmės“ prilygsta reikšmei, nustatytai lentelėje „Projekto rizika“ ir taip atpažinti, kad atributas, o tuo pačiu ir artefaktas, kuriam priskirtas šis atributas yra laikomas rizikingu.

2.2.16. „6. Pridėti reikšmes artefaktų atributams“ žingsnis

Atlikus projekto konfigūraciją į rizikų valdymo sistemą turi būti įvedamos atributų reikšmės, kurių „Atributo tipas“ klasės „Taisyklė“ reikšmė yra „Skaičius“ arba „Enumeratorius“ – tai reiškia, jog šios reikšmės yra įvedamos rankiniu būdu. Įvedus duomenis užpildoma lentelė „Atributų reikšmės“ ir nurodoma data, kada užfiksuota konkreti reikšmė. Šias reikšmes rekomenduojama atnaujinti tuo pat metu, kada yra užkraunami duomenis iš projekto valdymo sistemos, kadangi kai kurie atributai yra išvestiniai ir yra apskaičiuojami iš kitų atributų reikšmių, todėl tiksliam duomenų įvertinimui reikia duomenų reikšmių gautų vienu metu.

2.2.17. „7. Valdyti rizikos objektą“ žingsnis

Rizikos objekto valdymas yra sudarytas iš trijų pagrindinių žingsnių (14 pav).



14 pav. „7. Valdyti rizikos objektą“ žingsnis

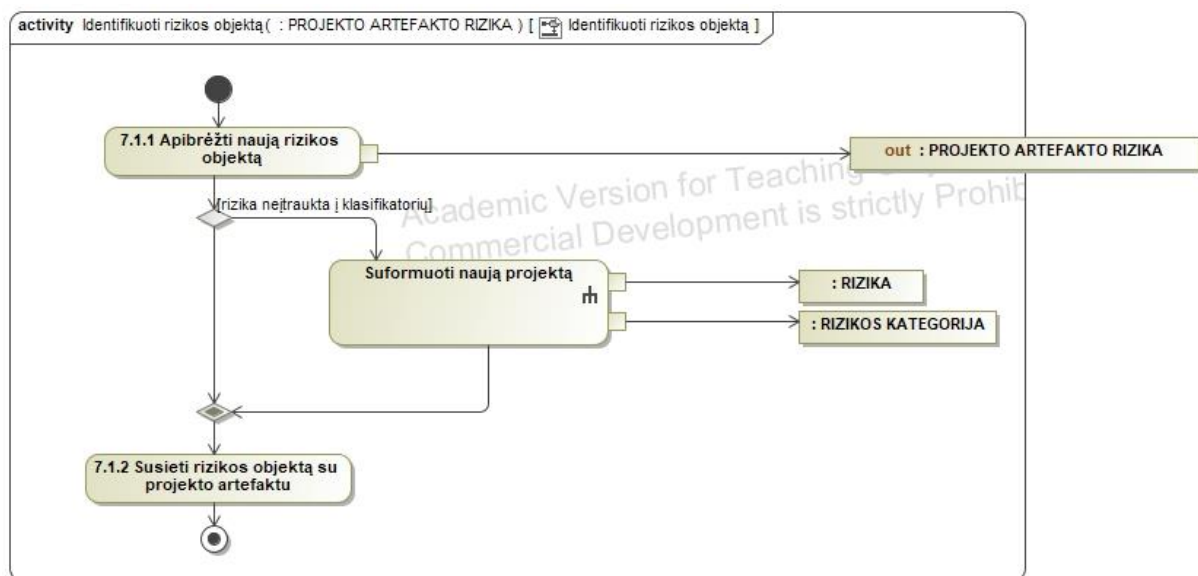
Šie žingsniai yra:

1. Rizikos objekto identifikavimas – šis žingsnis yra reikalingas tam, kad būtų galima ją kontroliuoti. Šio žingsnio metu yra užfiksuojama konkreti rizika bei su ja susietas projekto artefaktas;
2. Rizikos objekto analizė - atpažinus riziką reikalinga strategija, kaip šią riziką kontroliuoti bei valdyti. Rizikos analizės metu yra fiksuojami pagrindiniai rizikos atributai;
3. Rizikos objekto kontrolė – rizikos atributų reikšmių atnaujinimas bei tolimesnė stebėseną iki rizikos gyvavimo ciklo pabaigos – rizikos eliminavimo.

2.2.18. „7.1. Identifikuoti rizikos objektą“ žingsnis

Rizikos identifikavimo metu apibrėžiamas naujas rizikos objektas (15 pav.). Šiame žingsnyje nėra detalizuojami konkretūs objekto atributai, tačiau yra fiksuojama, ar atsiradusi rizika yra įtraukta į projekto komandos sudarytą rizikų klasifikatorių. Jei rizika nebuvo įtraukta į klasifikatorių, ji yra įtraukiama pagal naujos rizikos suformavimo procesą.

Toliau identifikuota rizika yra susiejama su konkrečiu projekto artefaktu, kuriam ji buvo užfiksuota.



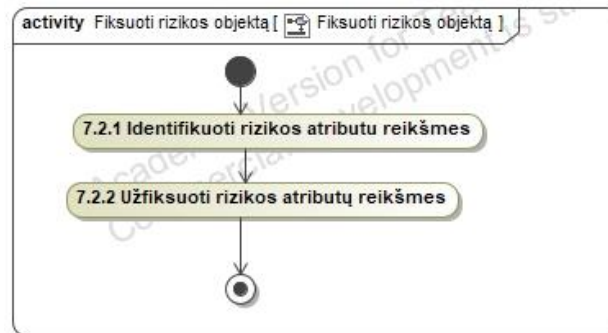
15 pav. „7.1 Identifikuoti rizikos objektą“ žingsnis

2.2.19. „7.2. Fiksuoti rizikos objektą“ žingsnis

Identifikuotas rizikos objektas toliau yra išanalizuojamas – nustatomos ir užfiksuojamos pagrindinės projekto artefakto rizikos atributų reikšmės (16 pav.), tokios kaip:

1. rizikos aprašymas;
2. nustatymo data;
3. tikimybė – įvertinama rizikos tikimybė dešimtaine išraiška;
4. poveikis – rizikos įvykio poveikis projekto artefaktui;
5. prioritetą – aukščiausio/vidutinio/žemo/trivialaus lygio prioritetą;

6. atsakingas asmuo – *Scrum* komandos narys, atsakingas už tolimesnę rizikos kontrolę ir eliminavimą;
7. statusas – nustatoma rizikos būseną: užfiksuota, išanalizuota, peržiūrėta, uždaryta;
8. paskutinės peržiūros data;
9. veiksmų planas – rizikos eliminavimo žingsniai;
10. pašalinimo data;
11. pašalinimo aprašymas – nurodoma, kaip rizika buvo pašalinta.



16 pav. „7.2 Fiksuoti rizikos objektą“ žingsnis

2.2.20. „7.3. Atnaujinti rizikos objekto atributų reikšmes“ žingsnis

Išanalizuota rizika yra peržiūrima atsakingo asmens ar bet kurio kito *Scrum* komandos nario. Patikrinamos rizikos objekto atributų reikšmės, įvertinama, ar informacija nėra pasenusi, ir jei taip, atributų reikšmės yra atnaujinamos. Kai rizika yra pašalinama, jos būseną pakeičiama į „Uždaryta“. Rizikos būsenos viso rizikos gyvavimo metu detalizuojamos 15 lentelė. Rizikos objekto gyvavimo ciklas.

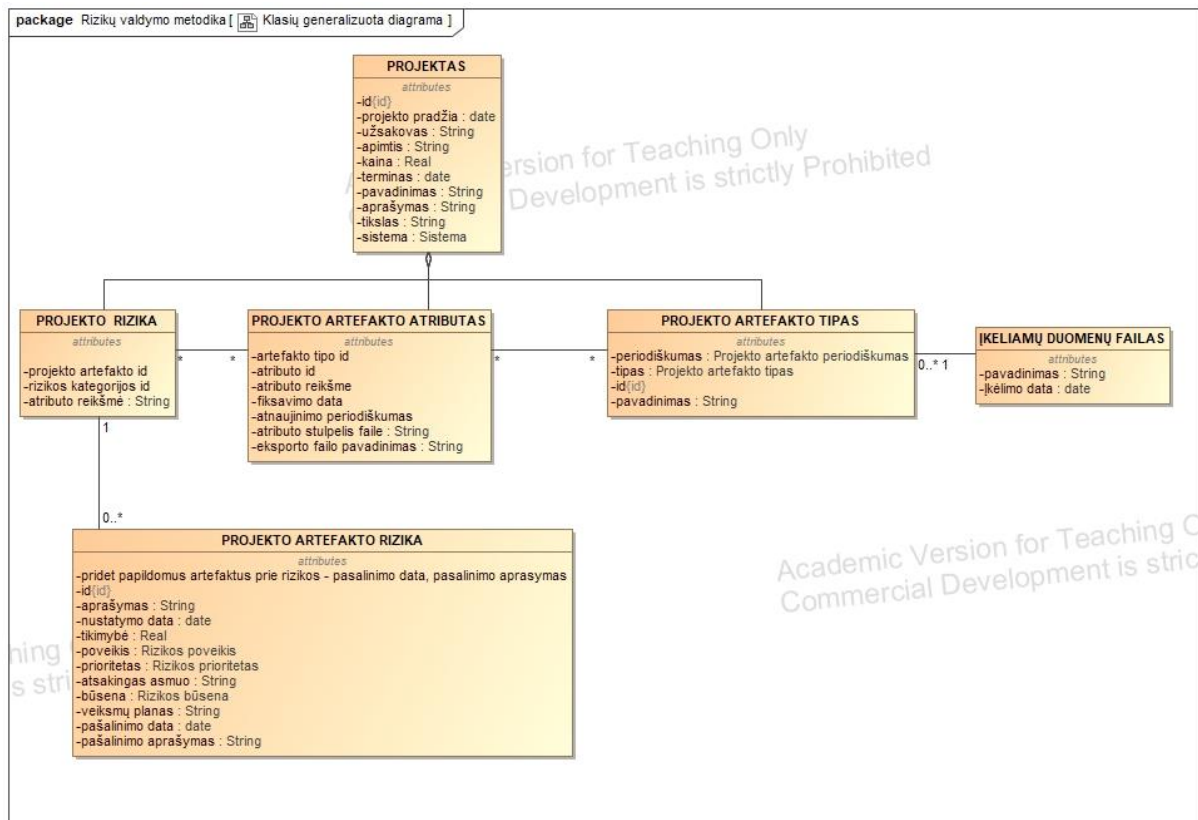
2.3. Reikalavimų apibendrinimas

Rizikų valdymo sistemai keliami pagrindiniai reikalavimai apimantys rizikų identifikavimą, analizę bei kontrolę visais projekto gyvavimo etapais. Projekto valdymo procesas yra suskaidytas į projekto artefaktus, kurie apima visą projekto gyvavimo ciklą ir sudaro projekto visumą. Kiekvienas projektui aktualus projekto artefaktas turi turėti priskirtus atributus, pagal kurių reikšmes identifikuojama, ar konkretus projekto artefaktas gali būti laikomas rizikingu. Atributas – tai artefakto duomuo. Rizikų valdymo metodikoje šie atributai yra susiejami su rizikomis, kurios yra suklasifikuotos į kategorijas. Su rizikos priskyrimu artefakto atributui identifikuojame, kokią reikšmę turi įgyti atributas, kad artefaktas galėtų būti laikomas rizikingu. Apibrėžus projekto sudedamąsias dalis bei atributams priskyrimus aktualias rizikas iš klasifikatoriaus toliau yra vykdomas duomenų importas iš projektų valdymo sistemos bei rankinis atributų reikšmių atnaujinimas įvedamiems bei enumeratoriaus atributams. Atributui pasiekus tam tikrą reikšmę, kuri sistemoje yra laikoma rizikinga, sistema informuoja naudotoją apie galimas rizikas. Tai užtikrina, jog neidentifikuotų ir vėlesnėse projekto stadijose atsiradusių rizikų neturėtų likti – rizika turėtų būti atpažinta vos atributui nukrypęs nuo normos.

3. Rizikų valdymo metodikos eksperimentinės realizacijos projektas

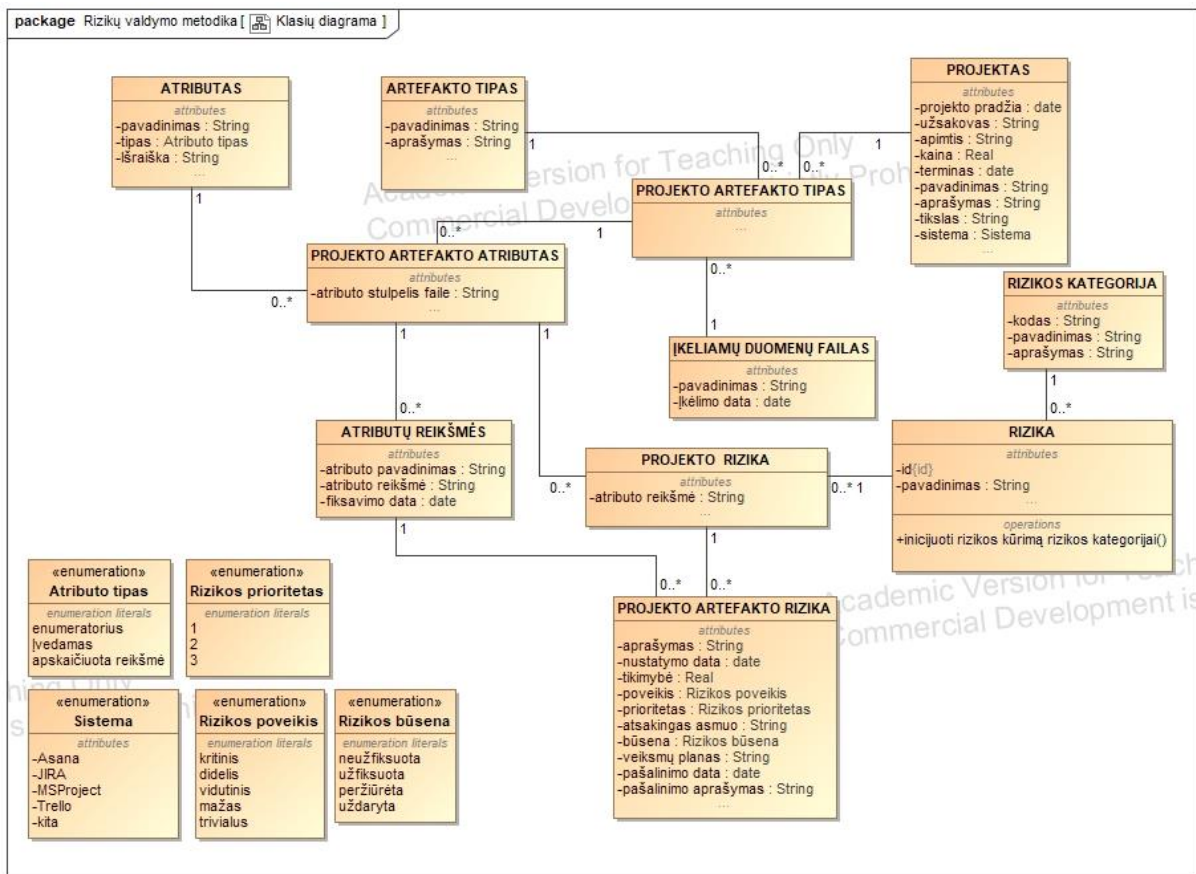
3.1. Esybių ryšių diagrama

Diagrama susideda iš klasių, kuri viena kitą papildo. Pagrindinė esybė „Projektas“ turi savo atributus bei agregavimo ryšius su kitomis klasėmis, kurios apibrėžia projekte naudojamus sistemos rizikų bei *Scrum* komponentų klasifikatorių objektus. Klasių diagrama vaizduoja sistemos sudedamąsias dalis su jų atributais (saugoma informacija), kuri nurodo, kokia informacija sistemoje yra saugoma. Projektuojamos rizikų valdymo sistemos esybių diagrama pavaizduota 17 pav.



17 pav. Esybių ryšių diagrama

Pavaizduota diagrama gali būti išskaidoma į išplėstinę klasių diagramą atvaizduojant tarpines lenteles, kuriose yra saugomi projekto objektų ryšiai bei jų sąsaja su klasifikatorių objektais. Pvz., „Projekto artefakto tipas“ lentelė atvaizduojama kaip tarpinė lentelė tarp „Projektas“ ir „Artefakto tipas“ klasių, o lentelės struktūra yra šių dviejų lentelių pirminių raktų sąjunga. Išplėsta esybių ryšių diagrama pavaizduota 18 pav. Diagramos elementai aprašyti 14 lentelėje.



18 pav. Rizikų valdymo modulio klasių diagrama

14 lentelė. Rizikų valdymo sistemos esybių lentelė

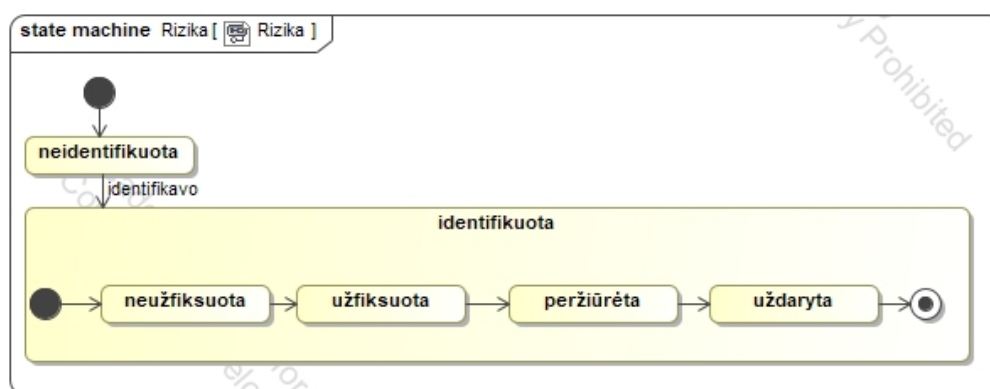
Esybė	Aprašymas	Atributai
Artefaktas	Esybę sudaro <i>Scrum</i> projekto šabloniniai artefaktų tipai, kurie vėliau yra siejami su konkrečiu projektu	Pavadinimas – artefakto pavadinimas, skyrelis 3.2.2. stulpelis „Pavadinimas“
Atributas	Esybę sudaro <i>Scrum</i> projekto atributai, šabloninis esybės objektų sąrašas pateikiamas skyrelyje 3.2.3	Pavadinimas – atributo pavadinimas, skyrelis 3.2.3. stulpelis „Pavadinimas“ Tipas – atributo duomenų tipas, skyrelis 3.2.3. stulpelis „Atributo tipas“. Atributas gali būti įvedamas, importuojamas arba išskaičiuojamas iš importuotų duomenų. Šio atributo reikšmė yra enumeratorius iš reikšmių: „Skaičius“ – įvedamas naudotojo tiesiogiai rizikų valdymo modulyje „Enumeratorius“ – pasirenkama reikšmė iš sąrašo „Išskaičiuojamas“ – apskaičiuojama reikšmė iš projekto valdymo modulio eksportuojamo ir į sistemą importuoto failo duomenų
Projektas	Esybę sudaro projekto atributai, apibrėžiantys projekto specifiką	Projekto pradžia – projekto pradžios data Užsakovas – klientas Apimtis – projekto apimtis Kaina – projekto kaina

Esybė	Aprašymas	Atributai
		<p>Terminas – projekto sutarties pabaigos data</p> <p>Pavadinimas – projekto pavadinimas</p> <p>Aprašymas – projekto aprašymas</p> <p>Tikslas – projekto siekiamo rezultato apibrėžimas, gali būti susiejama su artefaktu „Produkto tikslas“</p> <p>Sistema – kliento naudojama projekto valdymo sistema, pasirenkama iš enumeratoriaus reikšmių:</p> <p>JIRA Asana MS Project Kita</p>
Projekto artefakto tipas	Esybę sudaro projekto artefaktai, susiejami su konkrečiu projektu – pasirenkama, kokių artefaktų būseną bus sekama rizikų valdymo modulyje	Tarpinė lentelė artefaktams su projektu apjungti
Projekto artefakto atributas	Esybę sudaro projekto ir tam projektui priskirtų artefaktų identifikaciniai numeriai	Tarpinė lentelė artefaktams su atributais apjungti Atributo stulpelis faile – jeigu atributo taisyklė yra „Apskaičiuota reikšmė“, reikia nurodyti įkeliamų duomenų failo stulpelio reikšmę, kurios reikšmę fiksuoti kaip atributo reikšmę
Įkeliamų duomenų failas	Esybę sudaro projekto valdymo sistemos duomenų įkeliamų duomenų failų šablonai, kurie vėliau yra naudojami importuojamų atributų identifikacijai	<p>Pavadinimas – šabloninio failo pavadinimas</p> <p>Įkėlimo data – failo įkėlimo data</p>
Projekto artefakto rizika	Esybė skirta rizikos objektams, identifikuotiems projekte, užfiksuoti ir sekti. Esybę sudaro rizikos objektų informacija, reikalinga rizikingo artefakto būklei įvertinti	<p>Aprašymas – rizikos aprašymas</p> <p>Nustatymo data – data, kada rizika buvo užfiksuota</p> <p>Tikimybė – įvertinama, kokia rizikos įvykio tikimybė</p> <p>Poveikis – rizikos įvykio poveikis, pasirenkama iš enumeratoriaus reikšmių:</p> <ul style="list-style-type: none"> – Kritinis – Didelis – Vidutinis – Mažas – Trivialus <p>Prioritetas – rizikos sprendimo prioritetas, pasirenkama iš enumeratoriaus reikšmių:</p> <ul style="list-style-type: none"> 1 – aukščiausio prioriteto rizika 2 – vidutinio prioriteto rizika 3 – žemo prioriteto rizika <p>Atsakingas asmuo – nurodomas komandos žmogus, kuris yra atsakingas už šios rizikos valdymą</p> <p>Būsena – Rizikos būsena, pasirenkama iš enumeratoriaus reikšmių:</p> <ul style="list-style-type: none"> – Neužfiksuota – Užfiksuota – Peržiūrėta – Uždaryta

Esybė	Aprašymas	Atributai
		Veiksmų planas – nurodomi tolimesni žingsniai rizikos eliminavimui vykdyti Pašalinimo data – data, kai rizikos statusas įgyja reikšmę „Uždaryta“ Pašalinimo aprašymas – pateikiama pašalinimo ataskaita istorijai sekti
Rizikos kategorija	Esybę sudaro <i>Scrum</i> projekto rizikos kategorijos, šabloninis esybės objektų sąrašas pateikiamas skyrelyje 2.2.4	Pavadinimas – kategorijos pavadinimas, skyrelis 3.2.4. stulpelis „Pavadinimas“ Aprašymas – kategorijos aprašymas, skyrelis 3.2.4. stulpelis „Aprašymas“
Rizika	Esybę sudaro <i>Scrum</i> projekto rizikų tipai, šabloninis esybės objektų sąrašas pateikiamas skyrelyje 2.2.5	Pavadinimas – rizikos pavadinimas, skyrelis 3.2.5. stulpelis „Pavadinimas“
Projekto rizika	Esybę sudaro projekto artefaktų atributų ir tiems atributams priskirtų rizikų identifikaciniai numeriai. Taip pat nurodoma atributo reikšmė, kurią pasiekęs atributas tenkina nurodytos rizikos sąlygą – tampa rizikingu.	Tarpinė lentelė atributams su rizikomis apjungti Atributo reikšmė – nurodoma atributo reikšmė, kurią įgijęs atributas tampa rizikingu

3.2. Esybės „Rizika“ būsenų modelis

Rizikų būsenų diagrama nurodo visus įmanomus rizikos objekto būsenos atvejus viso rizikos gyvavimo ciklo laikotarpiu. Pakeitus rizikos būseną yra atnaujinama duomenų bazės lentelė „Projekto artefakto rizika“, sekanti konkrečių rizikos objektų duomenų pokytį. Rizikos gyvavimo ciklas projekte pateikiamas 19 pav. ir aprašomas 15 lentelėje.



19 pav. Projekto artefakto rizikos būsenų diagrama

15 lentelė. Rizikos objekto gyvavimo ciklas

Nr.	Būsena	Aprašymas
1.	Neidentifikuota	Rizika yra neidentifikuota sistemos
2.	Identifikuota - neužfiksuota	Rizika yra identifikuota sistemos, tačiau neužfiksuota sistemos naudotojo – neatliktas veiksmas „Fiksuoti riziką“

Nr.	Būsena	Aprašymas
3.	Identifikuota - užfiksuota	Atliekamas veiksmas „Fiksuoti riziką“
4.	Identifikuota - peržiūrėta	Atnaujinamos rizikos atributų reikšmės
5.	Identifikuota - uždaryta	Rizika eliminuojama ir uždaroma

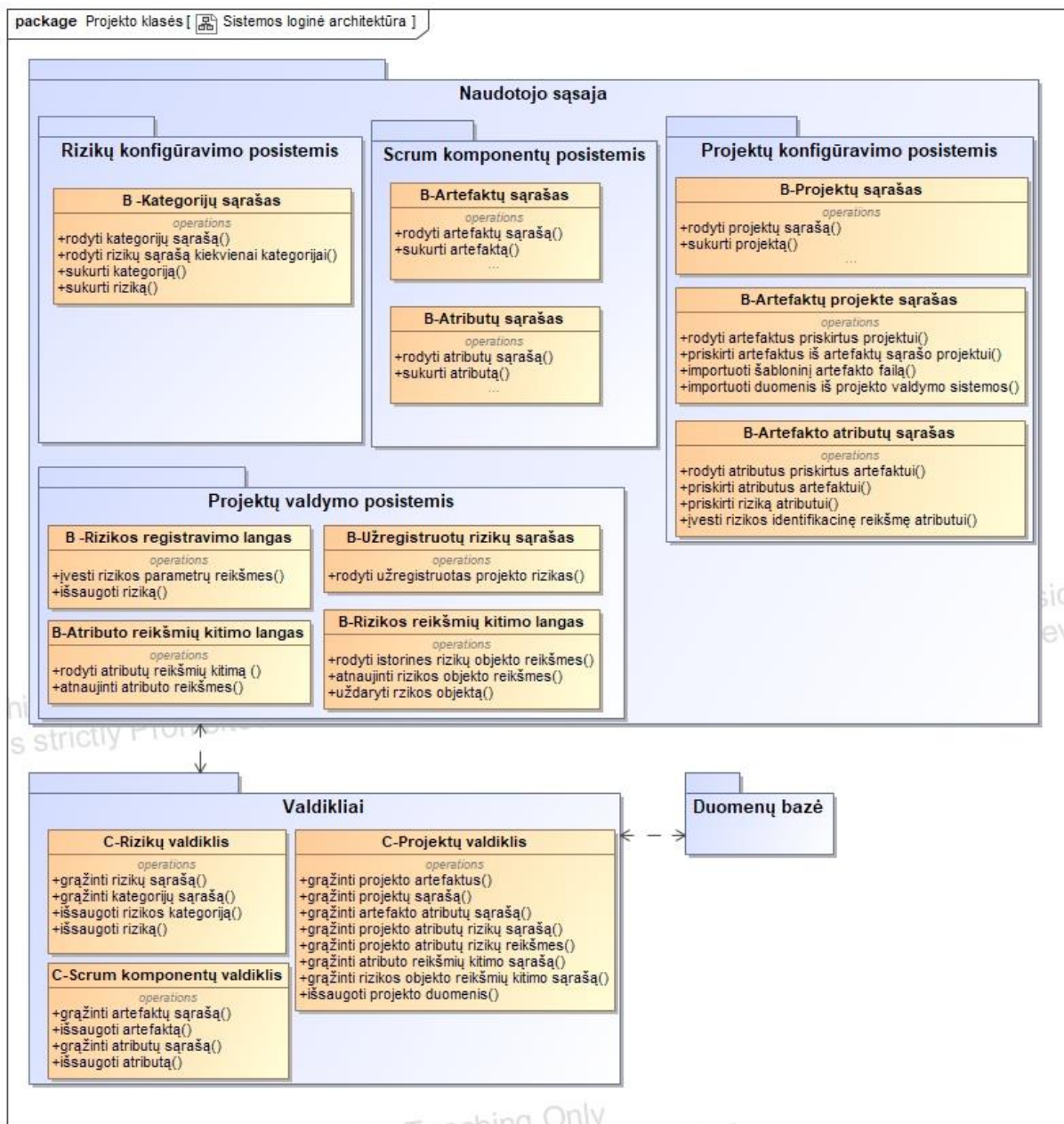
3.3. Sistemos loginė architektūra

Trijų lygių loginės architektūros komponentai yra suskirstoma į posistemius. Naudotojo sąsają sudaro keturi posistemiai, plačiau aprašomi 3.4 skyriuje. Valdiklių lygis yra sudarytas iš trijų valdiklių - rizikų valdiklis išsaugo rizikas bei rizikų kategorijas ir perduoda duomenis, susijusius su rizikų klasifikatoriumi duomenų bazei, *Scrum* komponentų valdiklis išsaugo artefaktus bei jiems priklausančius atributus ir perduoda duomenis, susijusius su *Scrum* komponentų klasifikatoriumi duomenų bazei. Projektų valdiklis išsaugo visą u konkrečiu projektu susijusią informaciją – projekto parametrus, priskirtus artefaktus, atributus, rizikas, istorinius duomenų pokyčius. Visų valdiklių bei naudotojo sąsajos langų funkcijos plačiau aprašomos 16 lentelėje. Duomenų bazės struktūra pateikiama 3.1 skyriuje klasių modelyje. Sistemos loginė architektūra pateikiama 20 pav.

16 lentelė Sistemos valdiklių aprašas

Esybė	Operacijos
<i>Controller</i> - Rizikų valdiklis	<ol style="list-style-type: none"> <i>grąžinti rizikų sąrašą</i> – grąžinamas sąrašas, sukurtas 2.2.5 skyrelyje <i>grąžinti kategorijų sąrašą</i> – grąžinamas sąrašas, sukurtas 2.2.4 skyrelyje <i>išsaugoti rizikos kategoriją</i> <i>išsaugoti riziką</i>
<i>Controller</i> – <i>Scrum</i> komponentų valdiklis	<ol style="list-style-type: none"> <i>grąžinti artefaktų sąrašą</i> – grąžinamas sąrašas, sukurtas 2.2.2 skyrelyje <i>grąžinti atributų sąrašą</i> – grąžinamas sąrašas, sukurtas 2.2.3 skyrelyje <i>išsaugoti artefaktą</i> <i>išsaugoti atributą</i>
<i>Controller</i> - Projektų valdiklis	<ol style="list-style-type: none"> <i>grąžinti projektų sąrašą</i> – naudotojui grąžinami projektai, sukurti 2.2.10 skyrelyje <i>grąžinti projekto artefaktus</i> – grąžinami projekto artefaktai, priskirti 2.2.10 skyrelyje <i>grąžinti artefakto atributų sąrašą</i> – grąžinamos atributų sąrašas, priskirtas 2.2.7 skyrelyje <i>grąžinti projekto atributų rizikų sąrašą</i> – grąžinamos atributo rizikos, priskirtos 2.2.8 skyrelyje <i>grąžinti atributo reikšmių kitimo sąrašą</i> – grąžinamos atributo reikšmės, atnaujinamos 2.2.15 skyrelyje <i>grąžinti rizikos objekto reikšmių kitimo sąrašą</i> – grąžinamos rizikos objekto reikšmės, atnaujinamos 2.2.20 skyrelyje <i>grąžinti projekto atributų rizikų reikšmes</i> – grąžinamos rizikos identifikacinės reikšmės, priskirtos 2.2.13 skyrelyje <i>išsaugoti projekto duomenis</i>
<i>Boundary</i> – Kategorijų sąrašas	<ol style="list-style-type: none"> <i>rodyti kategorijų sąrašą</i> – naudotojui rodomas rizikų klasifikatorius <i>rodyti rizikų sąrašą kiekvienai kategorijai</i> – naudotojui rodomas rizikų klasifikatorius <i>sukurti kategoriją</i> – atliekamas žingsnis <i>sukurti riziką</i> – atliekamas žingsnis

Esybė	Operacijos
<i>Boundary</i> – Artefaktų sąrašas	<ol style="list-style-type: none"> 1. rodyti artefaktų sąrašą – naudotojui rodomas artefaktų tipų sąrašas 2. sukurti artefaktą – atliekamas žingsnis, aprašytas 2.2.2 skyrelyje
<i>Boundary</i> – Atributų sąrašas	<ol style="list-style-type: none"> 1. rodyti atributų sąrašą – naudotojui rodomas artefaktų tipų sąrašas 2. sukurti atributą – atliekamas žingsnis, aprašytas 2.2.3 skyrelyje
<i>Boundary</i> – Projektų sąrašas	<ol style="list-style-type: none"> 1. rodyti projektų sąrašą – naudotojui rodomas sukurtų projektų sąrašas 2. sukurti projektą – atliekamas žingsnis, aprašytas 2.2.10 skyrelyje
<i>Boundary</i> – Artefaktų projekte sąrašas	<ol style="list-style-type: none"> 1. rodyti artefaktus priskirtus projektui – naudotojui rodomas projektui priskirtų artefaktų sąrašas 2. priskirti artefaktus iš artefaktų sąrašo projektui – atliekamas žingsnis, aprašytas 2.2.10 skyrelyje 3. importuoti šabloninį artefakto failą – atliekamas žingsnis, aprašytas 2.2.10 skyrelyje 4. importuoti duomenis iš projekto valdymo sistemos – atliekamas žingsnis, aprašytas 2.2.15 skyrelyje
<i>Boundary</i> – Artefakto atributų sąrašas	<ol style="list-style-type: none"> 1. rodyti atributus priskirtus artefaktui – naudotojui rodomas artefaktui priskirtų atributų sąrašas 2. priskirti atributus artefaktui – atliekamas žingsnis, aprašytas 2.2.7 skyrelyje 3. priskirti riziką atributui – atliekamas žingsnis, aprašytas 2.2.8 skyrelyje 4. įvesti rizikos identifikacinę reikšmę atributui – įvedama klasės „Projekto rizika“ reikšmė „Atributo reikšmė“
<i>Boundary</i> – Rizikos registravimo langas	<ol style="list-style-type: none"> 1. įvesti rizikos parametrų reikšmes – atliekamas žingsnis, aprašytas 2.2.19 skyrelyje 2. išsaugoti riziką
<i>Boundary</i> – Užregistruotų rizikų sąrašas	<ol style="list-style-type: none"> 1. rodyti užregistruotas projekto rizikas – rodomas sąrašas rizikų, užregistruotų rizikų valdymo lange
<i>Boundary</i> – Atributo reikšmių kitimo langas	<ol style="list-style-type: none"> 1. atnaujinti atributo reikšmes – atliekamas žingsnis, aprašytas 2.2.16 skyrelyje 2. rodyti atributo reikšmių kitimą – rodomos pakitusios reikšmės po žingsnių, atliekamų 2.2.15, 2.2.16 skyreliuose užbaigimo
<i>Boundary</i> – Rizikos reikšmių kitimo langas	<ol style="list-style-type: none"> 1. rodyti istorines rizikos objekto reikšmes 2. atnaujinti rizikos objekto reikšmes – atliekamas žingsnis, aprašytas 2.2.20 skyrelyje 3. uždaryti rizikos objektą

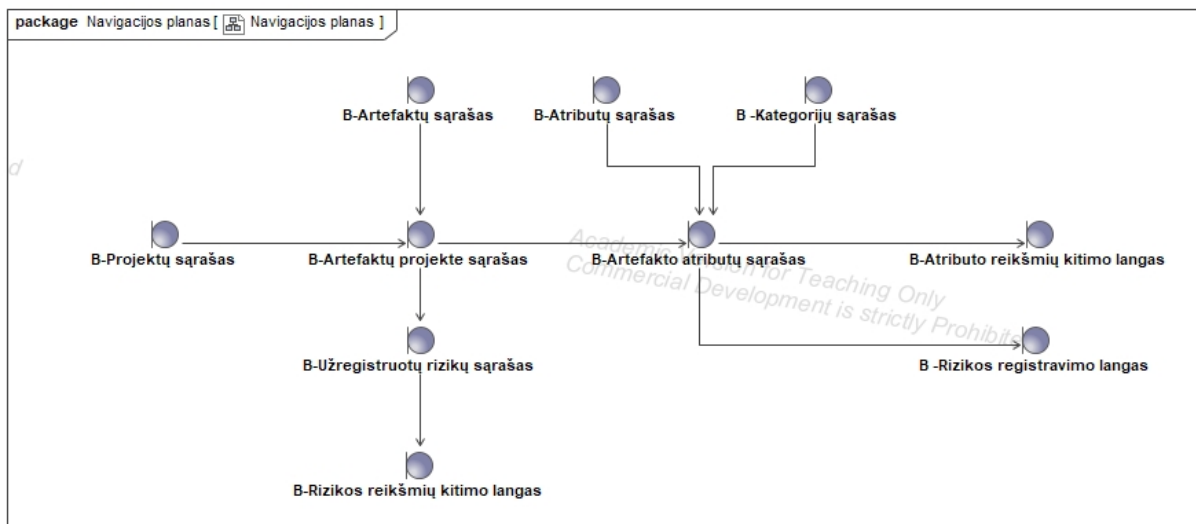


20 pav. Sistemos loginė architektūra

3.4. Naudotojo sąsajos modelis

Modelio pagrindinis procesas yra atliekamas dviejų pagrindinių rolių: rizikų dalykinės srities eksperto ir projekto dalyvio, kuris gali būti bet kuris *Scrum* komandos narys.

Rizikų valdymo sistemos navigacijos planas pateikiamas 21 pav.



21 pav. Rizikų valdymo sistemos navigacijos planas

Naudotojo sąsajos modelis yra sudarytas iš pagrindinių langų:

1. Projektų sąrašas – lange atvaizduojami sistemoje sukurti projektai, taip pat suteikiama galimybė sukurti naują projektą. Projekto kūrimo metu įvedami projekto atributai, tokie kaip pavadinimas, užsakovas, terminas. Projekto sąrašo langas pateikiamas prieduose – Priedas 5. 4 punktas;
2. Artefaktų projekte sąrašas – konkrečiam projektui pasirenkami aktualūs artefaktai iš artefaktų aibės, pateiktos 3.2.2. skyrelyje, kurių rizikų sekimą nori vykdyti projekto komanda. Taip pat, jeigu šabloninė aibė yra neišbaigta ir projekto komandai yra reikalingi papildomi artefaktai, galima sukurti naują artefaktą įvedus artefakto pavadinimą ir pridėjus jį į sąrašą. Prie projekto priskiriami artefaktai sudaro artefaktų sąrašą. Taip pat lange yra suimportuojami šabloniniai įkeliamų duomenų failai, kurie vėliau naudojami importuojamų atributų vedimui. Artefaktų projekte sąrašo langas pateikiamas prieduose – Priedas 5. 5 punktas;
3. Artefakto atributų sąrašas - projektui parinktiems artefaktams priskiriami atributai, pagal kurių reikšmes identifikuojamos rizikos projekto eigoje. Kiekvienam projekto artefaktui aktualūs atributai pridedami iš atributų sąrašo, pateikto 3.2.3. skyrelyje. Taip pat gali būti pridedami papildomi pasirinktiniai atributai pagal projekto specifiką jei šabloninis atributų sąrašas yra neišbaigtas. Kiekvienam projekto artefakto atributui yra galimybė priskirti riziką. Prie atributo pridėtas mygtukas „Pridėti riziką“ inicijuoja rizikos pridėjimą – atidaromas rizikų klasifikatorius, kuriame reikia pasirinkti, kokią riziką pridėti prie atributo. Pridėjus riziką užpildoma atributo reikšmė, identifikuojanti riziką, pvz. sprinto pajėgumo standartinio nuokrypio (atributo) pokytis lygus 10 naudotojo pasakojimo taškų (įvertis) gali būti identifikuojamas kaip sprinto prognozavimo rizika. Artefaktų atributų sąrašo langas pateikiamas prieduose – Priedas 5. 6 punktas;
4. Atributų reikšmių kitimo langas – šiame lange galima matyti atributų kitimą, iššauktą kiekvieno duomenų atnaujinimo įvykio. Taip pat lange galima priskirti atributo reikšmę įvedamiems ar iš reikšmių pasirenkamiems atributams. Atributų reikšmių kitimo langas pateikiamas prieduose – Priedas 5. 9 punktas;

5. Rizikos užfiksavimo langas – inicijavus rizikos užfiksavimą konkrečiam artefakto atributui įvedamos klasės „Projekto artefakto rizika“ atributų reikšmės. Rizikos užfiksavimo langas pateikiamas prieduose – Priedas 5. 7 punktas;
6. Užregistruotų rizikų sąrašas – užfiksavus riziką, ji atvaizduojama užregistruotų rizikų sąrašė kartu su visais kitais užregistruotais rizikos objektais rizikų švieslentėje. Užregistruotų rizikų sąrašo langas pateikiamas prieduose – Priedas 5. 8 punktas;
7. Rizikos reikšmių kitimo langas – užfiksavus riziką ir pasirinkus ją užregistruotų rizikų sąrašė, šiame lange galima inicijuoti rizikos parametrų atnaujinimą – atidaromas langas, aprašytas 4 punkte su užfiksuotomis reikšmėmis, kurias galima atnaujinti. Šiame lange taip pat galima fiksuoti rizikos uždarymą. Rizikos reikšmių kitimo langas pateikiamas prieduose – Priedas 5. 10 punktas.

Taip pat sistemoje egzistuoja sistemos *Scrum* komponentų bei rizikų klasifikatoriaus kūrimo langai, nurodyti navigacijos plane, kadangi jie yra pasiekiami iš kiekvieno sistemos lango per viršutinę meniu juostą:

1. Artefaktų sąrašas – įvedami artefaktai bei jų aprašymai. Artefaktų sąrašo langas pateikiamas prieduose – Priedas 5. 1 punktas;
2. Atributų sąrašas – įvedami atributai bei jų duomenų tipai. Atributų sąrašo langas pateikiamas prieduose – Priedas 5. 2 punktas;
3. Rizikų kategorijų sąrašas - įvedamos kategorijos, aprašytos 3.2.4. skyrelyje. Sukurtos rizikų kategorijos sudaro kategorijų sąrašą. Rizikų kategorijų sąrašo langas pateikiamas prieduose – Priedas 5. 3 punktas.

3.5. Realizacijos modelis

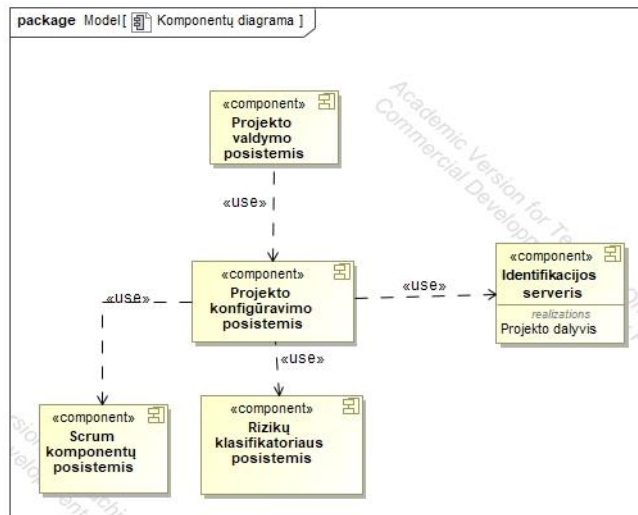
Metodika realizuota naudojant šiuos įrankius:

1. „Figma“ – sukuriami naudotojo sąsajos langai, kurie apjungiami naudotojo kelionės atvaizdavimui prototipe;
2. „Jira“ – eksportuojami eksperimentuose naudojamų projektų duomenys „Excel“ formatu;
3. „Excel“ – sukuriami artefaktai bei atributai, atliekami atributų skaičiavimai pagal „Jira“ duomenų failus bei identifikuojamos rizikos pagal nustatytas rizikos identifikacines reikšmes.

3.5.1. Komponentų diagrama

Projektuojamoje rizikų valdymo sistemoje yra numatyti šie posistemiai (22 pav.):

1. Identifikacijos serveris – renkama informacija apie asmenį, dirbantį su sistema (aktuali „Atsakingas asmuo“ atributui klasėje „Projekto artefakto rizika“);
2. Projekto konfigūravimo posistemis – renkama projekto rizikų valdymo moduliui reikalinga informacija;
3. Projekto valdymo posistemis – renkama informacija apie projekto artefaktų objektų reikšmes;
4. Rizikų klasifikatoriaus posistemis – renkama informacija apie *Scrum* projektų rizikas ir jų kategorijas;
5. *Scrum* komponentų posistemis – renkama informacija apie *Scrum* projektų artefaktus ir jų atributus.



22 pav. Rizikų valdymo sistemos komponentų diagrama

4. Eksperimentinis tyrimas

Remiantis 4 skyriuje aprašyta metodika buvo suprojektuotas sprendimas, kuriame buvo atlikti eksperimentai trims skirtingiems projektams metodikai patikrinti ir įvertinti. Pirmajame projekte yra kuriama internetinė platforma darbuotojų bei kandidatų vertinimo sistema siekiant atlikti skaidrų ir objektyvų darbuotojų vertinimo procesą taip išlaikant įmonės reputaciją bei darbuotojų moralę. Antrajame projekte kuriama banko elektroninės ir mobilios bankininkystės platforma vienam Karibų bankui, kurioje sukuriamos įvairios galimybės naudotojams sekti savo balansą, atsidaryti papildomas sąskaitas, atlikti vietines ir tarptautines pinigų perlaidas. Šiame projekte labai aktualus duomenų tikslumas bei validacijų korektiškumas, siekiant išvengti neteisėto disponavimo pinigais. Trečiajame projekte kuriama darbuotojų darbo užmokesčio skaičiavimo ir personalo valdymo sistema, kurioje yra sukuriama galimybė valdyti įmonės personalo paskyrimus, asmeninius duomenis, atlikti darbo laiko stebėseną bei kontrolę, valdyti darbo užmokesčius.

Kiekviename projekte *Scrum* meistras ir *Scrum* komanda pasirenka, kuriuos artefaktus nori sekti savo projekte pagal projekto specifiką. Artefaktai jau yra susieti su jiems priskirtais atributais ir rizikomis, tačiau šie sąrašai gali būti praplečiami ar susiaurinami projekto ribose pagal komandos poreikius. Visi eksperimentai atliekami pagal metodiką suprojektuotoje sistemoje, kurioje yra apibrėžti numatytieji artefaktų, atributų, rizikų kategorijų bei rizikų klasifikatoriai, t.y. įgyvendinti 2.2.6 metodikos žingsniai.

4.1. Projektas „DVS“: Darbuotojų atrankos ir vertinimo platforma

Realizuota rizikų valdymo sistema pritaikoma projekte, kuriame yra vystoma asmenų įdarbinimo bei darbuotojų vertinimo platforma. Ši platforma naudojama siekiant įvertinti darbuotojų bei kandidatų pagrindines stipriąsias savybes bei tobulintinas sritis ir pagal tai parinkti tinkamiausią darbo sritį tam tikram asmeniui. Taip pat įvertinama organizacijos dinamika - vadovavimo strategijos kūrimas ir naudojimas leidžia organizacijai siekti savo tikslų sąmoningai kuriant individualius vadovavimo stilius, komandos dinamiką ir organizacinę kultūrą.

Platforma veikia individualių testų bei apklausų metodu, kuriuos užpildo suinteresuotieji asmenys. Pagal testų rezultatus generuojamos ataskaitos, švieslentės, formuojami patarimai asmeniniam bei profesiniam tobulinimuisi.

4.1.1. Žingsnis „1. Apibrėžti projekto specifiką“

Sukurtoje sistemoje pasirenkami artefaktai, kurie bus sekami projekto vystymo metu. Kadangi pasirenkamas standartizuotas projektas, projekto dalyviui lieka tik pasirinkti naudojamus artefaktus ir naudojamą projektų vystymo sistemą, o artefaktų sąsaja su atributais bei atributų su rizikomis yra suteikta automatiškai.

Komandos dalyviui kuriant naują projektą sistemoje reikia apibrėžti projekto objekto pagrindinius atributus bei pasirinkti sistemą, su kurios duomenimis bus dirbama sistemoje. Projekto „DVS“ objekto atributų reikšmės nurodytos 17 lentelėje.

17 lentelė. Projekto „DVS“ aprašymas

Nr.	Klasės atributas	Reikšmė
1.	Užsakovas	Klientas „DVS“
2.	Pavadinimas	Projektas „DVS“
3.	Projekto pradžia	2022.01.01
4.	Terminas	2024.12.31
5.	Apimtis	3 metai
6.	Aprašymas	Projektas, skirtas sukurti internetinę platformą darbuotojų bei kandidatų vertinimui siekiant atlikti skaidrų ir objektyvų darbuotojų vertinimo procesą taip išlaikant įmonės reputaciją bei darbuotojų moralę
7.	Tikslas	Sukurti darbuotojų ir kandidatų vertinimo sistemą
8.	Sistema	JIRA

Nurodžius projekto parametrus parenkami artefaktai, kurie bus sekami projekte bei kiekvienam iš šių artefaktų įkeliamas įkeliamų duomenų failo šablonas. Artefaktai parenkami atsižvelgiant į poreikį konkrečiam projektui. Projektas „DVS“ yra ilgai vystomas projektas, kuriame jau yra sukurta programinė įranga ir projektas tęsiamas siekiant ją tobulinti pridėdant papildomas funkcijas. Kiekviena nauja funkcija turi savo funkcionalumo modulį projekto valdymo sistemoje. Projekto metu periodiškai atnaujinama versija – kas antrą sprintą yra paruošiamas naujos versijos paketas bei ruošiamas produkto išleidimo planas. Taip pat projektas yra plečiamas, todėl kas mėnesį į projektą priimami nauji darbuotojai, kurie yra reguliariai apmokomi. Kadangi projektas yra vystomas ilgai, yra sukauptas didelis techninių užduočių sąrašas, kuris turi būti įvertinamas kiekvieno sprinto metu. Kiekvienam naujam funkcionalumo moduliui vystyti yra reikalingas dizainas, kadangi didžioji dalis užduočių paveikia naudotojo sąsają. Pagal šią specifiką yra surenkamas aktualių artefaktų sąrašas – produkto atliekamų darbų sąrašas bei tikslas, kiekvieno funkcionalumo modulio įgyvendinimas, techninių užduočių sąrašas, dizaino reikalavimų surinkimas, sprinto atliekamų darbų sąrašas, produkto prieaugis, produkto išleidimo planas, naujų komandos narių apmokymas. Parinktų artefaktų bei jiems priskirtų failų pavadinimai iš sistemos JIRA pateikiami 18 lentelėje.

18 lentelė. Projekto "DVS" artefaktai ir jų įkeliamų duomenų failų šablonai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>
AR2 – Produkto tikslas	<i>Backlog</i>
AR8 - Funkcionalumo modulio įgyvendinimas	<i>Epic report</i>
AR10 - Techninių užduočių sąrašas	<i>TechDebt report</i>
AR12 - Dizaino reikalavimų surinkimas	Visos atributų reikšmės įvedamos, įkeliamų duomenų failas nenaudojamas A
AR14 - Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>
AR16 – Produktio prieaugis	<i>Sprint Report</i>
AR25 - Produkto išleidimo planas	<i>Version Report</i>

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas
AR26 – Naujų komandos narių apmokymas	Visos atributų reikšmės įvedamos, įkeliamų duomenų failas nenaudojamas

4.1.2. Žingsnis „2. Pasirinkti standartizuotą projektą“

Sukūrus projektą ir pasirinkus variantą naudoti standartizuotą projekto struktūrą, šiam projektui yra sukuriama posistemis. Pasirinkus artefaktus ir išsaugojus savo pasirinkimą, projekte yra priskiriami artefaktai, šių artefaktų atributai pagal standartizuotą šabloną, atributams yra priskiriamos rizikos.

Projekte nagrinėjamų artefaktų bei atributų kiekis pateikiamas 19 lentelėje.

19 lentelė. Projekto „DVS“ naudojamų komponentų kiekybiniai rodikliai

Elementas	Kiekis	Viso kiekis klasifikatoriuje	% priskirtų nuo viso kiekio klasifikatoriaus
Artefaktų kiekis	9	12	75
Unikalių atributų kiekis	36	45	80
Atributų kiekis	67	45	149
Rizikų kategorijų kiekis	17	18	95
Unikalių rizikų kiekis	48	58	83
Priskirtų rizikų kiekis	221	58	381

4.1.3. Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“

Sukurtam projektui yra pasirinkta JIRA sistema, todėl šiam projektui jau yra sukonfigūruoti šabloniniai įkeliamų duomenų failai kiekvienam artefaktui iš projekto valdymo sistemos su numatytais laukais, tačiau šie laukai taip pat gali būti konfigūruojami naudotojo.

Artefaktų sąsaja su įkeliamų duomenų failais nurodyti

20 lentelėje.

20 lentelė. Artefaktų įkeliamų duomenų failai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas	Naudojami įkeliamų duomenų failo laukai
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>	<i>Issue ID, Issue Type, Labels, Priority, Epic Link Summary, Sprint</i>
AR2 – Produkto tikslas	<i>Backlog</i>	<i>Epic Link Summary, Sprint Rankiniu būdu įvedami laukai</i>
AR8 - Funkcionalumo modulio įgyvendinimas	<i>Epic report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i>
AR10 - Techninių užduočių sąrašas	<i>TechDebt report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i>

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas	Naudojami įkeliamų duomenų failo laukai
AR12 - Dizaino reikalavimų surinkimas	Visos atributų reikšmės įvedamos, įkeliamų duomenų failas nenaudojamas	Rankiniu būdu įvedami laukai
AR14 - Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR16 – Produkto prieaugis	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR25 - Produkto išleidimo planas	<i>Version Report</i>	<i>Issue ID, Labels, Stories of Epic count, Fix versions</i>
AR26 – Naujų komandos narių apmokymas	Visos atributų reikšmės įvedamos, įkeliamų duomenų failas nenaudojamas	Rankiniu būdu įvedami laukai

4.1.4. Žingsnis „5. Priskirti rizikos reikšmę artefaktų atributams“

Sukūrus projektą ir pasirinkus projekto artefaktus reikia priskirti rizikos identifikatoriaus reikšmę visoms unikalioms rizikoms, o pasikartojančioms rizikoms šios reikšmės yra priskiriamos automatiškai. Naudotojui yra pateikiamas visų projekte naudojamų unikalų rizikų sąrašas, atrinktas pagal pasirinktus artefaktus ir joms yra priskiriama skaitinė arba enumeratoriaus reikšmė pagal atributo tipą.

Projekte „DVS“ naudojamų rizikų identifikatorių reikšmės yra pateikiamos prieduose.

4.1.5. Žingsnis „5. Importuoti projekto duomenis“

Duomenų importas vykdomas kiekvienam artefaktui atskirai į sistemą įkeliant duomenų iš projektų valdymo sistemos failą. Net jei skirtingų artefaktų atributų aibės yra persidengiančios, kitiems artefaktams atributų reikšmės nėra atnaujinamos pagal kito artefakto duomenis, tam, kad būtų išlaikomas vientisumas tarp importuojamų ir įvedamų atributų vieno artefakto režiuose. Tačiau skirtingiems artefaktams galima pernaudoti ir įkelti tą patį failą (projekto „DVS“ atveju tai būtų artefaktai AR14 ir AR16, naudojantys tą patį įkeliamų duomenų failą) tam, kad nereikėtų generuoti skirtingų failų, talpinančių identišką informaciją.

4.1.6. Žingsnis „6. Pridėti reikšmes artefaktų atributams“

Įvedamos reikšmės pridedamos tuo pačiu metu, kai yra importuojamas failas konkrečiam artefaktui, kadangi skaičiavimuose yra naudojamos funkcijos gali apimti vertes iš įvedamų ir importuojamų duomenų aibių, todėl šios abi aibės turi būti iš to paties laiko intervalo siekiant išvengti duomenų nevienareikšmiškumo. Naudotojui po failo įkėlimo pateikiama forma, kurioje reikia suteikti reikšmes įvedamiems artefaktams. Nesuteikus tų reikšmių, gali būti nepaskaičiuoti išvestiniai atributai.

Įvedamų reikšmių pavieniui naudotojas keisti negali. Tokiu atveju, kai visi atributai artefakto ribose yra įvedami, pavyzdžiui, AR12, atributų įvedimas ir atnaujinimas nėra apribojamas duomenų įkėlimo veiksmo, tačiau visų atributų reikšmės turi būti pakeičiamos vienu metu.

4.1.7. Žingsnis „7.1. Identifikuoti rizikos objektą“

Importavus duomenis iš projekto valdymo sistemos, šiuo atveju JIRA, sistema informuoja naudotoją apie atpažintas rizikas pagal įvertintus duomenis. Projekto dalyvis vėliau nusprendžia, ar atpažintas rizikingas objektas turi būti užfiksuotas, ar atributo vertės nukrypimas nuo normos (rizikingos reikšmės įgavimas) nėra laikomas rizika. Pavyzdys - staigus sprinto pajėgumo sumažėjimas, eksperimento atveju AR14 artefakto pajėgumo atributo kritimas 88 sprinto metu, nėra laikomas rizika, kadangi tai buvo numatyta dėl Kalėdų laikotarpio ir daugelio komandos narių atostogų.

21, 22 lentelėse pateikiami kiekybiniai projekto „DVS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

21 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių su duomenų atnaujinimu kiekis	Įvykių su atpažintomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	12	8	75
AR2 – Produkto tikslas	12	8	75
AR8 - Funkcionalumo modulio įgyvendinimas	4	4	100
AR10 - Techninių užduočių sąrašas	12	4	33
AR12 - Dizaino reikalavimų surinkimas	4	3	75
AR14 - Sprinto atliekamų darbų sąrašas	24	12	50
AR16 – Produkto prieaugis	24	12	50
AR25 - Produkto išleidimo planas	10	6	60
AR26 – Naujų komandos narių apmokymas	12	5	42

22 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su atpažintomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	8	24	17	12	5	63
AR2 – Produkto tikslas	5	17	17	7	1	20
AR8 - Funkcionalumo modulio įgyvendinimas	14	56	30	16	11	79

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su atpažintomis rizikomis	
					Kiekis	%
AR10 - Techninių užduočių sąrašas	7	14	8	7	3	43
AR12 - Dizaino reikalavimų surinkimas	4	12	9	7	2	50
AR14 - Sprinto atliekamų darbų sąrašas	13	35	24	12	5	38
AR16 – Produkto prieaugis	9	36	24	12	5	56
AR25 - Produkto išleidimo planas	5	20	20	11	2	40
AR26 – Naujų komandos narių apmokymas	5	7	7	4	1	20

23 lentelėje pateikiama informacija, kiek iš visų unikalių rizikų, priskirtų artefakto atributams, buvo atpažinta darbo su projektu metu.

23 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui

Artefaktas	Atpažintų rizikų kiekis	Atpažintų unikalių rizikų kiekis	% nuo viso priskirtų unikalių rizikų artefaktui	Atpažintų rizikų kategorijų kiekis	% nuo viso priskirtų rizikų kategorijų
AR1 – Produkto atliekamų darbų sąrašas	8	6	35	5	42
AR2 – Produkto tikslas	6	6	35	4	57
AR8 - Funkcionalumo modulio įgyvendinimas	8	8	27	6	38
AR10 - Techninių užduočių sąrašas	2	2	25	1	14
AR12 - Dizaino reikalavimų surinkimas	3	2	22	2	29
AR14 - Sprinto atliekamų darbų sąrašas	20	16	66	6	50
AR16 – Produkto prieaugis	16	15	63	8	75
AR25 - Produkto išleidimo planas	6	6	30	3	28
AR26 – Naujų komandos narių apmokymas	5	5	71	1	25

4.1.8. Žingsnis „7.2. Fiksuoti rizikos objektą“

Sistamai atpažinus rizikingą atributą naudotojas nusprendžia, ar atributo pokytis yra numatytas reiškinys, ar ne. Jei šis pokytis yra nenumatytas ir yra pasirenkama jį stebėti bei kontroliuoti, yra fiksuojama projekto rizika. Rizikai fiksuoti turi būti nurodomi visi projektavimo dalyje pateikti parametrai, tokie kaip rizikos prioritetas, atsakingas asmuo, veiksmų planas, rizikos poveikis. Užfiksuotoms rizikoms sistemoje yra sukuriamas atskiras langas, kuriame galima sekti visą rizikos gyvavimo ciklą ir istorinius įrašus nuo rizikos užfiksuotimo iki uždarymo.

24, 25, 26 lentelėse pateikiami kiekybiniai projekto „DVS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

24 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių su duomenų atnaujinimu kiekis	Įvykių su užregistruotomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	12	6	50
AR2 – Produkto tikslas	12	4	33
AR8 - Funkcionalumo modulio įgyvendinimas	4	4	100
AR10 - Techninių užduočių sąrašas	12	2	17
AR12 - Dizaino reikalavimų surinkimas	4	0	0
AR14 - Sprinto atliekamų darbų sąrašas	24	9	38
AR16 – Produkto prieaugis	24	8	33
AR25 - Produkto išleidimo planas	10	5	50
AR26 – Naujų komandos narių apmokymas	12	0	0

25 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su užfiksuotomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	8	24	17	12	5	63
AR2 – Produkto tikslas	5	17	17	7	4	80

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su užfiksuotomis rizikomis	
					Kiekis	%
AR8 - Funkcionalumo modulio įgyvendinimas	14	56	30	16	9	64
AR10 - Techninių užduočių sąrašas	7	14	8	7	2	29
AR12 - Dizaino reikalavimų surinkimas	4	12	9	7	0	0
AR14 - Sprinto atliekamų darbų sąrašas	13	30	24	12	4	31
AR16 – Produkto prieaugis	9	36	24	12	5	56
AR25 - Produkto išleidimo planas	5	20	20	11	2	40
AR26 – Naujų komandos narių apmokymas	5	7	7	4	1	20

26 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui

Artefaktas	Užfiksuotų rizikų kiekis	% nuo viso atpažintų rizikų artefaktui	Užfiksuotų rizikų kategorijų kiekis	% nuo viso atpažintų rizikų artefaktui
AR1 – Produkto atliekamų darbų sąrašas	6	75	5	100
AR2 – Produkto tikslas	3	50	1	14
AR8 - Funkcionalumo modulio įgyvendinimas	8	100	6	100
AR10 - Techninių užduočių sąrašas	2	100	1	100
AR12 - Dizaino reikalavimų surinkimas	0	0	0	0
AR14 - Sprinto atliekamų darbų sąrašas	9	45	6	100
AR16 – Produkto prieaugis	11	69	8	100
AR25 - Produkto išleidimo planas	5	83	3	100
AR26 – Naujų komandos narių apmokymas	3	60	1	100

4.1.9. Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“

Identifikavus ir užfiksavus projekto rizikas toliau seka užfiksuočių rizikų sekimas – reikšmių atnaujinimas ir atvaizdavimas rizikų švieslentėje. Reikšmių atnaujinimo procesas atliekamas tiek kartų, kiek tai teikia vertę projekto komandai ir kiek rizikos sudedamųjų dalių pokytis daro įtaką projekto eigai. Šis procesas yra vykdomas iki rizikos uždarymo.

Eksperimento metu pastebima, jog daugiausiai rizikų identifikuojama šiose kategorijose – „K1 Reikalavimų tenkinimo rizikos“, „K2 Komandos narių kompetencijų rizikos“, „K4 Apmokymo rizikos“, „K19 Žinių dalijimosi rizikos“ „K23 Su darbų planavimu susijusios rizikos“. Rizikos šiose kategorijose yra pasikartojančios, aptinkamos skirtinguose projekto artefaktuose. Tai identifikuoja, jog projekto komandoje yra atliekamos esminės pasikartojančios klaidos, turinčios įtaką darbų planavimui, prognozavimui bei teisingam komandos narių parinkimui bei pareigų paskirstymui. Šios rizikos yra sukeltos nuolatinės darbuotojų kaitos projekte, atsakomybės stokos bei netinkamų darbuotojų apmokymų procesų, todėl įvertinus rizikų valdymo sistemos duomenis rekomenduojama į rizikų aibę reaguoti kaip į visumą, o ne spręsti atskirus rizikos objektus lokaliai.

27 lentelėje pateikiama kiekviena unikali rizika, užfiksuota projekte, šios rizikos užfiksavimo kiekis, bei vidutiniai reikšmių ir intervalo iki uždarymo rodikliai.

27 lentelė. Užfiksuočių rizikų atnaujinimo rodikliai

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R1 Neapibrėžti reikalavimai	2	2	14	3
R2 Nevienareikšmiški reikalavimai	1	1	10	5
R3 Neteisingai apibrėžti reikalavimai	1	1	14	5
R5 Neatsižvelgiama į komandos narių turimas žinias – neteisingas pareigų paskirstymas	1	1	7	4
R6 Komandos nariai nepatiria iššūkių	1	1	30	4
R7 Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų	3	2	60	9
R8 Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius	1	1	30	5
R9 Apmokymo rizika – sugaištama daug laiko	2	1	70	12
R10 Apmokymo rizika – ribotos žinios apie Scrum procesus	1	1	30	4

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R11 - Apmokymo rizika - nesuteikta visa reikiama informacija naujiems darbuotojams	2	1	20	4
R11 Apmokymo rizika – nesuteikta visa reikiama informacija naujiems darbuotojams	1	1	14	2
R13 Neatidus testuotojų darbas	1	1	31	6
R22 Formalus kodo standartas – sumažintas našumas	1	1	14	6
R23 Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas	1	1	14	5
R25 Dokumentacijos sumažėjimas - nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas	2	1	30	6
R28 Dokumentacijos sumažėjimas – priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą	1	1	30	6
R29 Kodo peržiūros nebuvimas – nekokybiškas kodas	1	1	7	2
R37 Menkas kliento įsitraukimas – reikalavimų neatitikimas	2	1	62	6
R38 Menkas kliento įsitraukimas – ilgas atsakymo laukimas	1	1	14	3
R39 Menkas kliento įsitraukimas – neaiški vizija bei projekto eiga	3	2	45	8
R40 Menkas kliento įsitraukimas – atliktų užduočių perdarymas	1	1	21	5
R41 Tarpusavio priklausomybė tarp užduočių – darbų vėlavimai	1	1	14	3
R46 Per daug lygiagrečių prioritetų – neužtenka resursų	1	1	10	2
R50 Struktūrizuotos informacijos nebuvimas	1	1	31	6
R54 Versijos neatnaujinimas	1	1	5	3
R55 Versijos užduočių neatsekamumas	1	1	5	3
R56 Neproduktyvūs retrospektyviniai susitikimai	1	1	5	3
R57 Sprinto atliekamų darbų skaičiaus mažėjimas	2	1	10	4

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R59 Sprinto atliekamų darbų skaičiaus prognozės neatitikimas	3	2	21	4
R60 Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis	2	2	21	6
R61 Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių	1	1	14	2
R63 Perteklinis sprinto planas – komandos moralės kritimas	3	1	70	6

4.2. Projektas „EBS“: Elektroninės ir mobilios bankininkystės platforma

Realizuota rizikų valdymo sistema pritaikoma projekte, kuriame yra vystoma banko elektroninės ir mobilios bankininkystės platforma Karibų bankui. Šioje platformoje naudotojui yra suteikiamos funkcijos sekti savo balansą, atsidaryti papildomas sąskaitas, atlikti vietines ir tarptautines pinigų perlaidas. Ši platforma, priešingai nei projekte „DVS“, nėra nišinis kliento naudojamas ir parduodamas įrankis, o sistema, pritaikyta naudoti masiškai. Šiai sistemai aktualu greitaveika bei pajėgumas dirbti su dideliu srautu naudotojų.

4.2.1. Žingsnis „1. Apibrėžti projekto specifiką“

Sukurtoje sistemoje pasirenkami artefaktai, kurie bus sekami projekto vystymo metu. Kadangi šiam projektui taip pat pasirenkamas standartizuotas rizikų valdymo projektas, projekto dalyviui lieka tik pasirinkti naudojamus artefaktus ir naudojamą projektų vystymo sistemą, o artefaktų sąsaja su atributais bei atributų su rizikomis yra suteikta automatiškai.

Komandos dalyviui kuriant naują projektą sistemoje reikia apibrėžti projekto objekto pagrindinius atributus bei pasirinkti sistemą, su kurios duomenimis bus dirbama sistemoje. Projekto „EBS“ objekto atributų reikšmės nurodytos 28 lentelėje.

28 lentelė. Projekto „EBS“ aprašymas

Nr.	Klasės atributas	Reikšmė
1.	Užsakovas	Klientas „EBS“
2.	Pavadinimas	Projektas „EBS“
3.	Projekto pradžia	2022.06.01
4.	Terminas	2024.08.31
5.	Apimtis	2 metai 3 mėnesiai
6.	Aprašymas	Projektas, skirtas sukurti elektroninės bankininkystės platformą, kurioje banko naudotojai galėtų atlikti kasdienes, periodines bei vienkartinės su savo finansų valdymu susijusias užduotis
7.	Tikslas	Vystyti elektroninės bankininkystės sistemą

Nr.	Klasės atributas	Reikšmė
8.	Sistema	JIRA

Nurodžius projekto parametrus parenkami artefaktai, kurie bus sekami projekte bei kiekvienam iš šių artefaktų įkeliamas įkeliamų duomenų failo šablonas. Artefaktai parenkami atsižvelgiant į poreikį konkrečiam projektui. Projektas „EBS“ yra ilgai vystomas projektas, naudotojo sąsaja yra kuriama kito paslaugų tiekėjo, projekto komanda kuria sistemos loginę architektūrą bei duomenų bazę, todėl projekte nėra reikmės sekti dizaino reikalavimų surinkimo. Projekto užduotys nėra skaidomos į funkcionalumo modulius, o integracija su naudotojo sąsaja yra atliekama kiekvieno sprinto metu bei sprinto pabaigoje yra išleidžiama į produkciją, todėl projekte neskama nei funkcionalumo modulio įgyvendinimas, nei produkto išleidimo planas, o aktualūs produkto išleidimo plano atributai pridedami prie produkto prieaugio artefakto. Šiame projekte dirba pastovi komanda, nauji darbuotojai yra priimami ganėtinai retai, taip pat apmokymams yra skiriama daug laiko, todėl komandos narių apmokymo artefaktas šiam projektui neaktualus. Projekte sukauptas didelis techninių užduočių sąrašas, kuris turi būti įvertinamas kiekvieno sprinto metu. Pagal šią specifiką yra surenkamas aktualių artefaktų sąrašas. Parinktų artefaktų bei jiems priskirtų failų pavadinimai iš sistemos JIRA pateikiami 29 lentelėje.

29 lentelė. Projekto "EBS" artefaktai ir jų įkeliamų duomenų failų šablonai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>
AR2 – Produkto tikslas	<i>Backlog</i>
AR10 – Techninių užduočių sąrašas	<i>TechDebt report</i>
AR14 – Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>
AR15 – Sprinto tikslas	<i>Sprint Report</i>
AR16 – Produkto prieaugis	<i>Sprint Report</i>
AR23 – Produkto vystymo planas	<i>Timeline Board</i>
AR25 – Produkto išleidimo planas	<i>Version Report</i>

4.2.2. Žingsnis „2. Pasirinkti standartizuotą projektą“

Sukūrus projektą ir pasirinkus variantą naudoti standartizuotą projekto struktūrą, šiam projektui yra sukuriamas posistemis. Pasirinkus artefaktus ir išsaugojus savo pasirinkimą, projekte yra priskiriami artefaktai, šių artefaktų atributai pagal standartizuotą šabloną, atributams yra priskiriamos rizikos.

Projekte nagrinėjamų artefaktų bei atributų kiekis pateikiamas 30 lentelėje.

30 lentelė. Projekto „EBS“ naudojamų komponentų kiekybiniai rodikliai

Elementas	Kiekis	Viso kiekis klasifikatoriuje	% priskirtų nuo viso kiekio klasifikatoriaus
Artefaktų kiekis	7	12	58
Unikalių atributų kiekis	32	45	71
Atributų kiekis	53	45	112
Rizikų kategorijų kiekis	11	18	61
Unikalių rizikų kiekis	42	58	72
Priskirtų rizikų kiekis	117	58	201

4.2.3. Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“

Sukurtam projektui yra pasirinkta JIRA sistema, todėl šiam projektui jau yra sukonfigūruoti šabloniniai įkeliamų duomenų failai kiekvienam artefaktui iš projekto valdymo sistemos su numatytais laukais, tačiau šie laukai taip pat gali būti konfigūruojami naudotojo.

Artefaktų sąsaja su įkeliamų duomenų failais nurodyti 31 lentelėje.

31 lentelė. Artefaktų įkeliamų duomenų failai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas	Naudojami įkeliamų duomenų failo laukai
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>	<i>Issue ID, Issue Type, Labels, Priority, Epic Link Summary, Sprint</i>
AR2 – Produkto tikslas	<i>Backlog</i>	<i>Epic Link Summary, Sprint</i> Rankiniu būdu įvedami laukai
AR10 – Techninių užduočių sąrašas	<i>TechDebt report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i>
AR14 – Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR15 – Sprinto tikslas	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR16 – Produktio prieaugis	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR23 – Produkto vystymo planas	<i>Timeline Board</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai

4.2.4. Žingsnis „7.1. Identifikuoti rizikos objektą“

32, 33 lentelėse pateikiami kiekybiniai projekto „EBS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. Įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

32 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių su duomenų atnaujinimu kiekis	Įvykių su atpažintomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	10	6	60
AR2 – Produkto tikslas	10	3	30
AR10 – Techninių užduočių sąrašas	20	6	60
AR14 – Sprinto atliekamų darbų sąrašas	20	16	80
AR15 – Sprinto tikslas	20	3	15
AR16 – Produkto prieaugis	20	9	45
AR23 – Produkto vystymo planas	10	5	50

33 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su atpažintomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	6	18	12	5	4	67
AR2 – Produkto tikslas	1	8	8	2	1	100
AR10 – Techninių užduočių sąrašas	7	14	8	7	3	43
AR14 – Sprinto atliekamų darbų sąrašas	18	25	18	9	7	39
AR15 – Sprinto tikslas	7	16	10	6	2	29
AR16 – Produkto prieaugis	8	24	18	9	5	63
AR23 – Produkto vystymo planas	6	12	10	5	3	50

34 lentelė pateikiama informacija, kiek iš visų unikalių rizikų, priskirtų artefakto atributams, buvo atpažinta darbo su projektu metu.

34 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui

Artefaktas	Atpažintų rizikų kiekis	Atpažintų unikalių rizikų kiekis	% nuo viso priskirtų unikalių rizikų artefaktui	Atpažintų rizikų kategorijų kiekis	% nuo viso priskirtų rizikų kategorijų
AR1 – Produkto atliekamų darbų sąrašas	5	4	33	3	60
AR2 – Produkto tikslas	5	3	38	1	50
AR10 – Techninių užduočių sąrašas	8	6	75	5	71
AR14 – Sprinto atliekamų darbų sąrašas	18	14	78	7	78
AR15 – Sprinto tikslas	5	3	30	1	17
AR16 – Produkto prieaugis	7	7	39	5	56
AR23 – Produkto vystymo planas	3	3	30	2	40

4.2.5. Žingsnis „7.2. Fiksuoti rizikos objektą“

Sistamai atpažinus rizikingą atributą naudotojas nusprendžia, ar atributo pokytis yra numatytas reiškinys, ar ne. Jei šis pokytis yra nenumatytas ir yra pasirenkama jį stebėti bei kontroliuoti, yra fiksuojama projekto rizika. Rizikai fiksuoti turi būti nurodomi visi projektavimo dalyje pateikti parametrai, tokie kaip rizikos prioritetas, atsakingas asmuo, veiksmų planas, rizikos poveikis.

Užfiksuotoms rizikoms sistemoje yra sukuriamas atskiras langas, kuriame galima sekti visą rizikos gyvavimo ciklą ir istorinius įrašus nuo rizikos užfiksuotoms iki uždarymo.

35, 35, 37 lentelėse pateikiami kiekybiniai projekto „EBS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. Įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

35 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių su duomenų atnaujinimu kiekis	Įvykių su užregistruotomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	10	5	50
AR2 – Produkto tikslas	10	3	30
AR10 – Techninių užduočių sąrašas	20	4	20
AR14 – Sprinto atliekamų darbų sąrašas	20	14	70

AR15 – Sprinto tikslas	20	3	15
AR16 – Produkto prieaugis	20	9	45
AR23 – Produkto vystymo planas	10	3	30

36 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su užfiksuotomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	6	18	12	5	4	67
AR2 – Produkto tikslas	1	8	8	2	1	100
AR10 – Techninių užduočių sąrašas	7	14	8	7	4	57
AR14 – Sprinto atliekamų darbų sąrašas	18	25	18	9	7	39
AR15 – Sprinto tikslas	7	16	10	6	2	29
AR16 – Produkto prieaugis	8	24	18	9	3	63
AR23 – Produkto vystymo planas	6	12	10	5	2	33

37 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui

Artefaktas	Užfiksuotų rizikų kiekis	% nuo viso atpažintų rizikų artefaktui	Užfiksuotų rizikų kategorijų kiekis	% nuo viso atpažintų rizikų artefaktui
AR1 – Produkto atliekamų darbų sąrašas	4	80	3	100
AR2 – Produkto tikslas	3	60	1	100
AR10 – Techninių užduočių sąrašas	6	75	5	100
AR14 – Sprinto atliekamų darbų sąrašas	13	72	7	100
AR15 – Sprinto tikslas	3	60	1	100
AR16 – Produkto prieaugis	4	57	5	100
AR23 – Produkto vystymo planas	2	67	1	50

4.2.6. Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“

Identifikavus ir užfiksavus projekto rizikas toliau seka užfiksuotų rizikų sekimas – reikšmių atnaujinimas ir atvaizdavimas rizikų švieslentėje. Reikšmių atnaujinimo procesas atliekamas

tiek kartų, kiek tai teikia vertę projekto komandai ir kiek rizikos sudedamųjų dalių pokytis daro įtaką projekto eigai. Šis procesas yra vykdomas iki rizikos uždarymo.

Eksperimento metu pastebima, jog daugiausiai rizikų identifikuojama šiose kategorijose – „K3 – Bendradarbiavimo rizikos“, „K9 – Kokybės užtikrinimo rizikos“, „K10 – Kodo peržiūros rizikos“, „K11 – Formalaus kodo standarto tenkinimo rizikos“, „K12 – Su dokumentacija susijusios rizikos“. Sistemoje suteikiama galimybė į šių rizikų komplektą atsizvelgti kaip į visumą ir pritaikyti gautus duomenis tolimesnei rizikų objektų analizei vykdyti. Šio projekto rizikos yra sukeltos gerųjų praktikų ir struktūrizuotos informacijos trūkumo, kas yra susiję su techninio atliekamų darbų sąrašo palaikymu, po ko vėliau yra sukeliama su kodo kokybe susijusios problemos, ką šio projekto atveju ir identifiko rizikų valdymo sistema. Įvertinus projekto būklę rekomenduojama iškelti priežastinę problemą ir sprendimą pradėti nuo būtent šios rizikos, ir toliau atlikti stebėseną, ar pasekmės rizikos atsikartoja, ar buvo išspręstos kartu su esmine problema.

38 lentelėje pateikiama kiekviena unikali rizika, užfiksuota projekte, šios rizikos užfiksavimo kiekis, bei vidutiniai reikšmių ir intervalo iki uždarymo rodikliai.

38 lentelė. Užfiksuotų rizikų atnaujinimo rodikliai

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R6 Komandos nariai nepatiria iššūkių	1	1	7	2
R8 Bendradarbiavimo problemos tarp komandos narių – neperduodama informacija apie pokyčius	3	2	3	2
R13 Neatidus testuotojų darbas	1	1	2	2
R17 Bendradarbiavimas tarp kokybės užtikrinimo specialistų ir programuotojų – pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams	2	1	10	3
R18 – Testuotojų darbo atsilikimas nuo programuotojų sprinto metu	3	2	21	2
R21 – Formalaus kodo standarto nebuvimas – sunku suprasti kitų programuotojų kodą	1	1	14	2
R22 Formalus kodo standartas – sumažintas našumas	1	1	31	2
R23 Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas	2	1	62	4

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R24 Dokumentacijos sumažėjimas – trūksta scenarijų kokybės užtikrinimo specialistams	2	1	21	3
R25 Dokumentacijos sumažėjimas – nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas	2	1	21	3
R27 Dokumentacijos neatnaujinimas – paliekama informacija, kuri nebėra tiksli ir galiojanti esamam funkcionalumui	1	1	14	3
R28 Dokumentacijos sumažėjimas – priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą	2	1	10	3
R29 Kodo peržiūros nebuvimas – nekokybiškas kodas	1	1	10	3
R38 Menkas kliento įsitraukimas – ilgas atsakymo laukimas	1	1	7	2
R39 Menkas kliento įsitraukimas – neaiški vizija bei projekto eiga	2	1	31	2
R40 Menkas kliento įsitraukimas – atliktų užduočių perdarymas	1	1	7	3
R41 Tarpusavio priklausomybė tarp užduočių – darbų vėlavimai	2	1	7	2
R46 Per daug lygiagrečių prioritetų – neužtenka resursų	2	1	10	3
R50 Struktūrizuotos informacijos nebuvimas	3	2	7	3
R59 Sprinto atliekamų darbų skaičiaus prognozės neatitikimas	1	1	21	2
R63 Perteklinis sprinto planas – komandos moralės kritimas	1	1	14	3

4.3. Projektas „HRS“: Darbo užmokesčio apskaitos platforma

Realizuota rizikų valdymo sistema pritaikoma projekte, kuriame yra vystoma darbuotojų darbo užmokesčio skaičiavimo ir personalo valdymo sistema, kurioje yra vykdoma įmonių personalo apskaita. Užsakovas yra buhalterines paslaugas teikianti įmonė, kuri, naudodamasi savo turimu įrankiu, vykdo personalo darbo užmokesčio skaičiavimą ir personalo priežiūrą. Užsakovo poreikis – perkurti sistemą į naują įrankį, kadangi turimas įrankis yra neefektyvus ir neplačiai pritaikomas. Komandos užduotis – sukurti naują sistemą. Šis eksperimentas nuo pirmųjų dviejų skiriasi tuo, jog rizikų valdymo sistema čia yra integruojama ne į ilgai vystomą projektą, o į tik prasidėjusį, kuriame yra aktualios visos pradinės projekto vystymo fazės.

4.3.1. Žingsnis „1. Apibrėžti projekto specifiką“

Sukurtoje sistemoje pasirenkami artefaktai, kurie bus sekami projekto vystymo metu. Kadangi pasirenkamas standartizuotas projektas, projekto dalyviui lieka tik pasirinkti naudojamus artefaktus ir naudojamą projektų vystymo sistemą, o artefaktų sąsaja su atributais bei atributų su rizikomis yra suteikta automatiškai.

Komandos dalyviui kuriant naują projektą sistemoje reikia apibrėžti projekto objekto pagrindinius atributus bei pasirinkti sistemą, su kurios duomenimis bus dirbama sistemoje. Projekto „HRS“ objekto atributų reikšmės nurodytos 39 lentelėje.

39 lentelė. Projekto „HRS“ aprašymas

Nr.	Klasės atributas	Reikšmė
1.	Užsakovas	Klientas „HRS“
2.	Pavadinimas	Projektas „HRS“
3.	Projekto pradžia	2023.09.01
4.	Terminas	2024.12.31
5.	Apimtis	1 metai 4 mėnesiai
6.	Aprašymas	Projektas, skirtas sukurti internetinę platformą darbuotojų apskaitos bei personalo valdymo užduotims atlikti.
7.	Tikslas	Sukurti atnaujintą darb užmokesčio bei personalo valdymo sistemą.
8.	Sistema	JIRA

Nurodžius projekto parametrus parenkami artefaktai, kurie bus sekami projekte bei kiekvienam iš šių artefaktų įkeliamas įkeliamų duomenų failo šablonas. Artefaktai parenkami atsižvelgiant į poreikį konkrečiam projektui. Projektas „HRS“ yra naujai pradėtas projektas, kuriame reikia sukurti sistemą, pritaikant ankstesnės sistemos funkcionalumą. Produktas nėra kuriamas ant ankstesnės sistemos kodo bazės, tačiau turi paveldėti egzistuojančio sprendimo logines taisykles. Sistemos funkcionalumas yra kuriamas suskaidžius sistemą į funkcionalumo vienetus. Kiekvienam naujam funkcionalumo moduliui vystyti yra reikalingas dizainas, kadangi didžioji dalis užduočių paveikia naudotojo sąsają. Dizainui yra skiriamas didelis dėmesys, kadangi vienas iš pagrindinių skausmo taškų klientui, nusprendusiam perdaryti esamą sprendimą, buvo nepatogi naudotojo sąsaja bei naudotojo patyrimas naudojantis sistema. Produkto versijos yra atnaujinamos neperiodiškai – produktas išleidžiamas po funkcionalumo vienetų įgyvendinimo. Projekte dirba nauja projekto komanda, komandos sąstatas nepastovus, vyksta darbuotojų kaita, dažnai vykdomas naujų darbuotojų apmokymas. Kadangi projektas yra pradedamas nuo nulio, projekto būklei sekti nėra aktualu techninis atliekamų darbų sąrašas, kadangi techninė produkto kokybė yra užtikrinama su kiekvienu naudotojo pasakojimu. Komandai nereikia dirbti su paveldėtu kodu, tik reikia pritaikyti loginį esamo sprendimo funkcionalumą naujai kuriamai sistemai. Taip pat sistema turi būti integruojama su trečių šalių sistemomis, pvz., Sodra, kad būtų vykdomas efektyvus duomenų perdavimas apie darbuotojo nedarbingumo laikotarpius. Produktas nėra išleidžiamas į produkciją, kol nėra sukurta minimali priimtina produkto versija, todėl šiame projekte nėra

sekamas versijos išleidimo planas – visos užduotys priklauso vienai versijai „MVP“. Pagal šią specifiką yra surenkamas aktualių artefaktų sąrašas – produkto atliekamų darbų sąrašas bei tikslas, kiekvieno funkcionalumo modulio poreikių surinkimas bei įgyvendinimas, dizaino reikalavimų surinkimas, sprinto atliekamų darbų sąrašas, produkto prieaugis, produkto išleidimo planas, naujų komandos narių apmokymas.

Pagal šią specifiką yra surenkamas aktualių artefaktų sąrašas. Parinktų artefaktų bei jiems priskirtų failų pavadinimai iš sistemos JIRA pateikiami 40 lentelėje.

40 lentelė. Projekto "HRS" artefaktai ir jų įkeliamų duomenų failų šablonai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>
AR2 – Produkto tikslas	<i>Backlog</i>
AR6 – Funkcionalumo modulio poreikių surinkimas	<i>Epic Report</i>
AR8 – Funkcionalumo modulio įgyvendinimas	<i>Epic Report</i>
AR12 – Dizaino reikalavimų surinkimas	<i>Epic Report</i>
AR14 – Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>
AR16 – Produkto prieaugis	<i>Sprint Report</i>
AR26 – Naujų darbuotojų apmokymas	Nėra importuojamų reikšmių

4.3.2. Žingsnis „2. Pasirinkti standartizuotą projektą“

Sukūrus projektą ir pasirinkus variantą naudoti standartizuotą projekto struktūrą, šiam projektui yra sukuriama posistemis. Pasirinkus artefaktus ir išsaugojus savo pasirinkimą, projekte yra priskiriami artefaktai, šių artefaktų atributai pagal standartizuotą šabloną, atributams yra priskiriamos rizikos.

Projekte nagrinėjamų artefaktų bei atributų kiekis pateikiamas 41 lentelėje.

41 lentelė. Projekto „HRS“ naudojamų komponentų kiekybiniai rodikliai

Elementas	Kiekis	Viso kiekis klasifikatoriuje	% priskirtų nuo viso kiekio klasifikatoriaus
Artefaktų kiekis	8	12	67
Unikalių atributų kiekis	40	45	89
Atributų kiekis	63	45	140
Rizikų kategorijų kiekis	12	18	67
Unikalių rizikų kiekis	31	58	53
Priskirtų rizikų kiekis	202	58	348

4.3.3. Žingsnis „4. Priskirti stulpelių reikšmes importuojamiems atributams“

Sukurtam projektui yra pasirinkta JIRA sistema, todėl šiam projektui jau yra sukonfigūruoti šabloniniai įkeliamų duomenų failai kiekvienam artefaktui iš projekto valdymo sistemos su numatytais laukais, tačiau šie laukai taip pat gali būti konfigūruojami naudotojo.

Artefaktų sąsaja su įkeliamų duomenų failais nurodyti 42 lentelėje.

42 lentelė. Artefaktų įkeliamų duomenų failai

Artefaktas	Įkeliamų duomenų failo šablono pavadinimas	Naudojami įkeliamų duomenų failo laukai
AR1 – Produkto atliekamų darbų sąrašas	<i>Backlog</i>	<i>Issue ID, Issue Type, Labels, Priority, Epic Link Summary, Sprint</i>
AR2 – Produkto tikslas	<i>Backlog</i>	<i>Epic Link Summary, Sprint</i> Rankiniu būdu įvedami laukai
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	<i>Epic Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Sprint,</i> Rankiniu būdu įvedami laukai
AR8 – Funkcionalumo modulio įgyvendinimas	<i>Epic Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Sprint,</i> Rankiniu būdu įvedami laukai
AR12 – Dizaino reikalavimų surinkimas	<i>Epic Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Sprint,</i> Rankiniu būdu įvedami laukai
AR14 – Sprinto atliekamų darbų sąrašas	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR16 – Produkto prieaugis	<i>Sprint Report</i>	<i>Issue ID, Issue Type, Labels, Priority, Story Points, Stories of Epic count</i> Rankiniu būdu įvedami laukai
AR26 – Naujų darbuotojų apmokymas		Rankiniu būdu įvedami laukai

4.3.4. Žingsnis „7.1. Identifikuoti rizikos objektą“

43, 44 lentelėse pateikiami kiekybiniai projekto „HRS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

43 lentelė. Įvykių su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių duomenų atnaujinimu kiekis	Įvykių su atpažintomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	10	7	70
AR2 – Produkto tikslas	10	5	50

Artefaktas	Įvykių duomenų atnaujinimu kiekis	Įvykių su atpažintomis rizikomis kiekis	
		Kiekis	%
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	10	9	90
AR8 – Funkcionalumo modulio įgyvendinimas	10	6	60
AR12 – Dizaino reikalavimų surinkimas	10	3	30
AR14 – Sprinto atliekamų darbų sąrašas	21	13	62
AR16 – Produktio prieaugis	21	16	76
AR26 – Naujų darbuotojų apmokymas	10	6	60

44 lentelė. Atributų su atpažintomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su atpažintomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	8	21	17	8	6	75
AR2 – Produkto tikslas	4	14	11	7	2	50
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	7	17	14	7	5	71
AR8 – Funkcionalumo modulio įgyvendinimas	11	31	18	8	6	55
AR12 – Dizaino reikalavimų surinkimas	4	16	10	7	2	50
AR14 – Sprinto atliekamų darbų sąrašas	11	39	19	9	5	46
AR16 – Produktio prieaugis	13	42	21	9	8	62
AR26 – Naujų darbuotojų apmokymas	5	22	6	2	1	20

45 lentelėje pateikiama informacija, kiek iš visų unikalių rizikų, priskirtų artefakto atributams, buvo atpažinta darbo su projektu metu.

45 lentelė. Atpažintų rizikų kiekis kiekvienam artefaktui

Artefaktas	Atpažintų rizikų kiekis	Atpažintų unikalių rizikų kiekis	% nuo viso priskirtų unikalių rizikų artefaktui	Atpažintų rizikų kategorijų kiekis	% nuo viso priskirtų rizikų kategorijų
AR1 – Produkto atliekamų darbų sąrašas	19	11	65	6	75
AR2 – Produkto tikslas	8	7	64	5	71
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	18	11	78	7	100
AR8 – Funkcionalumo modulio įgyvendinimas	18	5	28	4	50
AR12 – Dizaino reikalavimų surinkimas	3	3	30	3	43
AR14 – Sprinto atliekamų darbų sąrašas	40	8	42	5	56
AR16 – Produkto prieaugis	40	14	67	9	100
AR26 – Naujų darbuotojų apmokymas	4	3	50	2	100

4.3.5. Žingsnis „7.2. Fiksuoti rizikos objektą“

Sistamai atpažinus rizikingą atributą naudotojas nusprendžia, ar atributo pokytis yra numatytas reiškinys, ar ne. Jei šis pokytis yra nenumatytas ir yra pasirenkama jį stebėti bei kontroliuoti, yra fiksuojama projekto rizika. Rizikai fiksuoti turi būti nurodomi visi projektavimo dalyje pateikti parametrai, tokie kaip rizikos prioritetas, atsakingas asmuo, veiksmų planas, rizikos poveikis.

Užfiksuotoms rizikoms sistemoje yra sukuriamas atskiras langas, kuriame galima sekti visą rizikos gyvavimo ciklą ir istorinius įrašus nuo rizikos užfiksuotimo iki uždarymo.

46, 47, 48 lentelėse pateikiami kiekybiniai projekto „HRS“ rodikliai viso laikotarpio, kada buvo dirbama su rizikų valdymo sistema. Įvertinama, kiek įvykių (t.y. Įkeliamų duomenų failo įkėlimo ir atributų reikšmių atnaujinimo) buvo atpažintos rizikos, kuriems atributams jos buvo atpažintos bei detalizuojamas atpažintų rizikų kiekis atributo lygmenyje.

46 lentelė. Įvykių su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Įvykių su duomenų atnaujinimu kiekis	Įvykių su užregistruotomis rizikomis kiekis	
		Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	10	6	60
AR2 – Produkto tikslas	10	0	0
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	10	7	70
AR8 – Funkcionalumo modulio įgyvendinimas	10	5	50
AR12 – Dizaino reikalavimų surinkimas	10	0	0
AR14 – Sprinto atliekamų darbų sąrašas	21	11	52
AR16 – Produkto prieaugis	21	11	52
AR26 – Naujų darbuotojų apmokymas	10	3	30

47 lentelė. Atributų su užfiksuotomis rizikomis kiekis kiekvienam artefaktui

Artefaktas	Priskirtų atributų skaičius	Rizikų skaičius	Unikalių rizikų skaičius	Unikalių rizikų kategorijų skaičius	Atributų skaičius su užfiksuotomis rizikomis	
					Kiekis	%
AR1 – Produkto atliekamų darbų sąrašas	8	21	17	8	5	63
AR2 – Produkto tikslas	4	14	11	7	0	0
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	7	17	14	7	5	71
AR8 – Funkcionalumo modulio įgyvendinimas	11	31	18	8	4	36
AR12 – Dizaino reikalavimų surinkimas	4	16	10	7	0	0
AR14 – Sprinto atliekamų darbų sąrašas	11	39	19	9	4	36
AR16 – Produkto prieaugis	13	42	21	9	6	46
AR26 – Naujų darbuotojų apmokymas	5	22	6	2	1	20

48 lentelė. Užfiksuotų rizikų kiekis kiekvienam artefaktui

Artefaktas	Užfiksuotų rizikų kiekis	% nuo viso atpažintų rizikų artefaktui	Užfiksuotų rizikų kategorijų kiekis	% nuo viso atpažintų rizikų artefaktui
AR1 – Produkto atliekamų darbų sąrašas	11	58	6	100
AR2 – Produkto tikslas	0	0	0	0
AR6 – Funkcionalumo modulio aukšto lygio poreikių surinkimas	10	55	7	100
AR8 – Funkcionalumo modulio įgyvendinimas	6	33	4	100
AR12 – Dizaino reikalavimų surinkimas	0	0	0	0
AR14 – Sprinto atliekamų darbų sąrašas	10	25	5	100
AR16 – Produkto prieaugis	13	33	9	100
AR26 – Naujų darbuotojų apmokymas	3	75	2	100

4.3.6. Žingsnis „7.3. Atnaujinti rizikos objekto atributų reikšmes“

Identifikavus ir užfiksavus projekto rizikas toliau seka užfiksuotų rizikų sekimas – reikšmių atnaujinimas ir atvaizdavimas rizikų švieslentėje. Reikšmių atnaujinimo procesas atliekamas tiek kartų, kiek tai teikia vertę projekto komandai ir kiek rizikos sudedamųjų dalių pokytis daro įtaką projekto eigai. Šis procesas yra vykdomas iki rizikos uždarymo. Projekto valdymo pradžioje iškart pastebima naujiems projektams gana dažna problema – darbų apimties vertinimas naudotojo pasakojimo taškais (angl. *Story points*) nėra patikrintas praktikoje, todėl nėra aiškus kiekvieno naudotojo pasakojimo svoris. Dėl šios priežasties kyla ir sprinto prognozavimo problemos, nėra išgrynintas vieno darbuotojo etato pajėgumas naudotojo pasakojimo taškais, tai iššaukia ir didelį apimties standartinį nuokrypį, todėl mažėja tiek kliento pasitikėjimas, tiek komandos moralė. Situacijos pagerėjimas pastebimas po pirmų 10 sprintų, kai buvo įgyvendinta dalis funkcionalumo, komanda buvo giliau susipažinusi su domenu bei siekiamo sprendimo reikalavimais. Taip pat apimties ir pajėgumo skaičiavimai buvo tikslesni, kadangi buvo galima atlikti skaičiavimus iš istorinių sprintų duomenų. Šiame projekte rizikų valdymo sistema padėjo atkreipti dėmesį į problemas, kurių ir buvo tikimasi naujo projekto pradžioje, tad galima teigti, jog naujuose projektuose sistemos vertė nėra labai reikšminga. Rizikų, kurios buvo naujos komandai, identifikuota minimaliai.

49 lentelėje pateikiama kiekviena unikali rizika, užfiksuota projekte, šios rizikos užfiksavimo kiekis, bei vidutiniai reikšmių ir intervalo iki uždarymo rodikliai.

49 lentelė. Užfiksuotų rizikų atnaujinimo rodikliai

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R1 Neapibrėžti reikalavimai	2	1	10	2
R2 Nevienareikšmiški reikalavimai	1	1	10	3
R3 Neteisingai apibrėžti reikalavimai	2	2	7	3
R4 Neišpildyti reikalavimai	2	1	5	2
R5 Neatsižvelgiama į komandos narių turimas žinias – neteisingas pareigų paskirstymas	1	1	10	2
R7 Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų	2	1	7	2
R8 Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius	2	2	5	2
R9 Apmokymo rizika - sugaištama daug laiko	1	1	8	3
R10 Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus	3	2	3	2
R15 Nesklandi <i>Scrum</i> procesų integracija komandai	2	2	5	2
R16 Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams	2	1	10	3
R19 Vienetų testavimai - sunkus pritaikymas paveldėtam kodui	1	1	14	3
R21 Formalaus kodo standarto nebuvimas - sunku suprasti kitų programuotojų kodą	1	1	4	2
R34 Apimties vertinimas - užduotis nuvertinta	2	1	4	2
R35 Apimties vertinimas - užduotis vertinama pilnai nesuprantant reikalavimų	3	1	7	3
R36 Apimties vertinimas - užduotis vertinama nežinant techninių apribojimų	2	1	5	2
R37 Menkas kliento įsitraukimas - reikalavimų neatitikimas	2	1	5	2
R38 Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas	2	1	5	2
R39 Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga	2	2	30	4

Užfiksuota unikali rizika	Užfiksavimo kiekis	Kiek atributų užfiksuota	Vid. Trukmė dienomis iki uždarymo datos	Vid. Reikšmių atnaujinimo kiekis
R40 Menkas kliento įsitraukimas - atliktų užduočių perdarymas	2	1	10	3
R41 Tarpusavio priklausomybė tarp užduočių – darbų vėlavimai	3	2	4	2
R42 Tarpusavio priklausomybė tarp užduočių – apimties išaugimas	2	1	7	2
R59 Sprinto atliekamų darbų skaičiaus prognozės neatitikimas	3	2	14	3
R60 Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis	3	2	5	2
R61 Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių	1	1	5	2
R62 Neišbaigtas sprinto planas – neužtektinai darbo komandai	1	1	14	2
R64 Sprinto pajėgumo nenusėjamumas – neadekvatus prognozavimas	3	2	30	4

4.4. Eksperimento išvados

Atliktų eksperimentų tikslas buvo nustatyti, kiek ir kokių rizikų galima identifikuoti ir valdyti skirtingo pobūdžio *Scrum* projektuose. Šiam tikslui buvo išnagrinėti 3 skirtingo tipo bei stadijos projektai, kuriems buvo paruošta rizikų valdymo sistema.

Pirmo eksperimento metu buvo pastebėta, kad projekto komandoje yra atliekamos esminės pasikartojančios klaidos, turinčios įtaką darbų planavimui, prognozavimui bei teisingam komandos narių parinkimui bei pareigų paskirstymui, todėl čia vietoj lokalaus problemos sprendimo siūloma atsižvelgti į situaciją plačiau ir įvertinti pasikartojančias atskiras rizikas kaip vieną visumą.

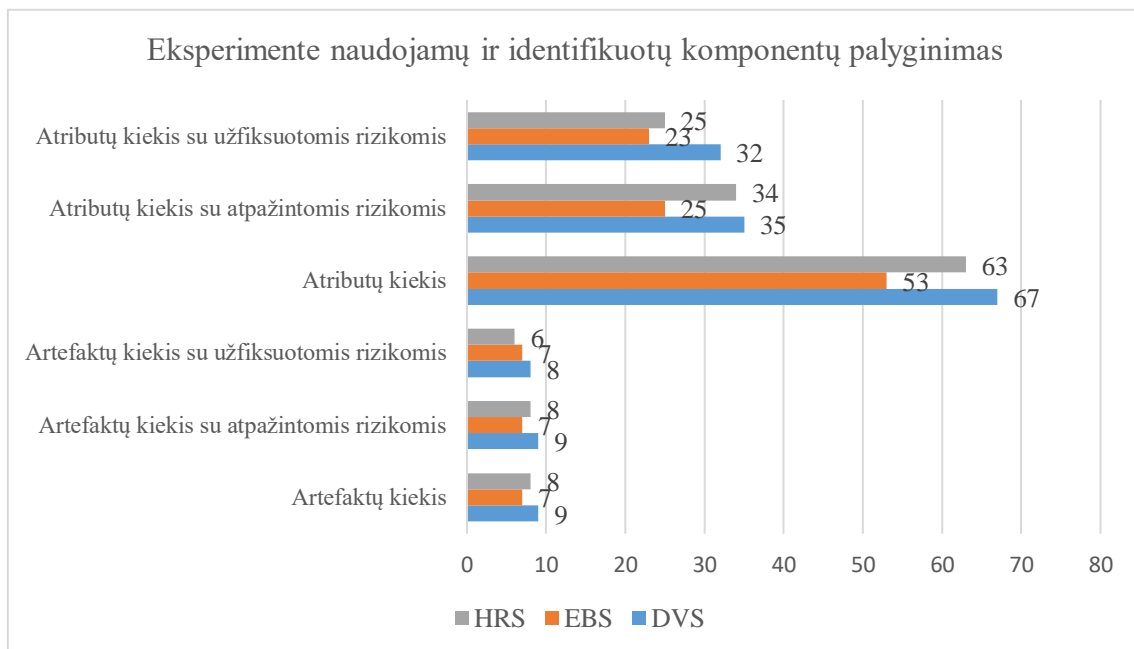
Antrojo eksperimento metu pastebima, jog dauguma rizikų gali būti laikomos išvestinėmis, t.y. atsiradusios dėl vienos probleminės vietos – struktūrizuotos informacijos bei sklandžios komunikacijos tarp kolegų nebuvimo, todėl šiam projektui tinkamas problemos sprendimo būdas būtų akcentuoti priežastį, o ne tvarkyti iš pasekmės kilusias problemas.

Trečio eksperimento metu pastebima, kad rizikų valdymo sistema identifikuoja klasikines naujo projekto problemas, kurios buvo pažįstamos komandai ir pastebėtos kituose naujuose projektuose, todėl galima teigti, jog ši sistema daugiau vertės duoda ilgiau besivystančiuose projektuose, kuriuose naujos rizikos gali būti praleistos ar ignoruojamos įsidirbusios komandos.

Eksperimentų metu naudojamas ir identifikuotas elementų tipų kiekis % lyginant su numatytu elementų tipų kiekiu metodikoje:

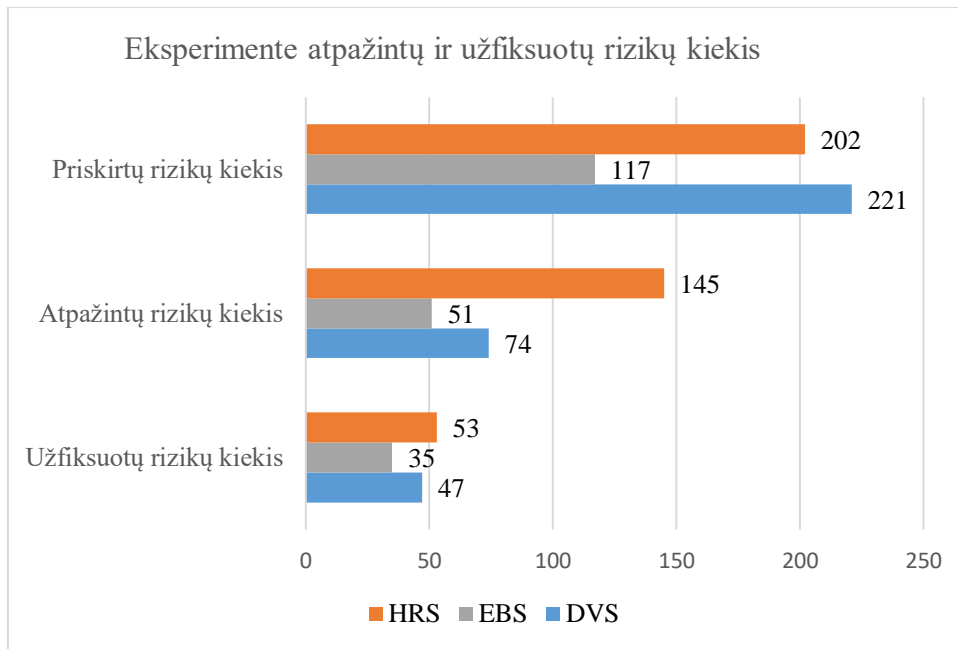
- Artefaktų iš klasifikatoriaus – 100%
- Atributų iš klasifikatoriaus – 100%
- Atpažinta rizikų iš klasifikatoriaus – 100%
- Užfiksuota rizikų iš klasifikatoriaus – 83%

Sukūrus projekto prototipą, pradinė projektinėje dalyje suformuota artefaktų aibė buvo stipriai sumažinta, ir taip įsitikinama, jog analitinėje tiriamojo darbo dalyje apibrėžta artefaktų aibė yra išbaigta ir papildomo skaidymo šiems artefaktams atlikti nereikia. Projekto būklę galima sekti naudojantis pagrindiniais artefaktais jiems pridodant atributus, kurių validacijos identifikuoja projekto būklę. Papildomas artefaktas gali būti kuriamas esant ypatingai projekto specifikai. 23 pav. pavaizduojami eksperimentų metu projektuose naudojami artefaktai, jų atributai bei šių komponentų naudojimo palyginimas pagal gautus rezultatus.



23 pav. Eksperimente naudojamų ir identifikuotų komponentų palyginimas

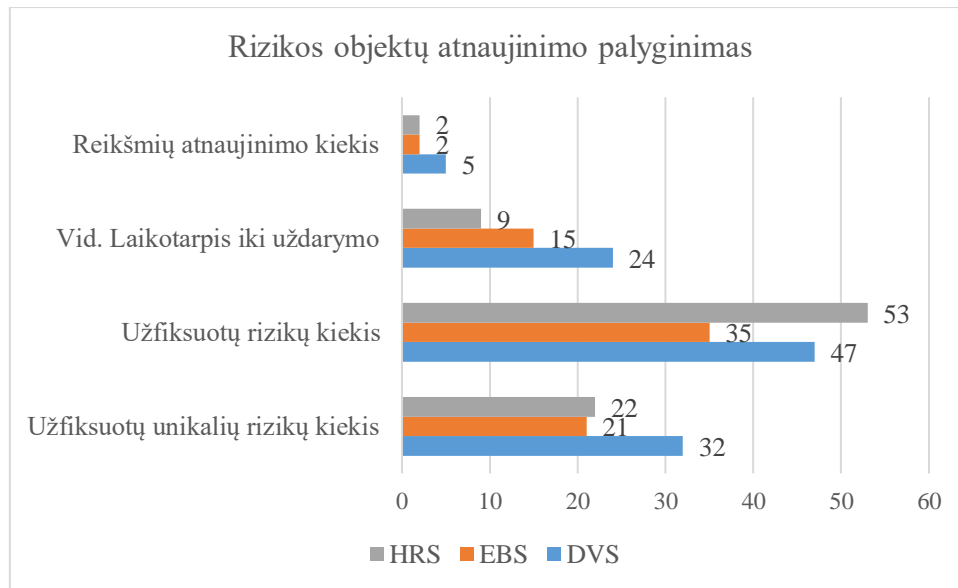
Eksperimente atpažintų bei užfiksuotų rizikų kiekis atvaizduotas 24 pav.



24 pav. Eksperimente atpažintų ir užfiksuotų rizikų kiekis

Skirtingiems projekto artefaktams buvo priskirti pasikartojantys atributai, kadangi vienas atributas gali būti aktualus kelių artefaktų būklei sekti. Importuojant duomenis iš projektų valdymo sistemos, skirtingiems artefaktams buvo fiksuojamos tos pačios rizikos dėl pasikartojančių atributų rizikingos reikšmės indikacijos. Tai parodo, jog sistemoje naudojant kelis panašaus pobūdžio artefaktus, renkama informacija gali tapti pertekline, o rizikingi įvykiai tampa dubliuoti. Tam išvengti, komandos dalyvis turi įvertinti, kokios rizikos jau buvo užfiksuotos projekte, bei dubliuotus įvykius praleisti.

Užfiksuotos rizikos ne visais atvejais išsprendavo iki kito Įkeliamų duomenų failo įkėlimo įvykio, todėl vykdant tolimesnį importą, rizika sistemoje vėl identifikuojama dėl rizikingos atributo reikšmės. Sistemoje nėra atsižvelgiama, kad šiam atributui praėjusio Įkeliamų duomenų metu jau buvo užfiksuota rizika ir šiuo metu ji yra sprendžiama. Papildomas rizikingo atributo atpažinimas gali būti naudingas komandai, kadangi tai parodo atributo būklę, ar situacija gerėja, tačiau čia reikia įvertinti, kad rizika jau yra užfiksuota ir to daryti pakartotinai nereikia.



25 pav. Rizikos objektų atnaujinimo palyginimas

4.4.1. Ekspertų nuomonė apie metodikos naudingumą

Buvo atliktas interviu penkių skirtingų pozicijų darbuotojams, kurie užima skirtingas roles, tačiau visi dalyvauja rizikų identifikavimo bei valdymo procesuose *Scrum* projektuose. Šie žmonės buvo pasirinkti dėl skirtingos patirties su IT projektais, dėl skirtingų pareigų bei požiūrio į rizikų valdymo procesą. Apklausoje dalyvavę asmenys aprašyti 50 lentelėje. Apklausoje dalyviams buvo pristatyta metodika, parodomas prototipas bei skaičiavimų failas, paaiškinta, kokios sistemos veikimo principai, kaip apskaičiuojami artefaktų atributai bei identifikuojamos jų rizikos. Pristatymo pabaigoje užduodami klausimai, kurie padėtų sužinoti specialistų nuomonę apie sukurtą metodiką.

50 lentelė. Apklausoje dalyvavę asmenys

Pašnekovo numeris	Vardas	Įmonė	Pozicija	Patirtis IT projektuose (metais)	Turimas IT išsilavinimas
1	Tomas	UAB „Devbridge LT“	Sistemų architektas	20	Taip
2	Skaistė	UAB „Devbridge LT“	Testuotoja	3	Ne
3	Alice	UAB „Devbridge LT“	Projekto vadovė	12	Ne
4	Simone	UAB „Devbridge LT“	Produkto dizaineris	7	Ne
5	Milda	UAB „Devbridge LT“	<i>Scrum</i> vadovė	10	Taip

Po pokalbio su penkiais skirtingais pašnekovais padarytos išvados:

1. Standartinio projekto kūrimo funkcionalumas yra labai pravarti sistemos dalis, kadangi tai drąstiškai sumažina konfigūravimo laiką, taip pat standartinis projektas yra sukuriamas

remiantis išplėstinėmis žiniomis apie rizikas projektuose bei galimas rizikų priežastis, kas daugeliui apklausoje dalyvavusių asmenų nėra intuityvu. Standartinį projektą galima lengvai modifikuoti, pridėti arba atsisakyti nereikalingų rizikų, todėl daug kas sutinka, jog naujas projektas be šablono turėtų būti kuriamas esant labai nišiniam projektui, kurio specifiška kardinaliai skiriasi nuo kitų analizuotų programinės įrangos kūrimo projektų

2. Apklausoje dalyviai išskiria privalumą, jog sukurtas sprendimas yra lengvai suprantamas, konfigūracija yra paprasta, kadangi atributų reikšmėms apskaičiuoti yra naudojamos pažįstamos formulės – tai leidžia papildyti atributų sąrašą nereikalaujant specialių žinių apie sukurtą produktą. Taip pat sistema atnaujinama kas mėnesį ar kas dvi savaites, todėl vienu iš privalumų išskiriamas ne per didelis įtraukimas.
3. Pagrindinis išskirtas metodikos trūkumas – daug atributų negali būti apskaičiuojami remiantis eksportuojamų duomenų failais, ypač susiję su darbuotojų morale, motyvacija, pervargimo problemomis bei kitais „minkštaisiais“ faktoriais, todėl šioje sistemoje taip pat neišvengiama rankinio darbo.
4. Pasak apklausos dalyvių sistemos atpažintos ir užregistruotos rizikos yra tikslios ir atitikdavo realų komandos statusą, todėl sistema yra laikoma veiksminga. Tačiau pabrėžiama, kad sistema pravarčiau naudotis jau įsivažiavusioms komandoms, kadangi būtent šiose komandose kyla neplanuotos ir nepastebėtos rizikos – didesnę patirtį ilgai besivystančiuose projektuose turintys žmonės yra labiau linkę ignoruoti projekto parametrų pakitimus ir valdyti projektą „iš inercijos“, kai naujuose projektuose komanda yra atsargesnė ir labiau įsitraukusi į rizikų identifikavimo bei valdymo procesą. Metodika šiuo atveju gali būti puiki pagalbinė priemonė, tačiau nemažai užfiksuotos informacijos yra puikiai pažįstamos.

5. Išvados

1. Išanalizavus *Agile Scrum* projektų struktūrą bei veikimo principus, buvo identifikuotos pagrindinės šių projektų sudedamosios dalys – projektų artefaktai, išanalizuoti jų veikimą apibūdinantys atributai. Nustatyta, kad reguliarius jų statusų sekimas leidžia užtikrinti ilgalaikį sėkmingą rizikų identifikavimą.
2. Išanalizavus rizikų gyvavimo ciklą bei pagrindines tendencijas valdant rizikas *Agile* projektuose, nustatyta, jog *Scrum* metodika neturi vieningos rizikų valdymo metodikos. *Scrum* projektų valdymo metodas skatina naudotis kasdienėmis ceremonijomis rizikų identifikavimui ir sekimui, tai neužtikrina nuoseklaus ilgalaikio rizikų sekimo ir analizavimo. Taip pat egzistuojantys *Scrum* rizikų valdymo metodai stinga proaktyvumo – rizikos užfiksuojamos per vėlai, taip pat trūksta automatikos, metodikose reikalaujama didelio komandos dalyvių įsitraukimo.
3. Siūlomo sprendimo projektavimo metu buvo sumodeliuota rizikų valdymo metodika, kuri pasinaudodama įkeliamų duomenų failais iš projektų valdymo sistemos identifikuoja *Scrum* artefaktus, kurie gali būti rizikingi, t.y. tam tikra artefakto sudedamoji dalis – artefakto atributas – įgauna reikšmę, kuri yra netenkinama pagal projekto konfigūracijos taisykles. Ši metodika nėra automatizuota, todėl komandos dalyvio įsitraukimas yra būtinas, tačiau tai leidžia greičiau ir anksčiau identifikuoti nepageidaujamus atributų pokyčius bei atlikti tolimesnę jų stebėseną.
4. Sukūrus prototipą eksperimentui buvo pasirinkti trys skirtingi realūs įmonės projektai, kurių trukmė yra 1 – 3 metai bei jie yra skirtingose projekto gyvavimo fazėse. Tai leido įvertinti visų metodikoje sukurtų klasifikatorių teikiamą naudą bei pritaikomumą. Lankstus artefaktų pasirinkimas projekte leidžia komandai išlaikyti *Scrum* projekto valdymo principus atliekant automatinius atributų reikšmių apskaičiavimus bei išvengiant perteklinių veiklų rizikoms valdyti.
5. Atlikus eksperimentus bei ekspertų apklausą, nustatyta, kad duomenų įkėlimas iš projektų valdymo sistemos leidžia identifikuoti rizikas, kurios neatpažįstamos trumpuose projekto vystymo intervaluose, tokiuose kaip sprintas. Nustatyta, jog tokia sistema aktualesnė seniai įsidirbusios komandoms su ilgai besivystančiais projektais, kadangi tokiuose projektuose lengva nepastebėti ar ignoruoti galimus projekto artefaktų pakitimus, kai naujuose projektuose į rizikų valdymo procesą yra kreipiamas didelis dėmesys ir sistema gali tapti pertekline.

Literatūros saraksts

1. *Manifesto for Agile Software Development*, 2001 [žiūrēta 2022-10-26]. Prieiga per: <https://Agilemanifesto.org/>
2. RUMPE, B. *Agile. Modeling with the UML*, 2002, p. 5-7
3. BOEHM, B., TURNER, R. *Management challenges to implementing Agile processes in traditional development organizations*. 2005, p. 30–39
4. ODZALY, E. E., GREER, D., & STEWART, D. *Agile risk management using software agents. Journal of Ambient Intelligence and Humanized Computing*, 2017, p. 6-8
5. PROJECT MANAGEMENT INSTITUTE, *Agile Practice Guide*, 2017, p. 10-11
6. ALQUADAH, M., *A Comparison of Scrum and Kanban for Identifying their Selection Factors*, 2018 April 5, p. 4-5
7. CHARI, K., AGRAWAL M., *Impact of incorrect and new requirements on waterfall software project outcomes*, 2018, p. 5-8
8. *Agile Adoption Statistics*, 2022 January 26, Prieiga per: <https://whattobecome.com/blog/adile-adoption-statistics/>
9. TAVARES, B. G., SANCHES da SILVA, C. E., DINIZ de SOUZA, A., *Risk management analysis in Scrum software projects*, Received 10 March 2016; received in revised form 6 January 2017; accepted 22 January 2017, p. 4-6
10. WEFLEN, E., MACKENZIE, C. A., RIVERO I. V., *An influence diagram approach to automating lead time estimation in Agile Kanban project management*, 2021 September 7, p. 2-3
11. GRANULO, A., TANOVIĆ, A., *Comparison of SCRUM and KANBAN in the Learning Management System implementation process*, 2019 November, p. 4
12. LEI, H., GANJEIZADEH, F., JAYACHANDRAN, P.K., OZCAN, P., *A statistical analysis of the effects of Scrum and Kanban on software development projects*, 2015, p. 3-4, 8.
13. SCHWABER, K., SUTHERLAND, J., *The Definitive Guide to Scrum: The Rules of the Game*, 2017, p. 3-4
14. SHALLOWAY, A., *Demystifying Kanban*, 2011 March, p. 3-4
15. BOISER, L., *Using Kanban for Risk Management in Agile Projects*, 2020 May
16. *The 2020 Scrum Guide*, 2020 [žiūrēta 2022-12-20]. Prieiga per: <https://Scrumguides.org/Scrum-guide.html>
17. BABOK. *A guide to the business analysis body of knowledge*. IIBA - International institute of . Business Analysis. 2015 p.4-5
18. SINGH, M., SAXENA, R., *Risk Management in Agile Model*, 2014, p.1-2.
19. TOMANEK, M., JURICEK, J., *Project risk management model based on prince2 and Scrum frameworks*, 2015 January, p.3-4
20. *Risk Management in Agile: How To Control Risk In Agile Project Management?* 2022 [žiūrēta 2022-12-22]. Prieiga per: <https://www.simplilearn.com/how-to-control-risk-in-Agile-project-management-article>
21. MORAN, A., *Risk Management in Agile projects*, 2016, p.3

22. CHO, J. J., *An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum*, 2010, p.143-168
23. MUDUMBA, V., LEE, D., *A New perspective on GDSD Risk Management*, 2010, p.4-6
24. COOKE-DAVIES, T.: The “real” success factors on projects. *International Journal of Project Management* 20(3), 185-190 (2002)
25. BENG LEAU, Y., KHONG LOO, W., YIP THAM, W., FUN, S.: Software Development Life Cycle AGILE vs Traditional Approaches. *IPCSIT* vol. 37, 162 – 167 (2012)
26. ALQUADAH, M., RAZALI, R.: A comparison of SCRUM and KANBAN for identifying their selection factors. In *Proceedings of 6th International Conference on Electrical Engineering and Informatics* (2018)
27. PEREIRA, J. C., ROSARIA de F. S. M. Russo.: Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Computer Science* vol. 138: 775–82 (2018)
28. TRZECIAK, M.: Sustainable Risk Management in I.T. Enterprises. *Risks* vol 9 (7) (2021)
29. MORAN, A.: Agile Risk Management. In: *Agile Risk Management*. SpringerBriefs in Computer Science. Springer (2014)
30. *Overview of formulas in Excel* [žiūrēta 2024-04-19]. Prieiga per: <https://support.microsoft.com/en-us/office/overview-of-formulas-in-excel-ecfdc708-9162-49e8-b993-c311f47ca173>

Priedai

1 priedas. Artefaktų sąrašas

Id	Artefaktas	Artefakto aprašymas
AR1	Produkto atliekamų darbų sąrašas (angl. <i>product backlog</i>)	Naujai sudaromas, tvarkingas sąrašas to, ko reikia produktui patobulinti.
AR2	Produkto tikslas (angl. <i>product goal</i>)	Produkto tikslas apibūdina būsimą produkto būseną, kuri gali būti <i>Scrum</i> komandos planavimo tikslas.
AR6	Funkcionalumo modulio aukšto lygio poreikių surinkimas (angl. <i>epic discovery</i>)	Procesas, kurio metu yra vykdomos sesijos su suinteresuotais asmenimis siekiant išsiaiškinti aukšto lygio kliento poreikius tam tikram funkcionalumo moduliui įgyvendinti
AR8	Funkcionalumo modulio įgyvendinimas (angl. <i>epic implementation</i>)	Procesas, kurį sprinto ar kelių sprintų metu atlieka <i>Scrum</i> projekto komanda, kad visi funkcionalumo modulyje esantys naudotojo pasakojimai atitiktų atlikto darbo apibrėžimą
AR10	Techninių užduočių sąrašas (angl. <i>tech debt</i>)	Užduočių sąrašas, skirtas produkto architektūrai bei kodo higienai palaikyti
AR12	Dizaino reikalavimų surinkimas (angl. <i>design discovery</i>)	Procesas, kurio metu yra surenkami detalūs reikalavimai produkto dizainui ir sudaromas naudotojo pasakojimų sąrašas
AR14	Sprinto atliekamų darbų sąrašas (angl. <i>sprint backlog</i>)	Sprinto atliekamų darbų sąrašą sudaro sprinto tikslas, sprintui atrinktų produktų sąrašo elementų rinkinys ir įgyvendinamas planas, kaip pasiekti produkto prieaugį
AR15	Sprinto tikslas (angl. <i>sprint goal</i>)	Programuotojų įsipareigojimas atlikti tam tikrus darbus, kurie priartintų komandą prie projekto tikslo.
AR16	Produkto prieaugis (angl. <i>product increment</i>)	Užduočių, atitinkančių atlikto darbo apibrėžimą, aibė per sprintą
AR23	Produkto vystymo planas (angl. <i>roadmap</i>)	Strateginis planas, kuriame apibrėžiamas tikslas arba pageidaujamas rezultatas ir nurodomi pagrindiniai žingsniai arba etapai, reikalingi jam pasiekti
AR25	Produkto išleidimo planas (angl. <i>release plan</i>)	Informacijos rinkinys, padedantis apibrėžti išleidimo apimtį, svarbius etapus ir terminus.
AR26	Naujų komandos narių apmokymas	Užduotis skirta naujų komandos narių įtraukimui į projektą

2 priedas. Atributų sąrašas

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
A2	Klaidų skaičius	Apskaičiuota reikšmė	=COUNTIF('failas'!“stulpelis“, "Bug")	AR1 AR8 AR14	R7, R8, R13, R17 R7, R8, R13, R17 R16, R17
A4	Standartinio nuokrypio apimtis	Įvedamas	Nurodoma skaitinė reikšmė	AR14	

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
			(duomenų tipas: integer)		
A5	Pajėgumas (angl. <i>velocity</i>)	Apskaičiuota reikšmė	=SUM('failas'!'s tulpelis')	AR8 AR14	
A6	Pajėgumo standartinis nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A5, A4)	AR14	R5, R22, R32, R57, R58, R59
A7	Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius	Apskaičiuota reikšmė	=COUNTIF('failas'!'stulpelis', "ClientApproved") =COUNTIF('failas'!N:N, "Sprint") /COUNT('failas' !C:C)	AR1 A10 AR23 AR25	R1, R39, R47, R49 R1, R39, R47, R49 R1, R39, R47, R49 R1, R39, R47, R49
A8	Užduočių, atitinkančių atliktos užduoties apibrėžimą, skaičius	Apskaičiuota reikšmė	=COUNTIF('failas'!'stulpelis', "Done")	AR16	R3, R4, R12, R34, R35, R36, R40, R42
A9	Užduoties išvestinių užduočių skaičius (angl. <i>subtask</i>)	Apskaičiuota reikšmė	=SUM('failas'!'s tulpelis')	AR8	R17, R42, R43
A10	Apimtis (angl. <i>estimate</i>)	Apskaičiuota reikšmė	=SUM('failas'!'s tulpelis')	AR8 AR10 AR14 AR16 AR25	
A11	Apimtys skirtumas nuo pajėgumo	Apskaičiuota reikšmė	A5-A10	AR8 AR14	R33, R34, R34, R36 R33, R34, R34, R36, R40
A12	Dizaino failas	Įvedamas	Reikšmė parenkama iš enumeratoriaus reikšmių: „Yra“, „Nėra“	AR6 AR8 AR12	R4, R15, R40, R39 R1, R4, R40 R39 R1, R4, R40 R39
A14	Kliento atsakymo laikas	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR6 AR8 AR12	R39 R38, R39 R39

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
A15	Komandos atsakymo laikas	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR6 AR8 AR12	R7, R12, R13 R7, R12, R15 R7, R12, R15
A16	Apmokymo laiko vidurkis asmeniui	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR2 AR26	R25 R24, R25, R9, R10, R11
A17	Išvestinių užduočių skaičiaus vidurkis	Apskaičiuota reikšmė		AR8	R42, R43, R60, R61
A18	Išvestinių užduočių skaičiaus standartinis nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A17, A4)	AR8	R42, R43, R60, R61
A19	Laiko, skirto kodo peržiūrai, vidurkis	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR16	R20, R21, R29, R30, R31, R32 R20, R21, R29, R30, R31, R32
A20	Žmonių, atliekančių kodo peržiūrą, skaičius	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR16	R29, R30, R31, R32 R29, R30, R31, R32
A21	Komentarų skaičius po kodo peržiūros	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR16	R7, R18, R21 R7, R18, R21
A23	Vidutinis vienetų testų kiekis užduočiai	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR14	R20, R22, R24 R20, R22, R24
A23	Vidutinis automatinių testų kiekis užduočiai	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR14	R20, R22, R24 R20, R22, R24
A24	Vidutinis kodo grąžinimo į vykdymo stadiją po peržiūros skaičius	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR14	R20, R22, R24 R20, R22, R24
A26	Užblokuotų užduočių skaičius	Apskaičiuota reikšmė	Iš eksportuojamų duomenų failo surenkamos visos užduotys, kurios turi ‚Blocked by‘ požymį susietą su užduotimi esančia sąrašė	AR8 AR10 AR16	R41, R42, R44, R57, R59 R41, R42

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
A29	Po kiek dienų uždarytas sprintas po numatytos datos	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR15 AR16	R4, R7, R10 R4, R7, R10
A32	Laikas skiriamas demonstraciniam susitikimams	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR16	R40, R37
A33	Laikas skiriamas retrospektyviniams susitikimams	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR16	R56
A34	Procentas laiko, skiriamo susitikimams nuo bendro komandos darbo laiko	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR8 AR14 AR16 AR26	R5, R9, R15, R49, R56 R5, R9, R15, R49, R56 R5, R9, R15, R49, R56 R5, R9, R15, R49, R56
A36	Perdaromų užduočių skaičius (angl. <i>reopened</i>)	Apskaičiuota reikšmė	=COUNTIF('failas'!“stulpelis“, "Reopened")	AR6 AR14	R2, R3, R7, R8, R17, R20, R21, R27 R2, R3, R5, R7, R8, R17, R20, R21, R27
A37	Neplanuotų užduočių kiekis sprints (angl. <i>hotfix</i>)	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR14 AR15 AR16	R37, R39, R40 R37, R39, R40 R37, R39, R40
A38	Dokumentacijos peržiūros periodiškumas	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR2 AR16 AR25 AR26	R25, R26, R28 R24, R25, R26, R27, R28
A39	Priklausomų užduočių skaičius viename sprints	Apskaičiuota reikšmė	=COUNTIF('failas'!“stulpelis“, "Subtask link")	AR14	R16, R35, R41 R41
A40	Neuždarytų versijų skaičius	Apskaičiuota reikšmė	=COUNTA(UNIQUE('failas'!“stulpelis“))	AR23 AR25	R54, R55 R54, R55
A41	Apimties st.nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A52, A4)	AR14 AR16	R64, R57, R59 R64, R57, R59, R58
A42	Užduočių kiekis funkcionalumo vienetė	Apskaičiuota reikšmė	=COUNTIF('06'!A:A,"Story") + COUNTIF('06'!A:A,"Spike") +	AR1 AR8 AR10	R12, R28, R39, R47 R60

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
			COUNTIF('06! A:A,"Task") + COUNTIF('06! A:A,"Design task")	AR14	R60
A43	Užduočių, atitinkančių atliktos užduoties apibrėžimą st.nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A8, A4)	AR16	R59, R64
A44	Paruoštų sprintų skaičius	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR1 AR2	R12, R28, R39, R47 R12, R28, R39, R47
A45	Procentas kritinių užduočių	Apskaičiuota reikšmė	=(COUNTIF('failas!' "stulpelis", "Critical") + COUNTIF('failas!' "stulpelis", "Blocker"))/ COUNT('failas!' "stulpelis")	AR1 AR10 AR14	R46, R47, R43
A46	Procentas mažo prioriteto užduočių	Apskaičiuota reikšmė	=(COUNTIF('failas!' "stulpelis", "Minor") + COUNTIF('failas!' "stulpelis", "Trivial"))/ COUNT('failas!' "stulpelis")	AR1 AR10 AR14	R46, R47
A47	Procentas atliktų užduočių	Apskaičiuota reikšmė	=A8/A42	AR16 AR25	R64, R57, R59 R64, R57, R59, R58
A48	Užduočių kiekio standartinis nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A42, A4)	A14 AR25	R64, R57, R59 R64, R57, R59, R58
A49	Procentas apimties skiriamas techninių užduočių atlikimui	Apskaičiuota reikšmė	=(COUNTIF('failas!' "stulpelis", "TechDebt"))	AR16 AR25	R20, R21, R22, R23 R20, R21, R22, R23
A50	Užduočių kiekis įkeliamų duomenų faile su nenurodytu funkcionalumo moduli	Apskaičiuota reikšmė	=COUNTBLANK('failas!' "stulpelis")	AR1 AR14	R50 R50

id	Atributas („Atributas“ klasė – „Pavadinimas“)	Atributo tipas („Atributas“ klasė – „Tipas“)	Išraiška	Yra rekomenduojami artefaktams	Galimos rizikos
A51	Funktionalumo modulių kiekis	Apskaičiuota reikšmė	=COUNTA(UNIQUE ('failas'!'stulpelis'))	AR1 AR2 AR23	R60, R61 R60, R61 R60, R61
A52	Retrospektyvi apimtis	Įvedamas	Nurodoma skaitinė reikšmė (duomenų tipas: integer)	AR16 AR25	
A53	Apimties ir retrospektyvios apimties skirtumas	Apskaičiuota reikšmė	A53-A10	AR16 AR25	R5, R7, R8, R34, R35, R36 R5, R7, R8, R34, R35, R36
A54	Retrospektyvios apimties st.nuokrypis	Apskaičiuota reikšmė	=ST.DEV(A52, A4)	AR16 AR25	R5, R7, R8, R36 R5, R7, R8, R36

3 priedas. Rizikų kategorijos

Nr.	Rizikos kategorija	Aprašymas
K1	Reikalavimų tenkinimo rizikos	Rizikos, susijusios su suinteresuotų asmenų keliamų reikalavimų tenkinimu kuriamam ar tobulinamam produktui
K2	Komandos narių kompetencijų rizikos	Rizikos, susijusios su projekto komandos turimomis kompetencijomis, resursų paskirstymas ne pagal žinomą sritį, gabumus, kompetencijų nuvertinimu ar pervertinimu
K3	Bendradarbiavimo rizikos	Rizikos, susijusios su projekto komandos tarpusavio bendradarbiavimu bei pasikeitimų eskalavimu visiems suinteresuotiems komandos nariams – skaidri komunikacija tarp produkto vadovų, programuotojų, testuotojų.
K4	Apmokymo rizikos	Rizikos, susijusios su naujų komandos narių apmokymu – nepakankama arba perteklinė informacija, laiko sąnaudos, tikslingos informacijos nebuvimas
K5	Atsakomybių pasidalinimo rizikos	Rizikos, susijusios su daugialype atsakomybe vienam komandos nariui mažesnėse komandose arba atsakomybės stoka didelėse komandose
K8	Su kūrimo procesu susijusios rizikos	Rizikos, susijusios su projektų valdymo pagal <i>Scrum</i> metodiką procesu – komandos žinių trūkumas apie metodiką, metodikos nepriėmimas ar neintegravimas, proceso nesilaikymas.
K9	Kokybės užtikrinimo rizikos	Rizikos, susijusios su produkto testavimu ir kokybės atitikimo validavimu – testavimui skiriamas laikas, testavimo procedūrų nebuvimas, su testų automatizavimu susijusios rizikos.
K10	Kodo peržiūros rizikos	Rizikos, susijusios su kodo peržiūros procesu – kodo peržiūros nebuvimas, nepakankamas komandos narių įsitraukimas.
K11	Formalaus kodo standarto tenkinimo rizikos	Rizikos, susijusios su vieningo formalaus standarto nebuvimu – nesuprantamas ir sunkiai palaikomas kodas
K12	Su dokumentacija susijusios rizikos	Rizikos, susijusios su dokumentacija – dokumentacijos nebuvimas, neatnaujinimas, resursų laiko sąnaudos perteklinei dokumentacijai kaupti
K13	Apimties vertinimo rizikos	Rizikos, susijusios su planavimo pokeriu – vieningos etaloninės pokerio skalės nebuvimas, nukrypimai nuo etalono, užduočių neįvertinimai ar pervertinimai

Nr.	Rizikos kategorija	Aprašymas ⁵
K14	Kliento įsitraukimo rizikos	Rizikos, susijusios su menku kliento įsitraukimu – laiko sąnaudos informacijai surinkti, neteisingai apibrėžti reikalavimai, darbo pajėgumo sulėtėjimas. Šios kategorijos rizikų įvykimas gali būti rizikų kategorijos K1. rizikų pasekmė
K15	Modulių priklausomybių rizikos	Rizikos, susijusios su skirtingų funkcionalumo modulių tarpusavio priklausomybėmis – užduočių blokavimai bei įtaka pakeitimams tarp skirtingų modulių
K16	Prioritetų rizikos	Rizikos, susijusios su neaiškia prioritetų vizija – neteisingas resursų paskirstymas, neišlaikomi terminai
K19	Žinių dalijimosi rizikos	Rizikos, susijusios su didesnių klaidų skaičiumi projekte, kai nėra efektyviai dalinamasi žiniomis tarp labiau ir mažiau patyrusių komandos narių. Šios kategorijos rizikų įvykimas gali būti rizikų kategorijos K12. rizikų pasekmė
K21	Atsekamumo rizikos	Rizikos, susijusios su produkto versijos sekimu bei produkto prieaugio nustatymu skirtingais veiklos etapais
K22	Su nuolatiniu tobulėjimu susijusios rizikos	Rizikos, susijusios su retrospektyviniais procesais – retro susitikimais, veiksmų plano sudarymu, nuolatiniu tobulėjimu.
K23	Su darbų planavimu susijusios rizikos	Rizikos, susijusios su produkto atliekamų darbų sąrašo skaidymu į sprintus ir darbų apimties prognozavimu komandai

4 priedas. Rizikos pagal rizikos kategoriją

Id.	Rizika	Rizikos kategorija
R1	Neapibrėžti reikalavimai	K1 Reikalavimų tenkinimo rizikos
R2	Nevienareikšmiški reikalavimai	K1 Reikalavimų tenkinimo rizikos
R3	Neteisingai apibrėžti reikalavimai	K1 Reikalavimų tenkinimo rizikos
R4	Neišpildyti reikalavimai	K1 Reikalavimų tenkinimo rizikos
R5	Neatsižvelgiama į komandos narių turimas žinias - neteisingas pareigų paskirstymas	K2 Komandos narių kompetencijų rizikos
R6	Komandos nariai nepatiria iššūkių	K2 Komandos narių kompetencijų rizikos
R7	Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų	K2 Komandos narių kompetencijų rizikos
R8	Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius	K3 Bendradarbiavimo rizikos
R9	Apmokymo rizika - sugaištama daug laiko	K4 Apmokymo rizikos
R10	Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus	K4 Apmokymo rizikos
R11	Apmokymo rizika - nesuteikta visa reikiama informacija naujiems darbuotojams	K4 Apmokymo rizikos
R12	Atsakomybės stoka didelėse komandose - neprisiimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus	K4 Apmokymo rizikos
R13	Neatidus testuotojų darbas	K9 Kokybės užtikrinimo rizikos
R15	Nesklandi <i>Scrum</i> procesų integracija komandai	K8 Su kūrimo procesu susijusios rizikos
R16	Testuotojų darbo atsilikimas nuo programuotojų sprinto metu	K9 Kokybės užtikrinimo rizikos

Id.	Rizika	Rizikos kategorija
R17	Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams	K9 Kokybės užtikrinimo rizikos
R18	Vienetų testavimai - neatidus programuotojų testavimas	K9 Kokybės užtikrinimo rizikos
R19	Vienetų testavimai - sunkus pritaikymas paveldėtam kodui	K9 Kokybės užtikrinimo rizikos
R20	Formalaus kodo standarto nebuvimas - sunku palaikyti kodą	K11 Formalaus kodo standarto tenkinimo rizikos
R21	Formalaus kodo standarto nebuvimas - sunku suprasti kitų programuotojų kodą	K11 Formalaus kodo standarto tenkinimo rizikos
R22	Formalus kodo standartas - sumažintas našumas	K11 Formalaus kodo standarto tenkinimo rizikos
R23	Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas	K11 Formalaus kodo standarto tenkinimo rizikos
R24	Dokumentacijos sumažėjimas - trūksta scenarijų kokybės užtikrinimo specialistams	K12 Su dokumentacija susijusios rizikos
R25	Dokumentacijos sumažėjimas - nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas	K12 Su dokumentacija susijusios rizikos
R26	Dokumentacijos versijos perrašymas tarp komandos narių	K12 Su dokumentacija susijusios rizikos
R27	Dokumentacijos neatnaujinimas – paliekama informacija, kuri nebėra tiksli ir galiojanti esamam funkcionalumui	K12 Su dokumentacija susijusios rizikos
R28	Dokumentacijos sumažėjimas - priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą	K12 Su dokumentacija susijusios rizikos
R29	Kodo peržiūros nebuvimas - nekokybiškas kodas	K10 Kodo peržiūros rizikos
R30	Kodo peržiūros nebuvimas - sumažinta galimybė tobulėti	K10 Kodo peržiūros rizikos
R31	Kodo peržiūros nebuvimas - nėra grįžtamojo ryšio	K10 Kodo peržiūros rizikos
R32	Kodo peržiūros nebuvimas - laiko sąnaudos siekiant kurti tai, kas jau sukurta	K10 Kodo peržiūros rizikos
R33	Apimties vertinimas - užduotis pervertinta	K13 Apimties vertinimo rizikos
R34	Apimties vertinimas - užduotis nuvertinta	K13 Apimties vertinimo rizikos
R35	Apimties vertinimas - užduotis vertinama pilnai nesuprantant reikalavimų	K13 Apimties vertinimo rizikos
R36	Apimties vertinimas - užduotis vertinama nežinant techninių apribojimų	K13 Apimties vertinimo rizikos
R37	Menkas kliento įsitraukimas - reikalavimų neatitikimas	K14 Kliento įsitraukimo rizikos
R38	Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas	K14 Kliento įsitraukimo rizikos
R39	Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga	K14 Kliento įsitraukimo rizikos

Id.	Rizika	Rizikos kategorija
R40	Menkas kliento įsitraukimas - atliktų užduočių perdarymas	K14 Kliento įsitraukimo rizikos
R41	Tarpusavio priklausomybė tarp užduočių – darbų vėlavimai	K15 Modulių priklausomybių rizikos
R42	Tarpusavio priklausomybė tarp užduočių – apimties išaugimas	K15 Modulių priklausomybių rizikos
R43	Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai	K15 Modulių priklausomybių rizikos
R44	Bendravimo stoka tarp komandų dirbančių prie vieno modulio - pakeitimų perrašymai	K18 Su komunikacija susijusios rizikos
R46	Per daug lygiagrečių prioritetų - neužtenka resursų	K16 Prioritetų rizikos
R47	Nėra prioritetų – neaiški vizija, kliento nusivylimas	K16 Prioritetų rizikos
R49	Neefektyvus bendravimas nuotoliniu būdu	K18 Su komunikacija susijusios rizikos
R50	Struktūrizuotos informacijos nebuvimas	K19 Žinių dalijimosi rizikos
R54	Versijos neatnaujinimas	K21 Atsekamumo rizikos
R55	Versijos užduočių ne atsekamumas	K21 Atsekamumo rizikos
R56	Neproduktyvūs retrospektyviniai susitikimai	K22 Su nuolatiniu tobulėjimu susijusios rizikos
R57	Sprinto atliekamų darbų skaičiaus mažėjimas	K23 Su darbų planavimu susijusios rizikos
R58	Sprinto atliekamų darbų skaičiaus didėjimas nedidinant komandos - kokybės kritimas	K23 Su darbų planavimu susijusios rizikos
R59	Sprinto atliekamų darbų skaičiaus prognozės neatitikimas	K23 Su darbų planavimu susijusios rizikos
R60	Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis	K23 Su darbų planavimu susijusios rizikos
R61	Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių	K23 Su darbų planavimu susijusios rizikos
R62	Neišbaigtas sprinto planas – neužtektinai darbo komandai	K23 Su darbų planavimu susijusios rizikos
R63	Perteklinis sprinto planas – komandos moralės kritimas	K23 Su darbų planavimu susijusios rizikos
R64	Sprinto pajėgumo nenusipėjamumas – neadekvatus prognozavimas	K23 Su darbų planavimu susijusios rizikos

5 priedas. Naudotojo sąsajos modelis

1. Kategorijų sąrašas

Risk management system Projects Artifacts Attributes Risk Classifier

Risk Classifiers + Add Risk Classifier

Requirements	Competencies	Communication	Onboarding
Risks associated with meeting stakeholder requirements for the product being developed or improved	Risks related to the competences of the project team, allocation of resources outside the known field, talents, underes...	Risks related to collaboration among the project team and escalation of changes to all interested team members - transpare...	Risks associated with training new team members - insufficient or redundant information, time costs, lack of targeted L...
R1 Undefined requirements	R5 Misallocation of responsibilities	R8 Information on changes is not communicated	R9 Time consumption on onboarding
R2 Ambiguous requirements	R6 Lack of challenge for team		R10 Lack of knowledge of Scrum processes
R3 Incorrect requirements	R7 Lack of competent specialists		R11 Not all required information given to new employees
R4 Unfulfilled requirements			R12 Lack of responsibility for failure to meet deadlines and delays in sprints
+ Add Risk	+ Add Risk	+ Add Risk	+ Add Risk

2. Artefaktų sąrašas

Risk Management System Projects Global Artifacts Attributes Risk Classifiers

Artefacts + Add Artefact

Artefact ID	Artefact Name	Description
AR1	Product Backlog	A prioritized list of work for the development team that is derived from the roadmap and its requirements.
AR2	Product Goal	A measurable and actionable target describing what the Scrum Team aims to build in the months ahead and for wha...
AR6	Epic Discovery	A process whereby sessions are conducted with stakeholders to clarify the high-level needs of the customer for the L...
AR8	Epic Implementation	The process, carried out by a Scrum project team during a sprint or multiple sprints, to ensure that all user stories in a...
AR10	Technical Debt	A list of tasks to maintain product architecture and code hygiene
AR12	Design Discovery	The process of gathering detailed requirements for a product design and compiling a list of user stories
AR14	Sprint Backlog	The sprint to-do list consists of the sprint goal, a set of items from the list of products selected for the sprint, and an L...
AR15	Sprint Goal	A commitment from the developers to perform certain tasks that will bring the team closer to the project goal.
AR16	Product Increment	A set of tasks that meet the definition of done in a sprint
AR23	Roadmap	A strategic plan that defines the objective or desired outcome and identifies the key steps or milestones needed to ac...
AR25	Release Plan	A set of information to help define the scope, milestones and timing of a release.
AR26	Onboarding	Task to involve new team members in the project

3. Atributų sąrašas

Attributes

[+ Add Attribute](#)

Attribute ID	Attribute Name	Rule	Expression
A2	Bug Count	Calculated	COUNT (Type='Bug')
A3	Time for Bug Fixes	Calculated	SUM (Story Point) WHERE (Type='Bug')
A4	Standard Deviation Interval	Integer	-
A5	Velocity	Integer	-
A6	St. Deviation of Velocity	Calculated	STDEV.S(A5, A4)
A7	Count of DoR Tasks	Calculated	COUNT (Label='ClientApproved')
A8	Count of DoD Tasks	Calculated	COUNT (Resolution='Done')
A9	Subtask Count	Calculated	COUNT
A10	Estimate	Calculated	SUM (Story Point)
A11	Estimate and Velocity Delta	Calculated	'A5-A11'
A12	Design File	Picklist	'Yes'/'No'
A14	Client Response Time	Integer	-
A15	Team Response Time	Integer	-

4. Projektų sąrašas

Projects

[+ Add Project](#)

Project "DVS" 

Development of a platform that is used to assess the core strengths and areas for improvement of employees and candidates, and to select the most suitable work area for a particular person

 Jira

Start date: 22/01/2024

Project "EBS" 

Development of a platform that is used to assess e-banking site to complete transactions, open accounts, make international wires.

 Jira

Start date: 22/01/2024

Project "HRS" 

Development of a platform that is used to manage personnel and calculate salaries for employees of the company.

 Jira

Start date: 22/01/2024

5. Artefaktų projekte sąrašas

x

Projects > Project "DVS"

←  Projects

Project "DVS"

Artefacts

Files



Development of a platform that is used to assess the core strengths and areas for improvement of employees and candidates, and to select the most suitable work area for a particular person

Created: 22/01/2024

Expires: 23/12/2025

Artefacts

Search artefacts




[+ Add Artefacts](#)  

<input type="checkbox"/>	Artefact Name	Description	Status	Attributes
<input type="checkbox"/>	Product Backlog	A prioritized list of work for the development team that is derived fro...	Not detected	8
<input type="checkbox"/>	Product Goal	A measurable and actionable target describing what the Scrum Team...	Not detected	5
<input type="checkbox"/>	Epic Implementation	The process, carried out by a Scrum project team during a sprint or m...	Not detected	14
<input type="checkbox"/>	Tech Debt	A list of tasks to maintain product architecture and code hygiene	Not detected	7
<input type="checkbox"/>	Sprint Backlog	The sprint to-do list consists of the sprint goal, a set of items from the l...	Not detected	13
<input type="checkbox"/>	Product Increment	A set of tasks that meet the definition of done in a sprint	Not detected	9
<input type="checkbox"/>	Release Plan	A set of information to help define the scope, milestones and timing o...	Not detected	5
<input type="checkbox"/>	Onboarding	Task to involve new team members in the project	Not detected	5


[Active Risks](#) [Closed Risks](#)



6. Artefakto atributų sąrašas


Risk Management System Projects Artefacts Attributes Risk Classifiers   

Projects > Project "DVS" > **Sprint Backlog**

 Jira Development of a platform that is used to assess the core strengths and areas for improvement of employees and candidates... **Created:** 22/01/2024 **Expires:** 23/12/2025

Attributes

Search attributes

[+ Add Attributes](#) 

Attribute Name	Value	Date Modified
Interval of standard deviation	-	22/01/2024
Velocity	-	22/01/2024
Standard deviation of velocity	-	22/01/2024
DoR Tasks (%)	-	22/01/2024
Estimation	-	22/01/2024
Velocity and Estimation delta	-	22/01/2024
Low priority items (%)	-	22/01/2024
Critical priority items (%)	-	22/01/2024
Number of tasks	-	22/01/2024
Standard deviation of tasks n.	-	22/01/2024
Time for tech debt (%)	-	22/01/2024
Number of hotfixes	-	22/01/2024
Meeting time	-	22/01/2024

Risks

Attribute: Standard deviation of velocity
Value: -
Date Modified: 22/01/2024

[+ Add Risks](#)

Risk name	Trigger Value	Status
R5	20	Not detected
R22	20	Not detected
R32	20	Not detected
R57	20	Not detected
R58	20	Not detected
R59	20	Not detected

7. Rizikos registravimo langas

Log Risk - R22 ×

Formal Code Standard - Lack of Efficiency

Description

57/300

Effect

Priority

Responsible person

Key actions

57/300

[Cancel](#) [Log Risk](#)

8. Užregistruotų rizikų sąrašas

R22 - Formal Code Standard - Lack of Efficiency

Priority 2

Assigned 10/11/2023

Logged 06/11/2023

9. Atributo reikšmių kitimo langas

Risk Management System
Projects Artefacts Attributes Risk Classifiers

Projects > Project "DVS" > Sprint Backlog

Jira
Development of a platform that is used to assess the core strengths and areas for improvement of employees and candidates...
Created: 22/01/2024 Expires: 23/12/2025

Attributes Search attributes

[+ Add Attributes](#)

Attribute Name	Value	Date Modified
Interval of standard deviation	3	22/01/2024
Velocity	52	22/01/2024
Standard deviation of velocity	3,79	22/01/2024
DoR Tasks (%)	100	22/01/2024
Estimation	66	22/01/2024
● Velocity and Estimation delta	14	22/01/2024
Low priority items (%)	5	22/01/2024
Critical priority items (%)	14	22/01/2024
Number of tasks	43	22/01/2024
Standard deviation of tasks n.	3,79	22/01/2024
Time for tech debt (%)	15	22/01/2024
Number of hotfixes	0	22/01/2024
Meeting time	20	22/01/2024

Value History ×




Attribute Velocity

Value 52

Date Modified 22/01/2024

[Edit](#)

10. Rizikos reikšmių kitimo langas


Risk Management System Projects Artefacts Attributes Risk Classifiers   

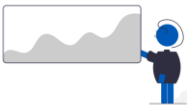
Projects > Project "DVS" > R22 - Formal Code Standard - Lack of Efficiency

R22 - Formal Code Standard - Lack of Efficiency Close Risk

Effect	Priority	Status	Action Items
Major	2	Active	<ol style="list-style-type: none"> 1. Organise tech sync 2. Review best practices 3. Update documentation in SharePoint

Date modified	Assigned	Modified by
01/12/2023	Adele Necionyte	Adele Necionyte

Risk History Search risks 



No risk history available

6 priedas. Projekto „DVS“ artefaktų ir atributų su priskirtomis rizikomis sąrašas

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
AR1.	Produkto atliekamų darbų sąrašas (angl. <i>product backlog</i>)	A2 – Klaidų skaičius	R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R8 - Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius R13 – Neatidus testuotojų darbas R17 - Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams
		A7 – Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius	R1 - Neapibrėžti reikalavimai R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas R49 - Neefektyvus bendravimas nuotoliniu būdu
		A42 – Užduočių kiekis funkcionalumo vienetė	R28 - Dokumentacijos sumažėjimas - priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas
		A45 - Procentas kritinių užduočių	R46 - Per daug lygiagrečių prioritetų - neužtenka resursų R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas
		A46 - Procentas mažo prioriteto užduočių	R46 - Per daug lygiagrečių prioritetų - neužtenka resursų

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
			R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas
		A50 - Užduočių kiekis Įkeliamų duomenų faile su nenurodytu funkcionalumo modulių	R50 - Struktūrizuotos informacijos nebuvimas
		A51 -Funkcionalumo modulių kiekis	R47 Nėra prioritetų – neaiški vizija, kliento nusivylimas R43 Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai R46 Per daug lygiagrečių prioritetų - neužtenka resursų
		A44 - Paruoštų sprintų skaičius	R1 Neapibrėžti reikalavimai R2 Nevienareikšmiški reikalavimai R37 - Menkas kliento įsitraukimas - reikalavimų neatitikimas R38 - Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga
AR2	Produkto tikslas	A50 - Užduočių kiekis Įkeliamų duomenų faile su nenurodytu funkcionalumo modulių	R50 - Struktūrizuotos informacijos nebuvimas
		A51 -Funkcionalumo modulių kiekis	R47 Nėra prioritetų – neaiški vizija, kliento nusivylimas R43 Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai R46 Per daug lygiagrečių prioritetų - neužtenka resursų
		A44 - Paruoštų sprintų skaičius	R1 Neapibrėžti reikalavimai R2 Nevienareikšmiški reikalavimai R37 - Menkas kliento įsitraukimas - reikalavimų neatitikimas R38 - Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga
		A38 - Dokumentacijos peržiūros periodiškumas	R24 - Dokumentacijos sumažėjimas - trūksta scenarijų kokybės užtikrinimo specialistams R25 - Dokumentacijos sumažėjimas - nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas R26 - Dokumentacijos versijos perrašymas tarp komandos narių R27 - Dokumentacijos neatnaujinimas – paliekama informacija, kuri nebėra tiksli ir galiojanti esamam funkcionalumui

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
			R28 - Dokumentacijos sumažėjimas - priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą
		A16 – Apmokymo laikas asmeniui	R9 Apmokymo rizika - sugaištama daug laiko R10 Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R11 Apmokymo rizika - nesuteikta visa reikiama informacija naujiems darbuotojams
AR8.	Funkcionalumo modulio įgyvendinimas (angl. <i>epic implementation</i>)	A2 – Klaidų skaičius	R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R8 - Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius R17 - Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams
		A3 - Laikas, skiriamas klaidų taisymui (procentas nuo bendro laiko)	R3 – Neteisingai apibrėžti reikalavimai R7 – Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R8 - Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius R29 - Kodo peržiūros nebuvimas - nekokybiškas kodas R34 - Apimties vertinimas - užduotis nuvertinta
		A5 - Pajėgumas	Nesekama, naudojama skaičiavimuose
		A9 - Užduoties išvestinių užduočių skaičius	R17 - Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams R42 - Tarpusavio priklausomybė tarp užduočių – apimties išaugimas R43 - Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis R61 – Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių
		A17 - Išvestinių užduočių skaičiaus vidurkis	R17 - Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams R42 - Tarpusavio priklausomybė tarp užduočių – apimties išaugimas R43 - Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis R61 – Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
		A18 - Išvestinių užduočių skaičiaus standartinis nuokrypis	<p>R17 - Bendradarbiavimas tarp QA ir programuotojų - pakeitimai, apimantis daugiau sistemos vietų neperduodami testuotojams</p> <p>R42 - Tarpusavio priklausomybė tarp užduočių – apimties išaugimas</p> <p>R43 - Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai</p> <p>R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis</p> <p>R61 – Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių</p>
		A10 - Apimtis	Nesekama, naudojama skaičiavimuose
		A26 - Užblokuotų užduočių skaičius	R61 – Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių
		A18 - Laiko, skirtas užduoties įgyvendinimui skirtumas nuo apimties vertinimo	<p>R1 - Neapibrėžti reikalavimai</p> <p>R2 - Nevienareikšmiški reikalavimai</p> <p>R3 - Neteisingai apibrėžti reikalavimai</p> <p>R33 - Apimties vertinimas - užduotis pervertinta</p> <p>R34 - Apimties vertinimas - užduotis nuvertinta</p> <p>R35 - Apimties vertinimas - užduotis vertinama pilnai nesuprantant reikalavimų</p> <p>R36 - Apimties vertinimas - užduotis vertinama nežinant techninių apribojimų</p> <p>R44 - Bendravimo stoka tarp komandų dirbančių prie vieno modulio - pakeitimų perrašymai</p>
		A12 - Dizaino failas	<p>R4 - Neišpildyti reikalavimai</p> <p>R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai</p> <p>R40 - Menkas kliento įsitraukimas - atliktų užduočių perdarymas</p> <p>R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga</p>
		A14 - Kliento atsakymo laikas	<p>R38 - Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas</p> <p>R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga</p>
		A15 - Komandos atsakymo laikas	<p>R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų</p> <p>R12 - Atsakomybės stoka didelėse komandose - neprisiimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus</p> <p>R13 - Pasitikėjimo stoka tarp komandos narių</p>
		A42 - Užduočių kiekis funkcionalumo vienete	R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
		A20 - Žmonių, atliekančių kodo peržiūrą, skaičius	R29 - Kodo peržiūros nebuvimas - nekokybiškas kodas R30 - Kodo peržiūros nebuvimas - sumažinta galimybė tobulėti R31 - Kodo peržiūros nebuvimas - nėra grįžtamojo ryšio R32 - Kodo peržiūros nebuvimas - laiko sąnaudos siekiant kurti tai, kas jau sukurta
		A24 – Vidutinis kodo gražinimo į vykdymo stadiją po peržiūros skaičius	R2 - Nevienareikšmiški reikalavimai R4 - Neišpildyti reikalavimai R5 - Neatsižvelgiama į komandos narių turimas žinias - neteisingas pareigų paskirstymas R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R15 - Nesklandi <i>Scrum</i> procesų integracija komandai R16 - Testuotojų darbo atsilikimas nuo programuotojų sprinto metu
		A20 - Žmonių, atliekančių kodo peržiūrą, skaičius	R29 - Kodo peržiūros nebuvimas - nekokybiškas kodas R30 - Kodo peržiūros nebuvimas - sumažinta galimybė tobulėti R31 - Kodo peržiūros nebuvimas - nėra grįžtamojo ryšio R32 - Kodo peržiūros nebuvimas - laiko sąnaudos siekiant kurti tai, kas jau sukurta
AR10	Techninių užduočių atliekamų darbų sąrašas (angl. <i>techdebt</i>)	A7 – Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius	R1 - Neapibrėžti reikalavimai R39 - Menkas kliento išitraukimas - neaiški vizija bei projekto eiga R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas R49 - Neefektyvus bendravimas nuotoliniu būdu
		A10 - Apimtis	Nesekama, naudojama skaičiavimuose
		A45 - Procentas kritinių užduočių	R46 - Per daug lygiagrečių prioritetų - neužtenka resursų R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas
		A46 - Procentas mažo prioriteto užduočių	R46 - Per daug lygiagrečių prioritetų - neužtenka resursų R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas
		A42 - Užduočių kiekis funkcionalumo vienetė	R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis R46 - Per daug lygiagrečių prioritetų - neužtenka resursų R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas R23 - Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
		A48 - Užduočių kiekio standartinis nuokrypis	R23 - Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas
		A26 - Užblokuotų užduočių skaičius	R61 – Funkcionalumo vienetas per stipriai išskaidomas – sunku palaikyti daug išskaidytų užduočių
AR12	Dizaino reikalavimų surinkimas	A30 – Laikas, skiriamas susitikimams	R4 - Neišpildyti reikalavimai R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R8 - Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius
		A12 - Dizaino failas	R4 - Neišpildyti reikalavimai R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R40 - Menkas kliento įsitraukimas - atliktų užduočių perdarymas R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga
		A14 - Kliento atsakymo laikas	R38 - Menkas kliento įsitraukimas - laiko sąnaudos – ilgas atsakymo laukimas R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga
		A15 - Komandos atsakymo laikas	R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R12 - Atsakomybės stoka didelėse komandose - neprisiimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus R13 - Pasitikėjimo stoka tarp komandos narių
AR14.	Sprinto atliekamų darbų sąrašas (angl. <i>sprint backlog</i>)	A4 – Standartinio nuokrypio apimtis	Nesekama, naudojama skaičiavimuose Parinkta atributo reikšmė – 3
		A5 - Pajėgumas	Nesekama, naudojama skaičiavimuose
		A6 – Pajėgumo standartinis nuokrypis	R5 - neteisingas pareigų paskirstymas R22 - formalus kodo standartas - sumažintas našumas R32 - kodo peržiūros nebuvimas - laiko sąnaudos siekiant kurti tai, kas jau sukurta R57 - Sprinto atliekamų darbų skaičiaus mažėjimas R58 - Sprinto atliekamų darbų skaičiaus didėjimas nedidinant komandos - kokybes kritimas R59 - Sprinto atliekamų darbų skaičiaus prognozės neatitikimas
		A7 – Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius	R1 - Neapibrėžti reikalavimai R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas R49 - Neefektyvus bendravimas nuotoliniu būdu
		A10 - Apimtis	R57 - Sprinto atliekamų darbų skaičiaus mažėjimas

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
			<p>R58 - Sprinto atliekamų darbų skaičiaus didėjimas nedidinant komandos - kokybės kritimas</p> <p>R62 Neišbaigtas sprinto planas – neužtektinai darbo komandai</p> <p>R63 Perteklinis sprinto planas – komandos moralės kritimas</p> <p>R64 Sprinto pajėgumo nenusipėjimas – neadekvatus prognozavimas</p>
		A11 - Apimties skirtumas nuo pajėgumo	<p>R62 Neišbaigtas sprinto planas – neužtektinai darbo komandai</p> <p>R63 Perteklinis sprinto planas – komandos moralės kritimas</p> <p>R64 Sprinto pajėgumo nenusipėjimas – neadekvatus prognozavimas</p>
		A30 – Laikas, skiriamas susitikimams	<p>R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai</p> <p>R8 - Bendradarbiavimo problemos tarp komandos narių - neperduodama informacija apie pokyčius</p> <p>R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus</p>
		A45 - Procentas kritinių užduočių	<p>R46 - Per daug lygiagrečių prioritetų - neužtenka resursų</p> <p>R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas</p>
		A46 - Procentas mažo prioriteto užduočių	<p>R46 - Per daug lygiagrečių prioritetų - neužtenka resursų</p> <p>R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas</p>
		A42 - Užduočių kiekis funkcionalumo vienetė	<p>R60 – Funkcionalumo vienetas nėra išskaidomas į atitinkamos apimties užduotis</p> <p>R46 - Per daug lygiagrečių prioritetų - neužtenka resursų</p> <p>R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas</p>
		A48 - Užduočių kiekio standartinis nuokrypis	R23 - Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas
		A49 - Apimtis techninių užduočių įgyvendinimui %	<p>R20 - Formalaus kodo standarto nebuvimas - sunku palaikyti kodą</p> <p>R21 - Formalaus kodo standarto nebuvimas - sunku suprasti kitų programuotojų kodą</p> <p>R22 - Formalus kodo standartas - sumažintas našumas</p> <p>R23 - Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas</p>
		A37 - Neplanuotų užduočių kiekis sprints (angl. hotfix)	<p>R37 - Menkas kliento įsitraukimas - reikalavimų neatitikimas</p> <p>R40 - Menkas kliento įsitraukimas - atliktų užduočių perdarymas</p>
AR16.	Produkto prieaugis (angl. <i>product increment</i>)	A8 – Užduočių, atitinkančių atliktos užduoties apibrėžimą, skaičius	<p>R4 - Neišpildyti reikalavimai</p> <p>R7 - Neatsižvelgiama į komandos narių turimą žinią - komandoje nėra tinkamų specialistų</p>

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
		A29 - Po kiek dienų uždarytas sprintas po numatytos datos	R4 - Neišpildyti reikalavimai R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R12 - Atsakomybės stoka didelėse komandose - neprisiimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus
		A38 - Dokumentacijos peržiūros periodiškumas	R24 - Dokumentacijos sumažėjimas - trūksta scenarijų kokybės užtikrinimo specialistams R25 - Dokumentacijos sumažėjimas - nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas R26 - Dokumentacijos versijos perrašymas tarp komandos narių R27 - Dokumentacijos neatnaujinimas – paliekama informacija, kuri nebėra tiksli ir galiojanti esamam funkcionalumui R28 - Dokumentacijos sumažėjimas - priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą
		A42 – Užduočių kiekis atliekamų darbų sąrašė	Nesekama, naudojama skaičiavimuose
		A47 - Procentas atliktų užduočių	R4 - Neišpildyti reikalavimai R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R12 - Atsakomybės stoka didelėse komandose - neprisiimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus
		A18 - Laiko, skirtas užduoties įgyvendinimui skirtumas nuo apimties vertinimo	R1 - Neapibrėžti reikalavimai R2 - Nevienareikšmiški reikalavimai R3 - Neteisingai apibrėžti reikalavimai R33 - Apimties vertinimas - užduotis pervertinta R34 - Apimties vertinimas - užduotis nuvertinta R35 - Apimties vertinimas - užduotis vertinama pilnai nesuprantant reikalavimų R36 - Apimties vertinimas - užduotis vertinama nežinant techninių apribojimų R44 - Bendravimo stoka tarp komandų dirbančių prie vieno modulio - pakeitimų perrašymai
		A32 - Laikas skiriamas demonstraciniams susitikimams	R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R37 - Menkas kliento įsitraukimas - reikalavimų neatitikimas

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
			R40 - Menkas kliento įsitraukimas - atliktų užduočių perdarymas
		A33 - Laikas skiriamas retrospektyviniams susitikimams	R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R56 - Neproduktyvūs retrospektyviniai susitikimai
		A49 - Apimtis techninių užduočių įgyvendinimui %	R20 - Formalaus kodo standarto nebuvimas - sunku palaikyti kodą R21 - Formalaus kodo standarto nebuvimas - sunku suprasti kitų programuotojų kodą R22 - Formalus kodo standartas - sumažintas našumas R23 - Formalus kodo standartas – nepalaikomas techninis atliekamų darbų sąrašas
AR25.	Produkto išleidimo planas (angl. <i>release plan</i>)	A7 - Užduočių, atitinkančių paruoštos užduoties apibrėžimą, skaičius	R1 - Neapibrėžti reikalavimai R39 - Menkas kliento įsitraukimas - neaiški vizija bei projekto eiga R47 - Nėra prioritetų – neaiški vizija, kliento nusivylimas R49 - Neefektyvus bendravimas nuotoliniu būdu
		A25 - Užduočių atsekamumas – priklausomų užduočių skaičius	R41 - Tarpusavio priklausomybė tarp užduočių – darbų vėlavimai R42 - Tarpusavio priklausomybė tarp užduočių – apimties išaugimas R43 - Tarpusavio priklausomybė tarp užduočių - viena užduotis atliekama neapgalvojant apie įtaką likusiai sistemai R44 - Bendravimo stoka tarp komandų dirbančių prie vieno modulio - pakeitimų perrašymai (angl. <i>override</i>)
		A38 - Dokumentacijos peržiūros periodiškumas	R24 - Dokumentacijos sumažėjimas - trūksta scenarijų kokybės užtikrinimo specialistams R25 - Dokumentacijos sumažėjimas - nesant dokumentacijai nepagalvojama, kaip tam tikras pakeitimas paveiks kitas sistemos vietas R26 - Dokumentacijos versijos perrašymas tarp komandos narių R27 - Dokumentacijos neatnaujinimas – paliekama informacija, kuri nebėra tiksli ir galiojanti esamam funkcionalumui R28 - Dokumentacijos sumažėjimas - priklausomybė nuo komandos žmonių, turinčių žinias apie produkto įgyvendinimą
		A40 - Neuždarytų versijų skaičius	R54 - Versijos neatnaujinimas R55 - Versijos užduočių ne atsekamumas
		A47 - Procentas atliktų užduočių	R4 - Neišpildyti reikalavimai R7 - Neatsižvelgiama į komandos narių turimas žinias - komandoje nėra tinkamų specialistų R10 - Apmokymo rizika - ribotos žinios apie <i>Scrum</i> procesus R15 - Nesklaidi <i>Scrum</i> procesų integracija komandai

Nr.	Artefaktas	Priskirti atributai	Atributų rizikos
			R12 - Atsakomybės stoka didelėse komandose - neprisimama atsakomybė už terminų nesilaikymą ir sprintų vėlavimus

7 priedas. Interviu su projekto „DVS“ vystymo komanda

Interviu su Tomu:

Adelė: Kokia tavo patirtis informacinių sistemų kūrimo projektuose, bei jų valdyme? Kiek metų?
Tomas: Dirbu informacinių sistemų kūrimo projektų valdyme bei kūriau architektūrą programinei įrangai jau 20 metų.

Adelė Ar naudotumėtės standartinio projekto kūrimo funkcionalumu?

Tomas: Standartinio projekto kūrimo funkcionalumas gali būti naudingas tuom, kad iš karto yra apgalvojami atributų ryšiai su konkrečiomis rizikomis, kas su sistema nedirbusiam žmogui nėra akivaizdu, o jau padirbus su pačia sistema galima sąrašą papildyti, jeigu pastebim, kad kažko trūksta. Manau, kad tai tikrai pravartu.

Adelė: Kiek laiko skirtumėte darbui su sistema vieno sprinto metu?

Tomas: Pripažinsiu, šiuo metu rizikų valdymui su komanda neskiriame daug laiko – aptariame galimas grėsmes per kasdienį susitikimą ir viskas, todėl manau būtų sunku priprasti skirti laiko šiam projekto valdymo aspektui, bet jei jau turėtume tokį įrankį, manau porą valandų per sprintą galėčiau atrasti.

Adelė: Ar identifikuotos rizikos buvo aktualios ir tiksliai įvertintos?

Tomas: Taip, pastebiu, jog didžioji dauguma rizikų, kaip kad darbuotojų kaita ir visos su tuo susijusios grėsmės buvo aktualios mūsų projekte, o čia dar ir pastebėtos anksčiau, kai mes neatkreipėme dėmesio, todėl iš pirmo žvilgsnio manau, kad sistema tikrai veiksminga.

Adelė: Ar naudotumėtės šia sistema?

Tomas: Reikėtų laiko priprast, nes ir dabar jau turime sistemų iki kaklo, tačiau čia matau, jog sistema nereikalauja didelio įsitraukimo. Žinoma, kuo daugiau pildysi, tuo tikslesnius duomenis gausi, tačiau net ir iš rečiau pildomų duomenų gali atpažinti galimas rizikas, todėl manau naudočiau.

Adelė: Ar rekomenduotumėte šią sistemą kitiems?

Tomas: Rekomenduočiau tokiems, kurie neapleistų sistemos, nes tokį dalyką lengva pamiršti ir apleisti, o jau kas pusę metų atsidarius jau gali būt tam tikras rizikas ir pramiegojęs. Bet iš esmės, manau įrankis yra pravartus, intuityvus, todėl taip, rekomenduočiau

Adelė: Kokias stipriausias ir silpnąsias puses išskirtumėte šiai sistemai ir metodikai?

Tomas: Manau, stiprioji pusė yra sistemos intuityvumas ir aiškumas – tai yra pritaikyta dirbti ir projektų vadovui, ir Scrum meistriui, ir programuotojui su skirtingomis žiniomis. Patiko formulių perpanaudojimas – kad atributams apskaičiuoti nereikėtų lįsti į kodą, o viskas gali būti konfigūruojama naudotojo sąsajoje su aiškia visiems pažįstama formuluote. Visgi, manau kad kartais ši metodika yra per abstrakti – įvertinami tik skaičiukai ir iškart keliami rizika. Kai realiam gyvenime labai dažnai rizika gali būti tiesiog pajautimas, ko nelabai apskaičiuosi.

Adelė: Ačiū už pokalbį, Tomai.

Interviu su Skaiste:

Adelė: Kokia jūsų patirtis informacinių sistemų kūrimo projektuose, bei jų valdyme? Kiek metų?

Skaistė: Dirbu testuotoja 4 metus, daugiausiai prie programinės įrangos kūrimo.

Adelė: Ar naudotumėtės standartinio projekto kūrimo funkcionalumu?

Skaistė: Manau naudočiau. Nors manau, kad mano konkrečiame projekte dalis standartinio projekto ryšių yra pertekliniai, dalis rizikų neaktualių, bet jau geriau susikurti pilną ir išsitrinti, ko nereikia, nei kurti viską nuo nulio.

Adelė: Kiek laiko skirtumėte darbui su sistema vieno sprinto metu?

Skaistė: Skirčiau kelias valandas – prieš sprintą, patikrinti galimas grėsmes ir pasiruošti veiksmų planą, tada sprinto metu patikrinčiau objektų statusą, atnaujinchiau reikšmes ir tada jau retro. Taip pat manau, kad šis įrankis galėtų būti retrospektyvinių susitikimų įrankis – peržiūrėti rizikas, patikrinam, kaip sprendėm, kaip galim spręsti geriau.

Adelė: Gera idėja, Skaiste! Ar identifikuotos rizikos buvo aktualios ir tiksliai įvertintos?

Skaistė: Taip, rizikos buvo tikslios, nors dalis buvo įvertinta kaip perteklinė informacija, nes metodika nėra pajėgi įvertinti, jog rizikos nėra, nes, pavyzdžiui, darbuotojas serga, todėl ir krito retrospektyvi apimtis, ar kažkas tokio.

Adelė: Ar naudotumėtės šia sistema?

Skaistė: Taip, kai pir minėjau, naudočiau ir sprinto metu, o labiausiai retro apžvalgose, ir bandyčiau įtraukti visa komandą.

Adelė: Kokias stipriausias ir silpnąsias puses išskirtumėte šiai sistemai ir metodikai?

Skaistė: Stiprioji pusė tai manau informacijos kaupimas ir auditavimas. Nes realiam gyvenime, įvyko rizika, išsprendėm ir pamiršom iki kito karto, o čia gerai tas, kad yra saugoma ir istorinė informacija, ką galime analizuoti ir tikrinti, kaip šią situaciją sprendėm anksčiau, kas pasisėkė, kas nepasisėkė. Silpnoji vieta manau būtų, kad vis tiek neišvengiame rankinio duomenų įvedimo ir reikia būti atsargiems, kad nepamirštume įvesti vienu metu, kad informacija nebūtų netikslė.

Adelė: Ačiū jums už pokalbį.

Interviu su Alice (versta iš anglų kalbos):

Adelė: Kokia jūsų patirtis informacinių sistemų kūrimo projektuose, bei jų valdyme? Kiek metų?

Alice: Esu patyrusi projekto vadovė su 8 metų bei dirbau su įvairių dydžių ir sudėtingumo projektais.

Adelė: Ar naudotumėtės standartinio projekto kūrimo funkcionalumu?

Alice: Tikrai taip, aš kaip ne inžinierė turėčiau problemų savarankiškai sudėlioti ryšius tarp atributų, artefaktų, rizikų, todėl šis funkcionalumas man būtų pravartus.

Adelė: Kiek laiko skirtumėte darbui su sistema vieno sprinto metu?

Alice: Skirčiau pusdienį per sprintą, prieš sprinto pradžią susikelt duomenis, susidėlioti rizikas, veiksmų planą, galbūt suorganizuočiau sesiją su komanda apsitarti tolimesnius veiksmus sklandžiam sprintui vykdyti. Taip pat manau skirčiau laiko per retro, patikrinti, kaip pavyko išspręsti ir ką galėjome daryti geriau.

Adelė: Ar identifikuotos rizikos buvo aktualios ir tiksliai įvertintos?

Alice: Taip, kiek peržiūrėjau, rizikos buvo tikslios ir mes komandoje jas pastebėjome per vėlai, todėl mano nuomone, sistema yra efektyvi.

Adelė: Ar naudotumėtės šia sistema?

Alice: Taip, naudočiau. Tačiau manau, kad tokia sistema būtų aktualesnė ilgai besivystantiems projektams. Esu dirbusi naujuose projektuose, ten mes į rizikas kreipiame didelį dėmesį, patys jas atpažįstame ir fiksuojame, na o ilguose projektuose jau dirbi iš inercijos, todėl lengva užmigti ir nepastebėti, kad projekte atsiranda spragų – pastebi jau po laiko.

Adelė: Kokias stipriausias ir silpnąsias puses išskirtumėte šiai sistemai ir metodikai?

Alice: Stiprioji dalis – intuityvumas ir patogi naudotojo sąsaja. Parodyta švieslentė puikiai tiktų prezentacijoms tiek klientui, tiek vidinei komandai. Silpnoji pusė – daug rizikų yra sunkiai įvertinamos skaičiais, todėl galiu teigti, jog metodika nepagauna visų rizikų, ypač susijusių su darbuotojų nuotaikom, morale, nepasitenkinimu. Tai yra toks dalykas, ką tu gali pajauti atėjęs į kambarį tarp žmonių ir pasiklausęs, ką jie šneka, o ne per Jiros failą.

Adelė: Ačiū už pokalbį, Alice.

Interviu su Simone (versta iš anglų kalbos):

Adelė: Kokia jūsų patirtis informacinių sistemų kūrimo projektuose, bei jų valdyme? Kiek metų?

Simone: Esu dizaineris su 7 metų patirtimi informacinių sistemų kūrimo projektuose. Kūriau įvairaus tipo dizaino sprendimus – mobilios aplikacijos, web aplikacijos, platformos.

Adelė: Ar naudotumėtės standartinio projekto kūrimo funkcionalumu?

Simone: Manau ne, nes mano sritis šioje metodikoje yra ganėtinai siaura ir sekčiau tik kelis atributus ir kelias rizikas, todėl man naudojant standartinį projektą tektų jį stipriai prasivalyti.

Adelė: Kiek laiko skirtumėte darbui su sistema vieno sprinto metu?

Simone: Kadangi pats dirbu ne sprintais, negaliu tiksliai įvertinti, kiek laiko praleisčiau per vieną sprintą, bet sakyčiau iki valandos.

Adelė: Ar naudotumėtės šia sistema?

Simone: Taip, tačiau manau, jog ganėtinai retai, jei projektas vystomas sklandžiai, galbūt ir apleisčiau sistemą.

Adelė: Kokias stipriąsias ir silpnašias puses išskirtumėte šiai sistemai ir metodikai?

Simone: Stiprioji pusė – intuityvus naudojimas, paprasta suprasti tiek programuotojui, tiek dizaineriui. Taip pat duomenų saugojimas, istorinės informacijos prieinamumas ir palyginimas. Silpnoji pusė – naudotojo įtraukimas, jei būtų tiesioginė integracija su projektų valdymo Sistema ir automatinis duomenų atnaujinimas, būtų išvis šaunu.

Adelė: Ačiū už pokalbį, Simone.

Interviu su Milda:

Adelė: Kokia jūsų patirtis informacinių sistemų kūrimo projektuose, bei jų valdyme? Kiek metų?

Milda: Esu Scrum meistrė su 7 metų patirtimi informacinių sistemų kūrimo projektuose.

Adelė: Ar naudotumėtės standartinio projekto kūrimo funkcionalumu?

Milda: Tikrai taip. Manau, kad daug pavaizduotų ryšių yra tikslūs ir aktualūs, tačiau pati gal ir nebūčiau greitai sugalvojus, todėl įsivažiavimui dirbti su sistema jau sutvarkyti ryšiai būtų pravartūs.

Adelė: Kiek laiko skirtumėte darbui su sistema vieno sprinto metu?

Milda: Spėčiau kelias valandas – sprinto pražioje, eigoje bei per retro. Ypač patogu per retro atsidaryti švieslentę ir patikrinti, kaip vykdėme sau priskirtus žingsnius, kaip pavyko susidoroti su nesklaidumais, kur galėtume pasitempti.

Adelė: Ar identifikuotos rizikos buvo aktualios ir tiksliai įvertintos?

Milda: Tikrai taip, didžioji dauguma rizikų buvo aktualios, ir turbūt tokios, į kurias įprastai komandoje net neatkreiptume dėmesio, o čia jos yra tiksliai identifikuotos.

Adelė: Ar naudotumėtės šia sistema?

Milda: Taip, naudočiau ir rekomenduočiau šią rizikų valdymo metodiką naudoti ir ateityje. Ši metodika yra pakankamai lanksti ir leidžia adaptuotis prie besikeičiančių projekto aplinkybių. Be to, ji yra integruota į Scrum ciklą, todėl rizikų valdymas tampa natūralia projekto dalimi.

Adelė: Kokias stipriausias ir silpnąsias puses išskirtumėte šiai sistemai ir metodikai?

Milda: Stiprioji pusė – lankstumas, sistema yra pritaikyta prie bet kokio pobūdžio projekto, gali susikonfigūruoti kaip tik nori ir manau gerai, jog pritaikytos pažistamos formulės. Kaip silpnąbę išskirčiau tai, jog dalis rizikų kyla iš tam tikrų diskusijų su klientų, iš tam tikrų diskusijų su komandos žmonėmis, o ne iš apskaičiuojamų duomenų, todėl tokias rizikas reikėtų identifikuoti patiems.

Adelė: Ačiū už pokalbį, Milda.

8 priedas. Straipsnis, pateiktas į ICIST 2024 konferenciją

Risk Management Methodology in Agile Projects using Calculations based on Attributes

Adele Necionyte and Lina Bisikirskiene

Kaunas University of Technology, Studentu str. 50, 51368 Kaunas, Lithuania
lina.bisikirskiene@ktu.edu

Abstract. This article discusses the importance of effective risk management in Agile projects. The Agile project management philosophy emphasizes open communication, flexibility in product development, and rapid market delivery. Unlike traditional waterfall methods, Agile iterative approach reduces risks by allowing on-going progress assessment. However, Agile alone may not effectively manage risks, especially those stemming from human factors and environmental influences. Existing project management tools lack comprehensive risk identification and management solutions, particularly concerning diverse project components and stakeholder engagement. Decentralization in Agile projects can lead to architectural risks due to documentation inconsistencies and client disengagement. To address these challenges, this article proposes a methodology for identifying risks of Agile project associated with artifacts and attributes of it. By regularly updating attributes' data, potential risks can be unveiled and addressed early, enhancing project team effectiveness and stakeholder clarity through transparent reporting.

Keywords: Agile Project, Artifacts, Attributes, Jira, Risk, Risk Management

1 Introduction

The Agile project development and management philosophy is popular and often applied to various projects. The main principles of this project management philosophy are open communication, the flexibility to evaluate and choose the best way of product development and deliver fast results to the market [1]. The iterative project management method, one of the essential differences between Agile and the traditional waterfall management methodology, reduces risks, but does not eliminate them to the fullest. Risks not identified and managed in time can cause negative consequences for the project's further course, such as exceeding the budget, late deadlines, and overworked human resources.

Agile project management methodology by itself is not sufficient to eliminate risks. Although the iterative methodology provides an opportunity to follow and regularly check the project's progress to notice inconsistencies, it may not be enough to manage risks. Agile project management methodology does not create an effective mechanism for solving project problems that arise in the later stages of project management, nor

does it define a unified methodology for managing risks for different project components.

Although the identification and management of risks is one of the key factors of project success, modern project management tools do not have a single solution that can define, timely identify, and manage potential risks that may arise during the project. One of the main reasons for this is that the main factor that determines the success of the project is the project team (together with different cultural environments, work processes, technology standards, different interests, and goals). So, the risk category of human factors can be difficult to perceive and manage [2]. Also, Agile project management structure can lead to additional risks in architecture management. Since the decentralization of information and requirements is emphasized in the projects, it may lead to a lack of documentation, consistency, and clarity e.g., functionality descriptions. It is also worth highlighting the category of risks - environmental risks that arise from the insufficient involvement of clients and other interested parties in the project [3].

Existing tools do not provide an environment for the identification and classification of risks in different processes, such as requirements definition and analysis, process and design modeling, work list creation and prioritization, work breakdown and acceptance criteria description, testing, and documentation. It is not sufficient to have recommendations to talk about risks in ceremonies and identify them once they appear. Since Agile projects contain data about the progress of the project, it can be used to track and predict upcoming risks. Agile projects are not an exception and require managing risks based on risk lifecycle (identification, control, and management, tracking and documentation) ensuring flexibility and agility. It means that the efficiency of risk identification and transparent management can be a success factor in a project.

This article presents the methodology that lets us define an Agile risk associated with Agile project artifacts and their attributes. Regular updates of attributes' data may unveil the potential risks and ensure that they will be solved in the initial stages. The project team gets valuable experience in being effective in risk management using predefined Agile risks and not wasting time on checking if the risk exists. Transparent reporting gives clarity to the team and stakeholders.

2 Literature

A project is a temporary and unique endeavor designed to achieve a specific goal. Projects are usually initiated to create or deliver a product, service, or result that is different from the usual routine operations of an organization. Projects have a defined scope, budget, and timeline, and they typically involve a team of people working together to complete the project [4]. Agile project life cycle is a framework for managing and delivering projects in an iterative and incremental approach [5]. The Agile methodology emphasizes flexibility, adaptability, and customer satisfaction.

The Agile Project Management Manifesto consists of four core values, which are broken down into twelve core principles that underpin each of the Agile project management methodologies [6]. The highest priority is to meet the customer's needs on time and deliver the software continuously. It is important to be receptive to changing

requirements, even late in development. Agile processes enable change to be used for the customer's competitive advantage. Business stakeholders and developers must work together every day throughout the project to keep the teams motivated and have the right environment, support, and trust. Any kind of bureaucracy prevents success and reduces team effectiveness. Therefore, attention is constantly paid to both technical excellence and useful design, which ensures flexibility in product development. So, success lies in simplicity - the art of maximizing the amount of undone work and focusing on the main work. Self-organizing teams regularly reflect on how to become more effective, so they adjust and adapt their behavior accordingly to achieve the best result in the fastest way.

2.1 Scrum and Kanban Methodologies

Even though Agile ensures project flexibility, it does not prevent the occurrence of risks. The two main Agile methodologies that are offered are Scrum and Kanban. Experimentally, these two methodologies were compared by A. Granulo and A. Tanovic [7] and M. Alquadi [8], paying attention to the definition of work principles, separation of roles and responsibilities, reception time, team size, amount of assigned work, prioritization, performance of work time, costs, and quality. The main difference is the size of the team. If it is needed to have a larger team or even several ones, then the Scrum methodology gives more effective control of the situation and better management of the potential risks [9]. A smaller team can achieve better results more efficiently when working with Kanban.

The success factor of Scrum projects is the implementation process that is carried out in iterations – sprints (in most cases two weeks). Each sprint is covered by the main sprint events – ceremonies. These events are designed to provide transparency and check the status of the project. If any event is unfulfilled, the transparency of the project progress will decrease and the opportunity to check the status of the project and identify anticipated risks will be lost [9].

K. Schwaber and J. Sutherland [9] agree that the iterative method used in Scrum optimizes risk control. Using sprints helps manage risk by limiting risk to one calendar month's worth of costs. However, the counterargument is that although this methodology helps identify risks more easily, it does not eliminate them [10]. So, the main recommendations for risk management in Scrum projects are the following [11]:

- Continuous feedback about project components is the key factor for project success. Potential risks must be identified during each sprint. Problems must be identified at the early stages.
- Reuse of knowledge about risks and lessons learned must be ensured. It is a continuous process with defined documentation.
- Recognition that risk management is more than a methodological process. Individual behavior and collaboration are important to Scrum project development. Scrum is more about people, behavior, and culture than about the processes.

2.2 Key risks in Agile projects

Risk management must be one of the key processes in software development. Agile projects are not the exception. These projects use iterative stages of project development, which aim to reduce the risk of uncertainty, set priorities, and confirm them with the customer during each iteration. These practices reduce or eliminate the risks that may arise when developing a project using traditional methods with separate analysis, development, and testing phases. Simplification in Agile may bring different types of risks and also give space for analyzing risks that appear in the wider ecosystem [12, 13]. Below are the main risks identified in projects developed according to the Scrum methodology [6, 14, 15]:

- Undefined requirements;
- Risks related to the project team (poor management of the team and its structure, uninsured cooperation, effectiveness, and consistency of training, not identified and not assured responsibilities, not assured trust);
- Risks related to the development process (lack of understanding and control of Scrum processes, lack of a testing strategy, e.g., unit and integration testing, lack of definition of code quality and standards, lack of documentation, improper project scope assessment and planning principles);
- Environmental factors (lack of customer involvement, shortcomings of the work environment, complexity of interdependencies between product modules, independence of modules, incompatibility of work tools);
- Applied technologies (communication challenges, lack of knowledge sharing, lack of error tracking process, inefficiency of version tracking process).

The list of risks cannot be finalized. Each team should pay attention to their project and review the list of risks. Risks can be analyzed within the team, the product, and the organization. It is important to assess at which level of the hierarchy this risk is and what are the possible ways to solve it.

2.3 Risk Management

To properly control risks, attention must be paid to their life cycle. Risk can be defined as deviations of actual results from expected ones. Usually associated with the probability of an undesirable, worthless, or harmful result, error, or loss. It is the effect of uncertainty on a change, decision, or company. The project team collaborates with other stakeholders to identify and assess risks, prioritize them, and address these risks by changing the conditions or events that give rise to the uncertainty: reducing the consequences, eliminating the source of the risk, completely avoiding the risk, not starting or continuing activities that cause risk [16]. Risk has three essential components: uncertainty, loss, and limited duration. There is always uncertainty as to whether a risk will occur. Since the risk may not occur, it is possible to manage and eliminate its occurrence [17].

In today's software projects, risk management is not just a thing that can be done if time and resources are available. This means that risk management must be

implemented at the beginning of the project, and risks must be assessed and identified throughout the development cycle [18].

Effective risk management includes identification, control and management; and tracking and documentation.

Identification. Identification of risk is necessary to control it. There are various methods that can be used to determine risk. Some of the most popular risk identification techniques include comprehensive risk checklists, document reviews, and analysis of assumptions and constraints. Most often, processes related to risk management are initiated by the project manager and provide the team with information on how to identify and manage emerging risks.

Control and management. After identifying the risk, a strategy is needed to control and manage the risk. Some methods suggest how to define risk for diverse types of projects. In traditional project management methods, a risk register is recommended to track how risks are managed and controlled [17]. This approach defines key risk attributes but requires the involvement of the entire team. Risk burndown charts can also be used, which depict the risk, the probability of the risk, the size of the loss (the amount of time lost if the risk were to occur), and the impact.

One of the most common risk analysis and management techniques used by Agile teams is the "What and Why" approach. The team organizes a meeting to find out what could happen in the project and then analyze the causes of the event. In the first case, the consequences of the occurrence of the risk are determined, and in the second case, the risk factors and circumstances are identified. The main advantage of this approach is its simplicity, especially for teams that may not otherwise be familiar with specialized risk management practices [14]. At the same time, it distinguishes that risk can be managed by evaluating it as a separate component during the sprint, or as a separate task. It is also possible to identify risk's tasks with certain color indicators that help determine risk priority and status. These indicators can also be linked to existing tasks.

Tracking and documentation. Risks should be tracked and documented throughout the process. Daily Scrum meetings are often used for these practices. Risk documentation is typically done in risk registers or after risk assessment as an additional sprint task. Risk monitoring provides information necessary to determine the approach to risk and further actions (avoid, accept, or exploit the risk). While some risk management and tracking include actions such as additional tasks to manage risk during the sprint, some risk management models require consideration and adjustment of certain project management and execution procedures. For example, for the risk of non-compliance with requirements during programming, teams use pair programming - the code for the implementation of sprint tasks must be checked by two programmers.

Although there are many recommendations for successful risk management in an Agile project, their practical application is small and ineffective [19]. This is due to several fundamental reasons:

- Lack of time for risk identification and control.
- Lack of data for risk identification.
- Historical data analysis and forecasting are not supported.
- Belief that all risks can be managed within the sprint.

To eliminate these problems, it is necessary to combine the principles of risk management, the types of risks detected in Agile projects, and the project artifacts with attributes, so that team members could have an effective infrastructure for identifying, registering, and eliminating risks. A methodology that incorporates these principles is presented below.

3 Methodology for Risk Management using Calculations based on Attributes

The methodology consists of two processes dedicated for two roles: Subject Matter Expert (SME) and Team Member (TM). The configuration process for SME consists of preparing a default risk management project with the predefined lists of risks, their categories, project artifacts, their attributes, and recommended risks for the attributes where the probability of occurrence of risks is highest and is necessary for an efficient risk identification process. This prepares the environment for the identification, analysis and tracking of risks. Team Member (e.g. product manager or scrum master) may use the default project, adjust it and start tracking risks based on the values of the attributes of the project. There is no specific role for the Scrum team to work on this process because this approach embraces transparency of the project status among the team members and the involvement of each member.

3.1 Process for Subject Matter Expert

A risk subject matter expert creates a Scrum project template, which can then be adapted to a specific project or modified according to the specifics of the project. The algorithm of the process is given in Figure 1. and described below.

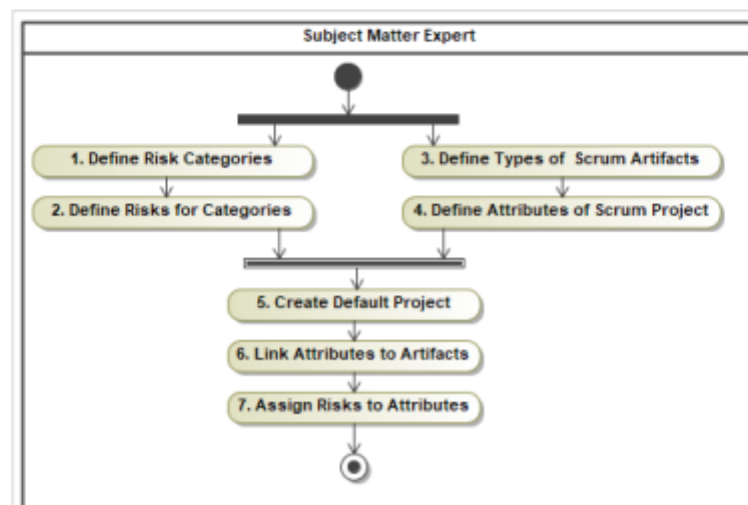


Fig. 1. The process of risk subject matter expert

Step 1. “Define Risk Categories”. The main risk categories that exist in the Scrum project management methodology are identified. There are 18 Risk Categories in the default project. The Risk Category contains Number, Name, Description. The examples of Risk Categories could be Requirements related risks, Communication risks and etc.

Step 2. “Define Risks for Categories”. Each Risk Category contains Risks. The Risk contains Number, Definition and parental Risk Category. The example of Risks is given in the Table 1.

Table 1. Risks added to Risk Category “Formal code standard risks”

No.	Definition	Category
R20	Lack of a formal code standard - difficult to maintain code	Formal code standard risks
R21	Lack of a formal code standard – difficult to understand other programmers' code	Formal code standard risks
R22	Lack of formal code standard - reduced performance	Formal code standard risks
R23	Lack of formal code standard - technical debt not supported	Formal code standard risks

Step 3. “Define Types of Scrum Artifacts”. Initially the default types of Scrum artifacts that exist in Scrum methodology are identified. The list of recommended artifact’s types is displayed below in the Table 2. Each artifact has a Number, Name and Description. There are 12 artifacts defined in the default project.

Table 2. List of artifacts of Scrum project

No.	Artifact	Description
AR1	Product Backlog	A prioritized list of work for the development team derived from the roadmap and its requirements.
AR2	Product Goal	A measurable and actionable target describing what the Scrum Team aims to build in the months ahead and for what purpose.
AR3	Epic Discovery	A process whereby sessions are conducted with stakeholders to clarify the high-level needs of the customer for the implementation of a given functionality module.
AR4	Epic Implementation	The process, carried out by a Scrum project team during a sprint or multiple sprints, to ensure that all user stories in a functionality module are consistent with the definition of work done.
AR5	Technical Debt	A list of tasks to maintain product architecture and code hygiene.

No.	Artifact	Description
AR6	Design Discovery	The process of gathering detailed requirements for a product design and compiling a list of user stories.
AR7	Sprint Backlog	The sprint to-do list consists of the sprint goal, a set of items from the list of products selected for the sprint, and an implementable plan to achieve the product increment.
AR8	Sprint Goal	A commitment from the developers to perform certain tasks that will bring the team closer to the project goal.
AR9	Product Increment	A set of tasks that meet the definition of done in a sprint.
AR10	Roadmap	A strategic plan that defines the objective or desired outcome and identifies the key steps or milestones needed to achieve it.
AR11	Release Plan	A set of information to help define the scope, milestones, and timing of a release.
AR12	Onboarding	Task to involve new team members in the project.

Step 4. “Define Attributes of Scrum Project”. The main attributes that exist for Artifacts in the Scrum Project Management methodology are identified. Since the same attribute can be used for several artifacts, they are identified separately and later linked to the artifacts. Example could be an attribute “Count of reopened tasks” can be tied both to Epic Implementation and Sprint Backlog artifacts to identify the underlying problem which requirements gathering, allocation of responsibilities between people with different seniority levels, lack of communication from the client. The attribute contains Number, Name, Attribute Type, Expression, and Applied Risks. An example of the attributes associated with artifact Sprint Backlog (AR7) is given in Table 3. The Attribute Type defines how attribute gets its value and it can be selected from these options:

- Picklist – attribute gets one of the predefined values. Mostly is used for attributes to define the statuses of readiness, EHRs, project health metrics.
- Integer – attribute gets a number value that is entered manually by the user.
- Calculated value – attribute gets a value that is imported from the external sources (e.g. JIRA) using exported Excel spreadsheets or calculated using values of other attributes.

The Expression is used for calculated attributes only to define the formulas how attribute value is going to be determined. As an example, the list of attributes recommended for the Artifact “Sprint Backlog”, is given in below in Table 3. The attribute “A6 Standard deviation of velocity” is calculated from attribute’s “A5 Velocity” values using standard deviation formula. It is selecting range of three values meaning that

deviation is calculated from three latest velocity values. Attribute “A7 DoR tasks count” identifies count of tasks meeting definition of ready. It is calculated counting data rows in the import file with the condition satisfied where column’s “Label” value is “ClientApproved”. It shows that tasks has been reviewed and approved by project team and client stakeholders. The Applied Risks identify Risks that will be followed in the project. SME linked Attributes with Risks based on literature [6, 14, 15] and experience. Team Member may define his own linking of Attributes and Risks in his project. One Attribute may have an association with several Risks. At the same time, Risk may have an association with several Attributes.

Table 3. List of attributes recommended for AR7 with linked Risks

No.	Name	Attribute type	Expression	Applied risks
A5	Velocity	Integer	N/A	Used for calculations
A6	St. dev. of velocity	Calculated	ST.DEV(A5, 3)	R5, R22, R32, R59
A7	DoR tasks count	Calculated	COUNTIF (Label, "ClientApproved")	R1, R39, R47, R49
A10	Estimated work	Calculated	SUM("StoryPoint")	R57, R58
A11	Estimate and Velocity delta	Calculated	A5-A10	R62, R63, R64
A45	% of critical tasks	Calculated	COUNTIF (Priority, "Critical")	R46, R47
A46	% of low priority tasks	Calculated	COUNTIF (Priority, "Low")	R46, R47
A49	% given for tech debt	Integer	N/A	R20, R21, R22, R23
A37	Hotfix count	Integer	N/A	R40

Step 5. “Create Default Project”. The default project is created. It will be used as an environment to store the default mappings of artifacts, attributes, and risks.

Step 6. “Link Attributes to Artifacts”. Each of the Artifacts from the list gets the assigned attributes whose data will be tracked in the project. All selected Artifacts must have at least one attribute assigned, otherwise it will not be possible to track the status of this Artifact.

Step 7. “Assign Risks to Attributes”. After selecting the Artifacts of the project and their attributes, it is possible to assign a risk to each of the attributes from the list. Once the assignment is done, the Risk gets an additional attribute Trigger Value which shows when the potential Risk appears, e. g. Trigger Value for risk R23 in Table 1. could be 15% of sprint tasks and stories meaning that if the actual value is less than 15% the attribute is considered risky.

3.2 Process for Team Member

Once the default project is created, it can be used by the project team. Team member's process consists of four parts described below and presented in Figure 2.

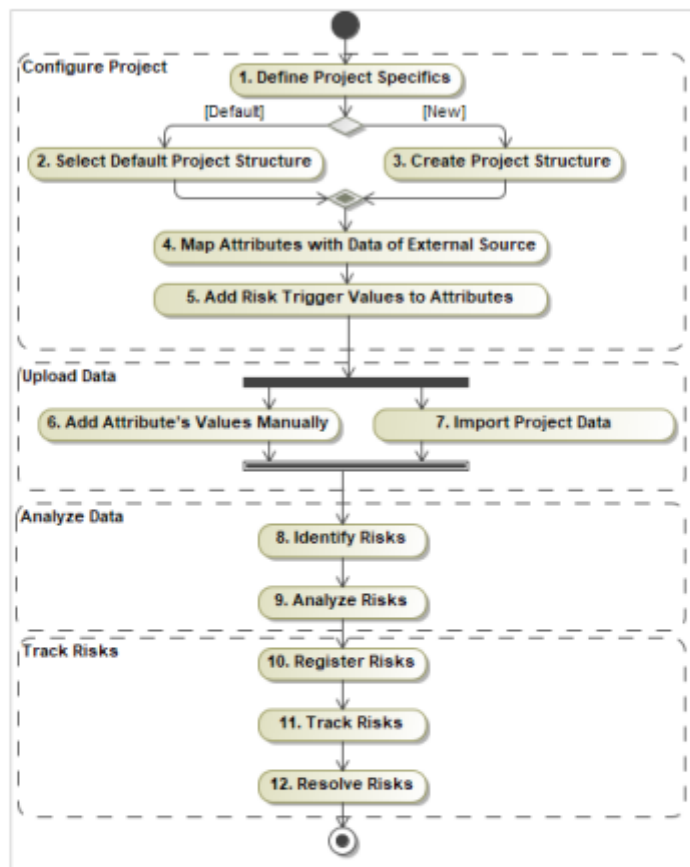


Fig. 2. The process of the project risk management for team member

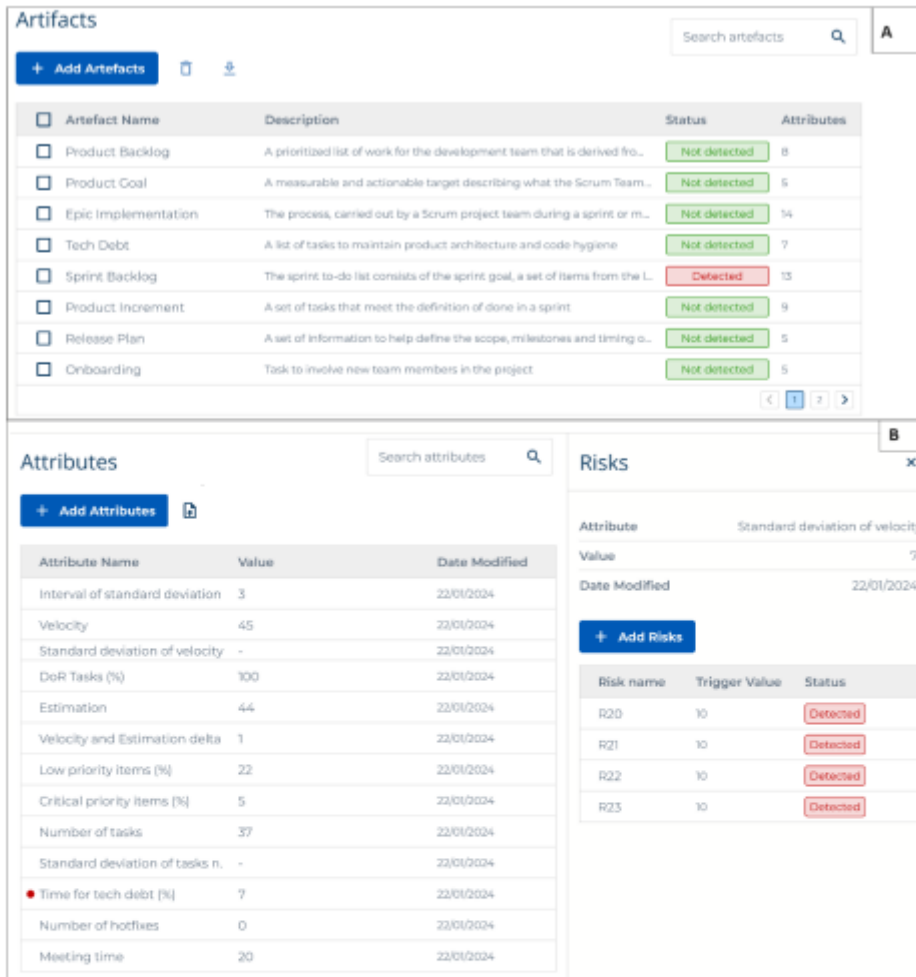
Configure project. To kick off working the user needs to create the project (Step 1. Define Project Specifics) and select either default project structure (Step 2. Select Default Project Structure) or build project from scratch (Step 3. Create Project Structure). By selecting default one user needs to choose Artifacts that are relevant in their work area. The attributes with recommended risks are already tied to the Artifact based on the mapping created by SME. If the user wants to use his own structure, he may create additional artifacts, attributes, risk categories, and risks. It means, that he has a full flexibility to create structure and mappings that reflects his own project. From the practical perspective, if the teams in the company use the same Scrum principles and tooling for their projects, they can make the setup once and reuse it for other projects.

The next step is to define external sources that give data about the project (Step 4. Map Attributes with Data of External Source). The user has to select the import file template for each of the Artifacts. The system maps required attribute values and expressions for the file imports, e.g. JIRA allows to export several types of Excel spreadsheets with the information about the project. The user has to link attributes from the system to attributes given by JIRA. The default mapping is given in the methodology, e.g. Velocity, but at the same time the user may define their own mapping. This step allows the user to create deductible attributes that are using values of other attributes, e.g. deviation of velocity which calculates the standard deviation using three last velocity values and applying standard deviation formula. The last action of configuration is to identify the trigger value for the risk at which point attribute is considered risky (Step 5. Add Risk Trigger Values to Attributes). Besides the value itself user needs to select operator of the Trigger value (greater than, less than, equal, not equal, in range of). This action can be done by the project team, because in each project they can identify different values of the same risk.

Upload data. After the configuration part is done, the following action to be done by user is data uploading. Data upload is done periodically. In most cases bi-monthly, since the sprint takes two weeks, but the team may decide differently and track data more often. Making it less frequent is not recommended in Agile projects because it may cause losing the pulse of the project. Team members may enter the attribute value manually in most cases if it is a picklist (Step 6. Add Attribute's Values Manually) or upload it though the file (Step 7. Import Project Data). The system can import the data and recognize the changes in the project, unexpected fluctuations of the variables and alert the user. Important part to notice for system users – both calculated from import file and manually entered attributes should be updated at the same time, because attributes that are derived from other values could be conflicted.

Analyze data. When system is populated with the data, the following part of the process is reached – system provides the risks identified based on the data added and attribute triggers selected (Step 8. Identify Risks). However, the risk is not logged automatically because team members need to evaluate (Step 9. Analyze Risks) whether highlighted Artifact with identified risk is at risk, e.g. significant drop in velocity in one data upload can cause concern but if team member is aware of holiday season for the organization, this velocity fluctuation is expected and should not be considered as risky item. An example of the list of artifacts for the project in the prototype is shown in Figure 3. A. "Sprint Backlog" artifact has the detected risks. An example of the list of attributes with the ability to register the identified risks is shown in Figure 3. B.

Track risks. If the user decides that variation of data is not expected, he can raise risk (Step 10. Register Risks). The risk is logged by adding risk parameters: impact, priority, responsible person, and action items to mitigate the risk. After the risk is detected, it is displayed in the dashboard of the system and requires further attention of the project team. It should be assigned to a responsible person until it is resolved. Project team tracks the following changes in data in upcoming data upload events and identifies if situation with Artifact is resolving (Step 11. Track Risks). The parameters of the risk are updated to follow its history. Once the risk is solved, it is marked in the system (Step 12. Resolve Risks).



**Fig. 3. A. List of artifacts for the project with the detected risks
B. List of attributes with the ability to register risks**

The user may import data and analyze it as many times as he needs. It helps to track the status of the project without making any manual calculations or comparisons with the historical data. Transparent usage of project status from various aspects enables team members to review project status more effectively avoiding misinterpretations.

4 Experiments

The prototype of the methodology is a web application. It supports all the main steps of the algorithm and allows the user to track the status of the project by identifying risky artifacts. Project data is uploaded from project management system JIRA using

Excel spreadsheets or entered manually. The experiments are dedicated to answering these research questions: can the methodology increase the number of the identified risks; can historical data of risks and attributes related to them unveil hidden long-term risks. The experiments are described below.

Project “DVS”. Development of a candidates’ selection and evaluation platform. Project goal: Developing an online platform for assessing employees and candidates to conduct a transparent and objective employee evaluation process, thus preserving the company’s reputation and employee morale. The platform works by means of individual tests and surveys completed by assigned users. The results of the tests are used to generate reports, dashboards, and advice for personal and professional development. Client: UK Employment Agency. Team size: 20 people. Timer period: 1 Year.

Project “EBS”. Development of an e-banking and mobile banking app. Project goal: Developing an online platform to provide all client users to access their financial accounts, complete domestic and international transactions, create savings, shared and accounts of other purpose conveniently and securely. Platform is enriched by additional features such as money requesting, financial insights. The primary objectives are to enhance user experience, streamline banking processes, and ensure security measures to safeguard sensitive financial information. Client USA Bank. Team size: 12 people. Timer period: 10 months.

A summary of the experiments is given in Table 4. reflecting the usage of default artifact and attributes with risks linked in specific project comparing with the full classification.

Table 4. Summary of Experiments

No.	Quantitative indicator	Project 1	Project 2
1.	% of used unique Artifact types	75	83
2.	% of used unique Attribute types	82	84
3.	% of relinked Attributes in Artifacts	152	159
4.	% of used Risk Categories	95	95
5.	% of used Risks	83	86
6.	% of relinked Risks with Attributes	381	395
7.	Count of events of Data upload (monthly/sprint)	12/24	10/20
8.	Count of Attributes with detected Risks	35	37
9.	Count of detected Risks	66	60
10.	Count of logged Risks	47	47
11.	Average time until Risk is resolved	1 month	2 weeks

It can be seen that the usage of Artifacts, Attributes, Risk categories and Risks in projects is between 75% - 95%, which allows to say that it was possible to apply most of the parameters of the methodology (Table 4. Rows 1, 2, 4, 5). The most frequent used attributes were A5 – Velocity, A6 – Standard deviation of velocity. After performing regular monthly and bi-weekly data uploads (Row 7), Attributes that potentially

have risks were identified (Row 8). The number of identified risks is presented in Row 9. After analyzing the risks, some of them were rejected, but most of them (71% - 78%) were logged. The most frequent logged risk was R63 Unpredictability of sprint performance - inadequate forecasting (six times).

Conclusions based on the experiments

According to the teams' feedback, their biggest discovery was that they underestimated the impact of risks that were long-term ones. Historical data allowed them to uncover long-standing problems that didn't seem like a big problem in the context of a single sprint. With the use of deducible attributes, teams began to assess the status of the project more responsibly. All team members were included in the risk analysis stage, thus eliminating situations where only senior team members examine risks and make decisions themselves.

From the methodology perspective, once it was created, the initially identified set of Artifacts was heavily reduced to make sure that the set of Artifacts defined in the analytical part of the research work is complete and that no further decomposition is needed. Some artifacts became attributes. Teams do not need a long list that could be difficult to track. Also, it was identified that once duplicate attributes are assigned to different project Artifacts, then once importing data the same risk indications were recorded for different Artifacts due to the risky value of duplicate attributes. To avoid this, the team member needs to assess which risks have already been captured in the project and omit the duplicated events.

The identified risks were not always resolved before the next import file upload event, so the risk is reidentified in the system because of the risky attribute value during the next import. The additional capture of a risky attribute may be useful to the team as it indicates the status of the attribute and whether the situation is improving, but it should be appreciated that the risk has already been captured.

5 Conclusions and directions for future research

In Agile projects, risk management is just as complex as traditional waterfall methods. Rapid code development underestimates the existence of long-term risks by not paying enough attention to project parameters whose values indicate inefficiencies, nor by recording information that is not parameterized, such as team comfort status. The literature focuses on describing potential risks, but the typical risk tracking suitable for traditional approaches is not acceptable for an Agile project as it seems forced and bureaucratic. The proposed methodology allows the usage of external sources that collect project attribute information and track project risks by applying derivative calculations. The creation of a default project ensures a quick start of the methodology. The methodology's flexibility ensures more successful application in the project if additional modifications are required. Experiments have shown that the risks in Agile projects are not eliminated due to the Scrum process. Teams admitted that they were not tracking long-term risks and did not pay attention to register Risks expecting to solve them in Scrum ceremonies. So, teams need to react and deal with risks effectively, otherwise the development process can be terminated.

Further research will be aimed at improving synchronization with JIRA and tools. It should ensure the ability to flag risks in the external sources (e.g. JIRA) with the possibility of turning them into tasks. Also, it will be aimed at expanding the set of calculated attributes, so that they cover the wider project eco-system, taking into account the needs of the projects.

References

1. Manifesto for Agile software development. Link: <https://agilemanifesto.org/> (2001)
2. Boehm, B., Turner, R.: Management challenges to implementing agile processes in traditional development organizations. *Journal IEEE Software* vol. 22(5), 30-39 (2005)
3. Odzaly, E. E., Greer, D., & Stewart, D.: AGILE risk management using software agents. *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, 823–841, (2018)
4. Cooke-Davies T.: The “real” success factors on projects. *International Journal of Project Management* 20(3), 185-190 (2002)
5. Beng Leau, Y., Khong Loo, W., Yip Tham, W., Fun, S.: Software Development Life Cycle AGILE vs Traditional Approaches. *IPCSIT* vol. 37, 162 – 167 (2012)
6. Project Management Institute.: Agile practice guide (2017)
7. Granulo, A., Tanovic, A.: Comparison of Scrum and Kanban in the learning management system implementation process. In *Proceed. of 27th Telecommunications Forum* (2019)
8. Alquadah, M., Razali, R.: A comparison of SCRUM and KANBAN for identifying their selection factors. In *Proceedings of 6th International Conference on Electrical Engineering and Informatics* (2018)
9. Schwaber, K., Jeff S.: *The Scrum Guide: The Definitive the Rules of the Game*. Scrum.Org and ScrumInc 11: 19 (2017)
10. Tavares, B. G., Sanches da Silva, C. E., Diniz de Souza, A.: Risk management analysis in SCRUM software projects. *International Transactions in Operational Research* vol 26 (5) 1884-1905 (2019)
11. Lei, H., Ganjeizadeh, F., Jayachandran, P.K., Ozcan, P.: A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing* vol. 43, 59-67 (2017)
12. Pereira, J. C., Rosaria de F. S. M. Russo.: Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Computer Science* vol. 138: 775–82 (2018)
13. Trzeciak, M.: Sustainable Risk Management in I.T. Enterprises. *Risks* vol 9 (7) (2021)
14. Moran, A.: Agile Risk Management. In: *Agile Risk Management*. SpringerBriefs in Computer Science. Springer (2014)
15. Cho, J.J.: Issues and challenges of agile software development with scrum. *Issues in Information Systems*, IX(2), 188-195 (2008)
16. BABOK. A guide to the business analysis body of knowledge. IIBA - international institute of business analysis (2015)
17. Singh, M., Saxena, R.: Risk management in agile model. *IOSR Journal of Computer Engineering* vol. 16(2) (2014)
18. Tomanek, M., Juricek, J.: Project risk management model based on Prince2 and Scrum frameworks. *International Journal of Software Engineering & Applications* 6, 81-88 (2015)
19. Mudumba, V., Lee, D.: A new perspective on GSD risk management. In *proceedings of the 5th IEEE International Conference on Global Software Engineering* (2010)