



Kauno technologijos universitetas

Ekonomikos ir verslo fakultetas

Reikalavimų inžinerijos procesų tobulinimas IT projektuose

Baigiamasis magistro projektas

Lukas Kusta

Projekto autorius

Doc. dr. Ramunė Čiarnienė

Vadovė

Kaunas, 2025



Kauno technologijos universitetas

Ekonomikos ir verslo fakultetas

Reikalavimų inžinerijos procesų tobulinimas IT projektuose

Baigiamasis magistro projektas

Inovacijų valdymas ir antreprenerystė (6211LX031)

Lukas Kusta

Projekto autorius

Doc. dr. Ramunė Čiarnienė

Vadovė

Prof. dr. Rūta Čiutienė

Recenzentė

Kaunas, 2025



Kauno technologijos universitetas

Ekonomikos ir verslo fakultetas

Lukas Kusta

Reikalavimų inžinerijos procesų tobulinimas IT projektuose

Akademinių sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdamas kitų asmenų autoriaus ar kitų teisių, laikydamasis 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 pašalintas iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Lukas Kusta

Patvirtinta elektroniniu būdu

Kusta, Lukas. Reikalavimų inžinerijos procesų tobulinimas IT projektuose. Magistro baigiamasis projektas / vadovė doc. dr. Ramunė Čiarnienė; Kauno technologijos universitetas, Ekonomikos ir verslo fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Vadyba, Verslas ir viešoji vadyba.

Reikšminiai žodžiai: reikalavimų inžinerija, procesų tobulinimas, IT projektai, programinės įrangos kūrimas, reikalavimų valdymas

Kaunas, 2025. 77 p.

Santrauka

Informacinės technologijos šių dienų pasaulyje įgauna vis didesnę reikšmę dėl suteikiamų skaitmenizavimo, procesų automatizavimo, nuotolinio mokymosi ir daugelio kitų kasdienio gyvenimo ir verslo palengvinimo galimybių. Remiantis statistikos duomenimis (Statista, 2024), investicijos į IT sektorių bėgant metams tendencingai auga, tačiau sėkmingai užbaigiami tik 29% IT projektų, 59% susiduria su problemomis, o 19% projektų yra visiškai nutraukiami. Anot Palumbo ir kt. (2024), netikslus verslo poreikių nustatymas yra dažniausiai pasitaikanti IT projektų nesėkmės priežastis. Vis dėlto mokslinėje literatūroje reikalavimų inžinerijos procesų tobulinimo tema nėra plačiai nagrinėjama, tad šio **tyrimo tikslas** – pasiūlyti sprendimus reikalavimų inžinerijos procesų tobulinimui IT projektuose.

Uždaviniai:

1. Pagrįsti reikalavimų inžinerijos procesų tobulinimo IT projektuose problematiką;
2. Išanalizuoti reikalavimų inžinerijos procesų tobulinimo teorinius sprendimus, taikomus IT projektuose;
3. Pagrįsti reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo metodologiją;
4. Remiantis tyrimo rezultatais pateikti rekomendacijas reikalavimų inžinerijos procesų tobulinimui IT projektuose.

Tyrimo objektas – reikalavimų inžinerijos procesų tobulinimas.

Darbo metodai: mokslinės literatūros analizė ir sintezė, loginė analizė, pusiau struktūruotas interviu, kokybinė turinio analizė.

Pagrindiniai rezultatai: atlikus mokslinės literatūros analizę nustatyta, jog poreikis tobulinti reikalavimų inžinerijos procesus kyla dėl organizacijų siekio įgyti konkurencinį pranašumą, sumažinti klaidų skaičių, padidinti klientų pasitenkinimą, sumažinti sąnaudas ir pagerinti sėkmingų projektų rodiklius. Atlikus mokslinės literatūros analizę buvo identifikuoti šie reikalavimų inžinerijos procesų iššūkiai IT projektuose: dokumentavimo trūkumas, netinkamas prioritetų nustatymas, silpnas pokyčio valdymas, netinkamai aprašyti reikalavimai, kliento neprieinamumas, netinkama komunikacija, netikslus reikalavimų nustatymas ir interpretavimas. Siekiant tobulinti reikalavimų inžinerijos procesus, turi būti naudojamos metrikos, leidžiančios pamatuoti pažangą. Kadangi reikalavimų inžinerija yra daugiau susijusi su kūrybiniu darbu nei su tikslumu, todėl šių procesų metrikos dažniau yra kokybinės nei kiekybinės. Remiantis moksliniais tyrimais nustatyta, jog daugiausiai iššūkių kyla reikalavimų surinkimo etape, todėl pagrindinis dėmesys darbe skiriamas šiai sričiai. Analizuojant mokslinę literatūrą identifikuoti tiek tradiciniai: interviu,

dokumentų analizė, naudojimo atvejai, tiek kūrybiniai reikalavimų surinkimo metodai: smegenų šturmas, dirbtuvės, šešios skrybėlės (angl. *Six Thinking Hats*), įvykių modeliavimas (angl. *Event Storming*), naudotojų patirties žemėlapis (angl. *User Story Mapping*). Mokslinių tyrimų rezultatai rodo, jog kūrybiniai reikalavimų surinkimo metodai padeda praplėsti mąstymo ribas ir skatina į problemą pažvelgti iš skirtingų pusių, tokiu būdu atrandant neišsakytus reikalavimus. Mokslinėje literatūroje taip pat minima, jog tokie metodai dažniausiai nėra taikomi atskirai, todėl empirinio tyrimo metu siekiama išsiaiškinti reikalavimų surinkimo metodų derinimo naudą.

Dauguma empirinio tyrimo metu nustatytų reikalavimų inžinerijos iššūkių IT projektuose sutampa su mokslinėje literatūroje paminėtais iššūkiais. Vis dėlto atliekant empirinį tyrimą buvo atrasti literatūroje nepaminėti iššūkiai: rėmimasis neteisingomis prielaidomis, vizijos nebuvimas, programinės sistemos kūrimas suvokiamas kaip tiriamasis darbas, kurio rezultatai nebūtinai būna aiškūs iš karto. Remiantis atlikto tyrimo rezultatais, pateikiamos šios rekomendacijos: projekto pradžioje išgryninti aiškią viziją ir įvardinti požymius, kurie rodytų problemos išsprendimą; naudoti „5 kodėl“ analizės metodą esminiems klientų poreikiams ir šakninėms problemoms nustatyti. Rekomenduojama taikyti suderintus reikalavimų surinkimo metodus, kurie leidžia pirmuoju etapu sukurti bendrą projekto viziją, o vėliau, įsigilinus į detales, sukurti detalius reikalavimus. Aukšto lygio reikalavimams surinkti rekomenduojami įvykių modeliavimo (angl. *Event Storming*), naudotojų patirties žemėlapio (angl. *User Story Mapping*) ar konteksto žemėlapio (angl. *Context Mapping*) metodai. Sekančiuose reikalavimų surinkimo etapuose rekomenduojama naudoti interviu ir prototipavimo metodus reikalavimų išgryninimui. Remiantis moksline literatūra ir ekspertų rekomendacijomis siūloma nuolat validuoti reikalavimus bei vertinti prioritetus, siekiant sukurti realius verslo poreikius ir kliento lūkesčius atitinkančius produktus.

Kusta, Lukas. Improvement of Requirements Engineering Processes in IT Projects. Master's Final Degree Project / supervisor assoc. prof. dr. Ramunė Čiarnienė; School of Economics and Business, Kaunas University of Technology.

Study field and area (study field group): Management, Business and Public Management.

Keywords: requirements engineering, process improvement, IT projects, software development, requirements management

Kaunas, 2025. 77.

Summary

Information technologies are gaining increasing importance in today's world due to created opportunities for digitalization, process automation, remote learning and many other ways to ease everyday life and business development. According to statistics, investments in the IT sector have been growing over the years, however, only 29% of IT projects are successfully completed, 59% encounter problems, and 19% of all projects are completely terminated. According to statistical studies, inaccurate elicitation of business needs is the leading cause of IT project failure. However, the topic of requirements engineering process improvement is not widely studied in the scientific literature, so the purpose of this study is to propose solutions for the improvement of requirements engineering processes in IT projects.

Tasks:

1. Justify the problem of requirements engineering process improvement in IT projects.
2. Analyze the theoretical solutions for requirements engineering process improvement applied in IT projects.
3. Justify the research methodology for the improvement of requirements engineering processes in IT projects.
4. Based on the results of the study, propose recommendations for improving requirements engineering processes in IT projects.

The object of the study is the improvement of requirements engineering processes.

Methods: analysis of scientific literature, semi-structured interviews, qualitative content analysis.

Main results: during the literature analysis, it was determined that the need to improve requirements engineering processes arises from the need of organizations to gain a competitive advantage, reduce the number of errors, increase customer satisfaction, reduce costs and increase the indicators of successful projects. The following challenges of requirements engineering processes in IT projects were identified: lack of documentation, improper prioritization, poor change management, poorly described requirements, customer inaccessibility, inadequate communication, inaccurate elicitation and interpretation of requirements. In order to improve requirements engineering processes, metrics must be established that allow measuring progress. Since requirements engineering is more related to creative work, the metrics of these processes are more often qualitative than quantitative. It was found that the requirements elicitation poses the most challenges, so the main attention was paid to this area. The literature analyzes both traditional methods: interviews, document analysis, use cases, and creative requirements elicitation methods: brainstorming, workshops, “Six Thinking Hats”, “Event Storming”, “User Story Mapping”. The

results of scientific researches state that creative requirements gathering methods help to expand the boundaries of thinking and encourage looking at the problem from different angles, thus discovering hidden requirements. It is also mentioned that such methods are usually not applied separately, so the study aims to find out the benefits of combining requirements elicitation methods.

The challenges of requirements engineering in IT projects identified after conducting an empirical study largely coincide with the challenges found in scientific literature. The following challenges, not mentioned in the literature, were discovered: relying on incorrect assumptions, lack of vision, and the development of a software system is perceived as research work, the results of which are not necessarily clear immediately. Based on the results of the study, the following recommendations are presented: at the beginning of the project, a clear vision and definition of the problem to be solved should be clarified. It is recommended to apply the "5 whys" analysis method to identify essential customer needs and determine root problems. It is recommended to apply combined requirements elicitation methods, which allow creating a general picture and vision of the project in the first stage, and later, after delving into the details, to create detailed requirements. "Event Storming", "User Story Mapping" or "Context Mapping" methods are recommended for eliciting high-level requirements. In the following stages, it is recommended to use interview and prototyping methods to clarify requirements. Both in the literature and in expert recommendations, a proposal to constantly validate requirements and assess priorities is observed in order to create products that meet real business needs and customer expectations.

Turinys

Paveikslų sąrašas.....	8
Lentelių sąrašas.....	9
Įvadas.....	10
1. Reikalavimų inžinerijos procesų tobulinimo IT projektuose problemos analizė.....	11
1.1. IT projektų svarba ir iššūkiai.....	11
1.2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose svarba.....	15
2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose teoriniai sprendimai.....	18
2.1. Reikalavimų inžinerijos samprata ir svarba.....	18
2.2. Reikalavimų inžinerijos procesai.....	21
2.3. Procesų analizės ir tobulinimo metodai.....	24
2.3.1. Pagrindiniai veiklos rodikliai.....	24
2.3.2. Procesų vizualizavimas.....	26
2.3.3. Ishikawa diagrama.....	28
2.3.4. „5 kodėl“ metodas.....	30
2.3.5. SCAMPER metodas.....	32
2.4. Reikalavimų surinkimo proceso tobulinimas.....	33
2.4.1. Tradiciniai reikalavimų surinkimo metodai.....	34
2.4.2. Kūrybiniai reikalavimų surinkimo metodai.....	35
2.5. Reikalavimų inžinerijos procesų tobulinimo IT projektuose teorinis modelis.....	39
3. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo metodologija.....	41
4. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo rezultatai ir diskusija.....	44
4.1. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo duomenų analizė.....	44
4.1.1 Reikalavimų inžinerijos iššūkiai IT projektuose.....	44
4.1.2 Reikalavimų inžinerijos procesų vertinimo metrikos IT projektuose.....	48
4.1.3 Reikalavimų inžinerijos procesų tobulinimo metodai IT projektuose.....	50
4.1.4 Reikalavimų surinkimo metodai IT projektuose.....	53
4.2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo diskusija.....	64
4.3. Reikalavimų inžinerijos procesų tobulinimo IT projektuose rekomendacijos.....	67
Išvados.....	70
Literatūros sąrašas.....	72
Informacijos šaltinių sąrašas.....	77
Priedai.....	78
1 priedas. Pusiaus struktūruoto interviu klausimynas.....	78

Paveikslų sąrašas

1 pav. IT išlaidos, apimančios programinę įrangą, įrenginius, paslaugas, duomenų centrus, komunikacijas (Statista, 2024).....	11
2 pav. Terminus ar biudžetą viršijusių IT projektų statistika (Palumbo et al., 2024).....	13
3 pav. IT projektų biudžeto ir terminų viršijimą lemiančios priežastys (Palumbo et al., 2024).....	14
4 pav. Reikalavimų inžinerijos procesų diagrama (Crowder & Hoff, 2022).....	22
5 pav. Kiekybinių KPI klasifikavimas (Gadatsch, 2023).....	25
6 pav. Užsakymo vykdymo proceso pavyzdys, naudojant BPMN (Pergl et al., 2019).....	28
7 pav. Ishikawa diagramos šablonas (Lawless, 2024).....	29
8 pav. „5 kodėl“ metodo taikymo iliustracija (PMWorld 360, 2024).....	31
9 pav. Reikalavimų inžinerijos procesų tobulinimo IT projektuose konceptualus teorinis modelis.....	40
10 pav. Reikalavimų inžinerijos iššūkių IT projektuose vizualinis modelis.....	44
11 pav. Pagrindinių reikalavimų inžinerijos procesų rodiklių IT projektuose vizualinis modelis.....	48
12 pav. Gerųjų reikalavimų inžinerijos praktikų IT projektuose vizualinis modelis.....	51
13 pav. Ekspertų taikomų reikalavimų rinkimo metodų IT projektuose vizualinis modelis.....	53
14 pav. Interviu privalumų vizualinis modelis.....	55
15 pav. Interviu trūkumų vizualinis modelis.....	56
16 pav. Dirbtuvių privalumų vizualinis modelis.....	57
17 pav. Dirbtuvių trūkumų vizualinis modelis.....	58
18 pav. Svarbiausių reikalavimų surinkimo aspektų vizualinis modelis.....	60

Lentelių sąrašas

1 lentelė. Proceso apibrėžimai pagal skirtingus autorius (Hilmer, 2016).....	21
2 lentelė. Interviu respondentų duomenys.....	42
3 lentelė. Reikalavimų inžinerijos iššūkių IT projektuose subkategorijos ir pagrindžiančios citatos.....	45
4 lentelė. Pagrindinių reikalavimų inžinerijos procesų rodiklių IT projektuose subkategorijos ir pagrindžiančios citatos.....	48
5 lentelė. Šakninės priežasties analizės taikymą IT projektų reikalavimų inžinerijoje pagrindžiančios citatos.....	50
6 lentelė. Gerųjų reikalavimų inžinerijos praktikų IT projektuose subkategorijos ir pagrindžiančios citatos.....	51
7 lentelė. Reikalavimų surinkimo metodų IT projektuose subkategorijos ir pagrindžiančios citatos.....	53
8 lentelė. Interviu privalumų subkategorijos ir pagrindžiančios citatos.....	55
9 lentelė. Interviu trūkumų subkategorijos ir pagrindžiančios citatos.....	56
10 lentelė. Dirbtuvių privalumų subkategorijos ir pagrindžiančios citatos.....	57
11 lentelė. Dirbtuvių trūkumų subkategorijos ir pagrindžiančios citatos.....	59
12 lentelė. Svarbiausių reikalavimų surinkimo aspektų subkategorijos ir jas pagrindžiančios citatos.....	60
13 lentelė. Literatūroje ir empiriniame tyrime nustatytų reikalavimų inžinerijos iššūkių palyginimas....	64
14 lentelė. Reikalavimų inžinerijos procesų tobulinimo IT projektuose rekomendacijos.....	67

Ivadas

Reikalavimų inžinerijos procesai informacinių technologijų (toliau – IT) projektuose daro tiesioginę įtaką projekto sėkmei, išteklių panaudojimo efektyvumui ir suinteresuotųjų šalių pasitenkinimui (Palumbo et al., 2024). Anot Rasheed'o ir kt. (2021), reikalavimų inžinerijos procesų valdymas daro didžiausią įtaką sėkmingam projekto įgyvendinimui. Daugeliu atvejų reikalavimų inžinerijos procesuose pasitaiko klaidų, todėl šią sritį nagrinėjantys mokslininkai pabrėžia procesų sudėtingumą. Vis dėlto nors mokslinėje literatūroje aptariamos reikalavimų inžinerijos problemos ir iššūkiai, pastebimas reikalavimų inžinerijos procesų tobulinimą IT projektuose nagrinėjančių tyrimų trūkumas. Tokio pobūdžio mokslinių tyrimų autoriai savo darbuose neretai nurodo reikalavimų inžinerijos procesų tobulinimą kaip ateities tyrimų kryptį.

Temos aktualumas. Reikalavimų inžinerijos procesų tobulinimo temos aktualumas grindžiamas tiek mokslinių tyrimų rekomendacijomis, tiek autoriaus praktine patirtimi „Agile“ projektuose. Reikalavimų inžinerijos procesų klaidos daro didelę įtaką projekto sėkmės rodikliams, blogina santykius su klientais, trukdo pasiekti iškeltus projekto tikslus. Taigi, reikalavimų inžinerijos procesų tobulinimo tyrimas būtų naudingas IT projektus vykdančioms kompanijoms, siekiančioms patobulinti dabartinius procesus, padidinti konkurencinį pranašumą ir sumažinti laiko sąnaudas atliekant pakeitimus. Šis tyrimas taip pat prisideda prie mokslinių tyrimų spragos užpildymo, išsiaiškinant ne tik IT projektų reikalavimų inžinerijos iššūkius, tačiau taip pat pateikiant mokslinę literatūrą ir ekspertų gerosiomis praktikomis pagrįstas rekomendacijas.

Tyrimo objektas - reikalavimų inžinerijos procesų tobulinimas.

Problema: kaip patobulinti reikalavimų inžinerijos procesus IT projektuose?

Tyrimo tikslas: pasiūlyti sprendimus reikalavimų inžinerijos procesų tobulinimui IT projektuose.

Uždaviniai:

1. Pagrįsti reikalavimų inžinerijos procesų tobulinimo IT projektuose problematiką;
2. Išanalizuoti reikalavimų inžinerijos procesų tobulinimo teorinius sprendimus, taikomus IT projektuose;
3. Pagrįsti reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo metodologiją;
4. Remiantis tyrimo rezultatais pateikti rekomendacijas reikalavimų inžinerijos procesų tobulinimui IT projektuose.

Darbo metodai: analizuojama ir sisteminama mokslinė literatūra, atliekamas pusiau struktūruotas interviu ir kokybinė turinio analizė.

Darbo struktūra: magistro baigiamasis darbas sudarytas iš šių pagrindinių dalių: įvado, mokslinės literatūros analizės, empirinio tyrimo metodologijos pagrindimo, empirinio tyrimo rezultatų analizės, rekomendacijų ir išvadų.

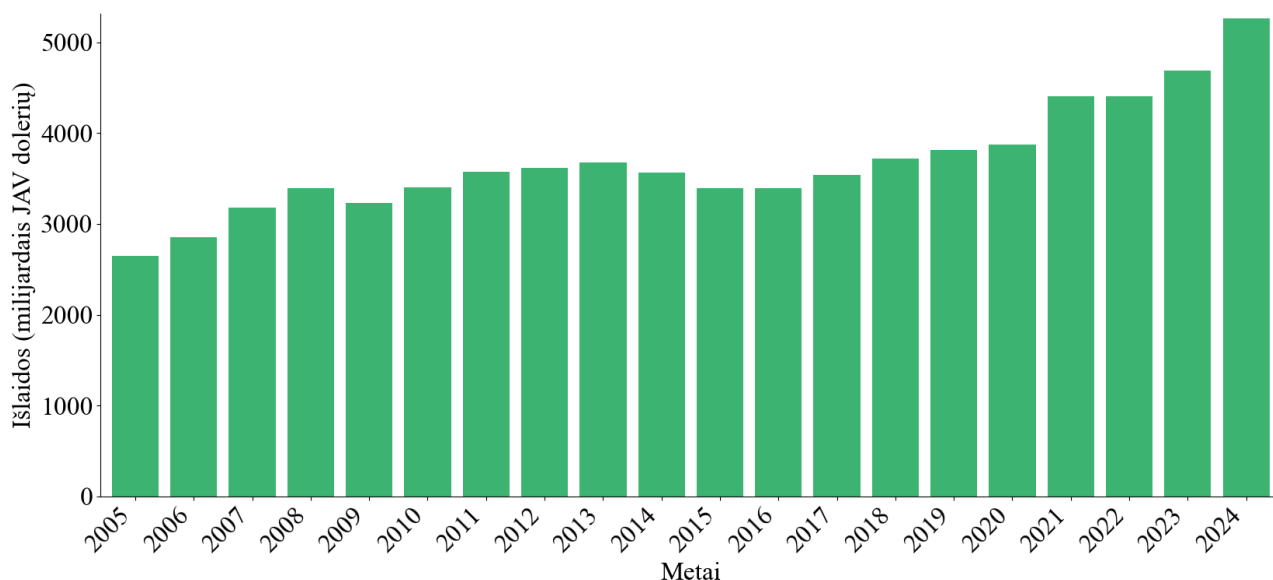
1. Reikalavimų inžinerijos procesų tobulinimo IT projektuose problemos analizė

1.1. IT projektų svarba ir iššūkiai

Informacinių technologijų sritis šių dienų pasaulyje įgauna vis didesnę reikšmę dėl sukuriamos didelės pridėtinės vertės įvairiuose sektoriuose. IT suteikiamos galimybės yra plačiai pritaikomos nuo gamybos linijų optimizavimo iki animacinių filmų kūrimo ir ugdymo proceso tobulinimo (Vululleh, 2021). Analizuojant mokslinę literatūrą nustatyta, kad informacinių technologijų daromą įtaką galima išskaidyti į šias pagrindines grupes:

- **efektyvumo ir produktyvumo didinimas.** IT priemonės plačiai naudojamos įvairių pasikartojančių užduočių automatizavimui. IT sprendimai sutrumpina procesų atlikimo laiką, optimizuoja medžiagų sąnaudas – didina produktyvumą ir taupo išteklius įvairiose pramonės šakose (Mohylna & Makarova, 2023; Panduro, 2023);
- **sprendimų priėmimas ir valdymas.** Duomenų analizės įrankiai suteikia galimybę priimti geriau apgalvotus, pamatuotus ir duomenimis pagrįstus sprendimus (Mohylna & Makarova, 2023; Panduro, 2023);
- **išlaidų mažinimas.** Šiuolaikinės IT priemonės suteikia įrankius, leidžiančius efektyvinti gamybos linijas ir optimizuoti išteklių panaudojimą, sumažinant rankinio darbo poreikį, tokiu būdu sumažinant įmonės patiriamas išlaidas (Küsbeci, 2023);
- **konkurencinis pranašumas.** IT priemonės suteikia organizacijoms konkurencinį pranašumą ir galimybę įgyvendinti inovatyvius sprendimus, kurie yra ypač svarbūs greitai besikeičiančiose rinkose (Parkhatova & Erzhanova, 2022);
- **švietimas ir edukacija.** IT priemonės sukuria geresnę mokymosi patirtį, stipriai išplečia prieinamą informaciją, sukuria nuotolinio mokymosi galimybes ir padeda įgyti šiandienos visuomenėje būtinus skaitmeninius įgūdžius (Levkivska & Levkivskyi, 2024).

Informacinių technologijų svarba akivaizdžiai matoma žvelgiant į pasaulines išlaidas informacinėms technologijoms (1 pav.). Šios išlaidos apima programinę įrangą, įrenginius, paslaugas, duomenų centrus, komunikacijas.



1 pav. IT išlaidos, apimančios programinę įrangą, įrenginius, paslaugas, duomenų centrus, komunikacijas (Statista, 2024)

Iš 1 paveiksle pateikto grafiko galima matyti, jog išlaidos informacinėms technologijoms tendencingai didėja bėgant laikui. Pavyzdžiui, lyginant 2023 ir 2024 metus, pastebimas 12% metinis išlaidų augimas nuo 4 687 mlrd. iki 5 260 mlrd. JAV dolerių, o per 10 metų išlaidos informacinėms technologijoms išaugo beveik 50%. Ši statistika pagrindžia, jog informacinės technologijos įgauna vis didesnę svarbą.

Nors informacinių technologijų sritis sulaukia vis daugiau susidomėjimo ir investicijų, tačiau IT projektų valdymas vis dar išlieka sudėtingas. Didelė dalis IT projektų viršija biudžetą ar grafiką. Žvelgiant į istorinius duomenis, randami atvejai, kai nesėkmingi IT projektai privedė įmones prie bankroto. Tokio atvejo pavyzdys – kompanija „KMart“, kurią sužlugdė nepavykęs 1,2 mlrd. JAV dolerių vertės informacinių sistemų modernizavimo ir tobulinimo projektas (Hannah, 2022). Projektas gali būti vadinamas nepasisekusiu, kai nepasiekiami iškelti tikslai arba nepatenkinami sėkmės kriterijai, tačiau vieno teisingo apibrėžimo literatūroje greičiausiai nėra. Dažnai nesėkmės suvokimas priklauso nuo suinteresuotųjų šalių, vadovų ir turimos informacijos. Sutarimą, jog projektas buvo nesėkmingas, dažniausiai pasiekia pagrindiniai projekto sprendimų priėmėjai. Kai toks susitarimas pasiekiamas, projektas laikomas žlugusiu ir pradedama nesėkmę lėmusių priežasčių analizė (Money et al., 2021).

Nesėkmingi programinės įrangos projektai mažina įmonių konkurencinį pranašumą, nesėkmės sukelia finansinius nuostolius, suinteresuotųjų šalių nepasitenkinimą bei mažina arba visiškai sunaikina planuotą projekto naudą. Nepaisant programinės įrangos kūrimo projektų svarbos ir didelių investicijų, projektų nesėkmės išlieka dažnos. Sėkmingų projektų dalis sudaro vos 29%, 59% susiduria su problemomis, o 19% projektų yra visiškai nutraukiami. Didelių projektų sėkmės rodikliai dar žemesni – sėkmingai užbaigiama vos 10% (Nizam, 2022).

Anot Money ir kt. (2021), Nizam'o (2022), Rasheed'o ir kt. (2021) atliktų tyrimų, pagrindinės IT projektų nesėkmių priežastys gali būti suskirstytos į šias grupes:

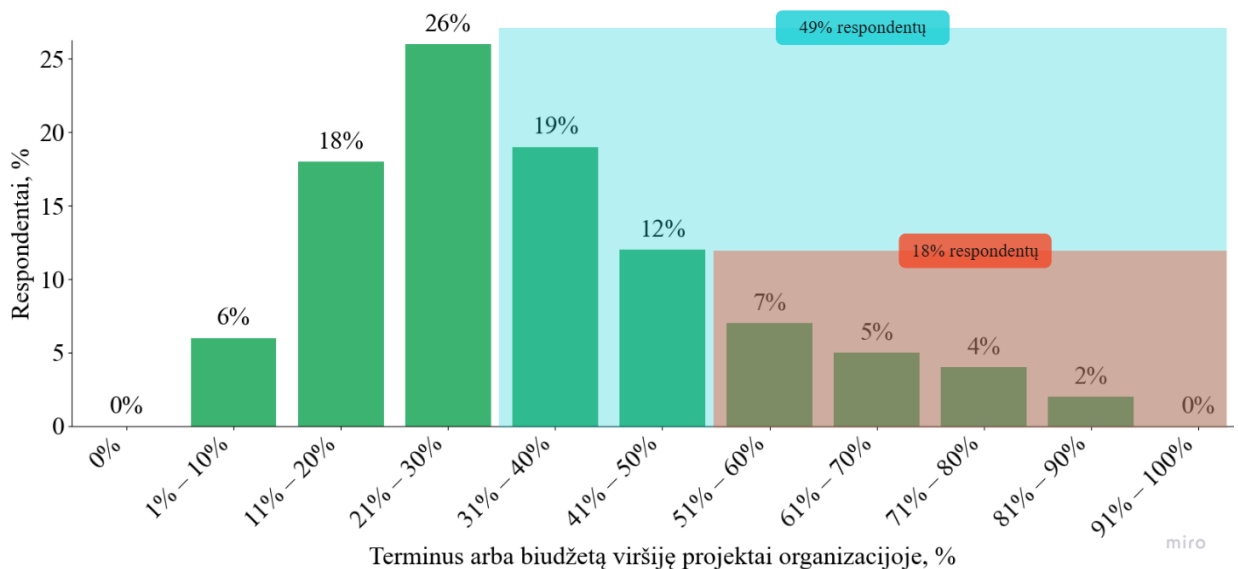
- **netinkamas reikalavimų valdymas:**

- dviprasmiški reikalavimai. Skirtingi suinteresuotieji asmenys reikalavimus interpretuoja savaip ir dėl to kyla nesusipratimai bei klaidos. Anot Jarzębowicz'iaus ir Poniatowska'os (2020) tyrimo, dažniausiai pasitaikančios klaidos reikalavimuose buvo susijusios su dviprasmiškumu, neišsamumu ir neįmanomu testavimu. Dviprasmiški reikalavimai sudarė didžiausią dalį – 22%, ypač projektuose su pasyviais reikalavimų rinkimo metodais;
- pertekliniai reikalavimai. Jie apsunkina valdymo procesą ir yra kaip papildomas informacinis triukšmas, keliantis chaosą;
- nerealistiški reikalavimai – tokie reikalavimai, kurių neįmanoma įgyvendinti dėl įvairių apribojimų, pavyzdžiui laiko spaudimo arba technologinių pasirinkimų;
- per mažas dėmesys reikalavimų gryninimui ir neišgryninti reikalavimai. Dažnai sutinkamas požiūris, jog reikalavimų peržiūra (iteracinis procesas) yra per daug laiko reikalaujanti veikla. Tai lemia esminių detalių apibrėžimo trūkumą, paliekama per daug vietos interpretacijoms ir klaidoms;
- netiksli reikalavimų interpretacija. Projekto komanda ir klientas tuos pačius reikalavimus interpretuoja skirtingai, dėl to atsiranda klaidos;
- nesusekama reikalavimų kaita. Nuolat vykstantys ir nevaldomi reikalavimų pokyčiai projekto eigoje atima daug laiko ir kelia riziką projektui viršyti biudžetą ir grafiką;

- **netinkama projekto komandos sandara.** Komandos nariai neturi tinkamų įgūdžių, reikalingų projekto vykdymui. Tai lemia procesų neefektyvumą ir prastą rezultatą;

- **nepakankami resursai.** Projekto įgyvendinimui skiriama per mažai darbuotojų arba trūksta tinkamos infrastruktūros, įrankių;
- **prastas planavimas ir neįgyvendinami terminai.** Dėl prasto planavimo netinkamai nustatoma preliminarai projekto apimtis ir pateikiamas per daug optimistinis terminas projekto įgyvendinimui. Užduotys priskiriamos komandos nariams ne pagal jų gebėjimus, darbų eiga nėra optimizuota, skubama pradėti darbus be išsamaus planavimo;
- **nenumatytos rizikos.** Nepavyksta identifikuoti potencialių rizikų ankstyvame projekto etape. Nesant rizikos valdymo plano, projektas gali būti sustabdytas dėl netikėtai atsiradusių kliūčių;
- **netinkamai naudojamos technologijos.** Pasirenkamos technologijos, kurios netinka projekto poreikiams arba yra pasenusios. Jeigu projekto tikslas papildyti esamą sistemos funkcionalumą, gali atsitikti taip, jog per vėlai sužinoma, kad anksčiau naudotos technologijos nesuderinamos su naujai pasirinktomis.

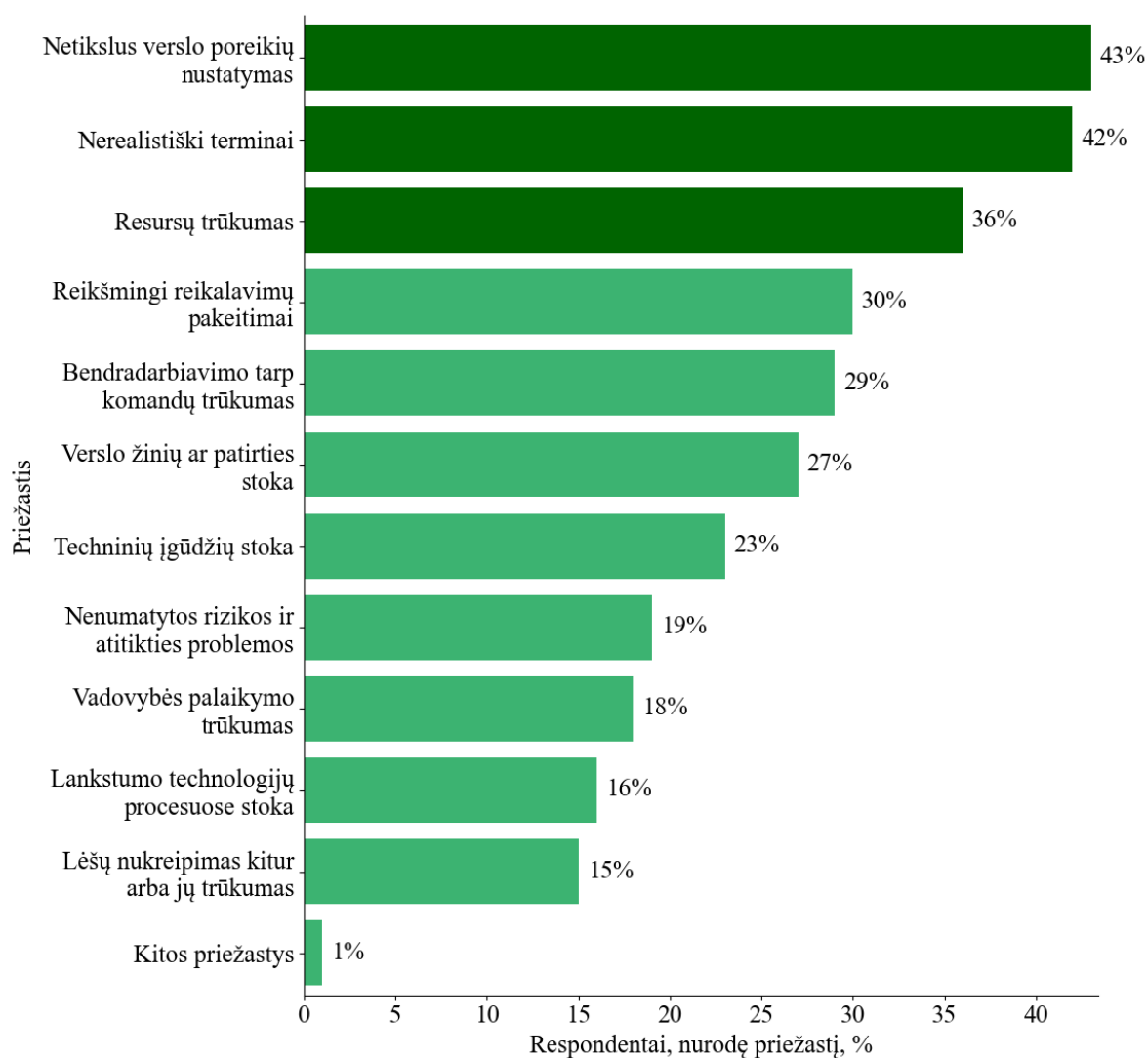
Viena iš pirmaujančių pasaulinių verslo konsultavimo įmonių („Boston Consulting Group“) – atliko 403-jų globalių verslų C lygio vadovų apklausą su tikslu išsiaiškinti, kodėl IT projektai dažnai vėluoja ir viršija biudžetą. Kompanijos atliktame tyrime buvo apklausti 25 skirtingų industrijų kompanijų vadovai, daugiausiai dėmesio skiriant naujiems sprendimams ir programoms, kurios yra pritaikytos specifiniams verslo poreikiams. Į tyrimo apimtį nebuvo įtrauktos didelio masto išteklių planavimo sistemos, migracijos į debesijos sprendimus ir kiti dideli projektai, kurie gali būti veikiami daugelio išorinių veiksnių (Palumbo et al., 2024). 2 paveiksle pateikiamas grafikas, iliustruojantis, kokia dalis programinės įrangos kūrimo projektų patiria vėlavimus arba biudžeto viršijimus. Horizontali ašis nurodo vėluojančių ar biudžetą viršijančių projektų procentinę dalį organizacijos IT projektų portfelyje. Vertikali ašis rodo respondentų dalį, kurie nurodė, kad atitinkama jų organizacijos projektų dalis patyrė vėlavimą ar biudžeto viršijimą.



2 pav. Terminus ar biudžetą viršijusių IT projektų statistika (Palumbo et al., 2024)

Iš 2 paveikslo matyti, kad beveik pusė respondentų teigė, jog daugiau nei 30% IT projektų jų organizacijose susiduria su vėlavimu arba biudžeto viršijimu. Beveik 20% respondentų teigė, jog mažiau nei patenkinami rezultatai yra gaunami daugiau nei pusėje IT projektų (Palumbo et al., 2024).

„Boston Consulting Group“ atliktoje apklausoje vadovai taip pat buvo paprašyti nurodyti priežastis, dėl kurių IT projektai dažniausiai netelpa į laiko ir biudžeto rėmus. Rezultatai pateikiami 3 paveiksle



3 pav. IT projektų biudžeto ir terminų viršijimą lemiančios priežastys (Palumbo et al., 2024)

Iš 3 paveiksle pavaizduotų Palumbo ir kt. (2024) tyrimo rezultatų galima išskirti tris svarbiausias priežastis, dėl kurių IT projektai netelpa į numatytus biudžeto ir laiko apribojimus:

- **netikslus verslo poreikių nustatymas.** Trūksta bendro supratimo tarp projekto įgyvendinimo komandos ir verslo dėl konkrečių kuriamos sistemos tikslų, kas lemia nesusipratimus ir neefektyvų projekto įgyvendinimą;
- **nerealūs terminai.** Neatsižvelgiant į realias galimybes ir resursus nustatomi per trumpi arba neįgyvendinami projekto terminai;
- **nepakankami finansiniai ir žmogiškieji ištekliai.** Projektui skiriama per mažai lėšų ir darbuotojų, kas trukdo sėkmingai užbaigti projektą laiku neviršijant biudžeto.

„Boston Consulting Group“ apklausos rezultatai patvirtina anksčiau aptartų Nizam'o (2022), Money ir kt. (2021) tyrimų rezultatus, iš kurių matyti, jog netikslus verslo poreikių nustatymas, nerealistiški terminai ir resursų trūkumas patenka į pagrindinių projekto nesėkmės sukeliančių problemų sąrašą. Apibendrinant mokslinės literatūros analizės rezultatus galima daryti išvadą, jog netikslus verslo poreikių nustatymas IT projektuose yra viena iš pagrindinių problemų, lemiančių projekto nesėkmę. Todėl įgyvendinant IT projektus labai svarus vaidmuo tenka reikalavimų inžinerijos procesams.

1.2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose svarba

Reikalavimų inžinerija yra programinės įrangos kūrimo ciklo dalis, nagrinėjanti reikalavimų surinkimą, dokumentavimą ir valdymą. Reikalavimų inžinerija apibrėžiama kaip sisteminis, iteracinis procesas, kurio metu analizuojama problema, įvairiomis formomis dokumentuojami analizės rezultatai ir nuolat tikrinamas surinktos informacijos tikslumas. Reikalavimų inžinerijos procesai gali skirtis priklausomai nuo projekto, tačiau išskiriamos šios bendros veiklos pagal Chirumamilla'ą ir kt. (2018):

- **reikalavimų išgryninimas.** Ši veikla apima suinteresuotųjų šalių poreikių išaiškinimą, išsamų lūkesčių supratimą, konteksto analizę ir apribojimų, pavyzdžiui, nefunkcinių reikalavimų, nustatymą bei vertinimą. Ši veikla taip pat apima konkrečios sistemos veikimą atspindinčių mentalinių modelių kūrimą;
- **reikalavimų analizė.** Šios veiklos tikslas yra užtikrinti, kad reikalavimai būtų aiškūs, nuoseklūs ir išsamūs, nepaliekant vietos dviprasmybėms. Jei procese iškyla prieštaravimų tarp skirtingų reikalavimų, vykdomas derybų etapas, kuriame siekiama suderinti visų suinteresuotųjų šalių interesus. Ši veikla glaudžiai susijusi ir dažnai vyksta kartu su reikalavimų išgryninimu, jų dokumentavimu ir validavimu;
- **reikalavimų dokumentavimas.** Ši veikla analizuoja reikalavimų išgryninimo metu surinktą informaciją ir neformaliai išdėstytus reikalavimus transformuoja į reikalavimų specifikacijos modelį. Šis modelis kuriamas iteracijomis, nuolat jį tobulinant, grįžtant į reikalavimų išgryninimo etapą ir pereinant į validavimo etapą;
- **reikalavimų validacija.** Ši veikla užtikrina, kad sukurtas reikalavimų modelis atitinka suinteresuotųjų šalių poreikius;
- **reikalavimų valdymas.** Ši veikla apima reikalavimų pokyčių valdymą po išgryninimo ir dokumentavimo. Reikalavimų valdymas atliekamas lygiagrečiai su reikalavimų analize, siekiant išvengti prieštaraujančių ar dviprasmiškų reikalavimų.

Reikalavimų inžinerija yra viena iš svarbiausių IT projekto įgyvendinimo dalių, kuri sudaro tvirtą pagrindą visam projekto gyvavimo ciklui ir tiesiogiai lemia projekto sėkmę arba nesėkmę (Money et al., 2021; Nizam, 2022; Palumbo et al. 2024). Reikalavimų inžinerijos procesų tikslas yra užtikrinti bendro supratimo sukūrimą tarp visų suinteresuotų šalių apie kuriamą produktą, taikomus sėkmės kriterijus. Tikslus ir aiškus reikalavimų pateikimas stipriai sumažina klaidingų interpretacijų ir nesusipratimų riziką. Kaip teigiama mokslinėje literatūroje, aiškūs reikalavimai sudaro tinkamas prielaidas kurti kliento lūkesčius atitinkantį ir verslo poreikius atliepiantį produktą nuo pat projekto pradžios. Vis dėlto jeigu įmonė neturi pakankamai gerai ištobulintų reikalavimų inžinerijos procesų, labai tikėtina, jog projekto komanda susidurs su netiksliu planavimu, pertekliniu darbu ir neefektyviu resursų paskirstymu (Hegade, 2020). Siekiant sukurti aiškų, struktūruotą projekto įgyvendinimo planą, sklandūs reikalavimų inžinerijos procesai yra būtina sąlyga. Aiškiai struktūruota reikalavimų dokumentacija leidžia greitai identifikuoti galimus pokyčius ir juos efektyviai valdyti, neviršijant nustatyto projekto biudžeto ir terminų (Hegade, 2020).

Dėl prastai išpildytų reikalavimų inžinerijos procesų trūksta kuriamos sistemos reikalavimų išgryninimo ir nuoseklumo. Prastai išgryninti reikalavimai dažniausiai sukuria nesusipratimus tarp projekto suinteresuotųjų šalių ir daro tiesioginę neigiamą įtaką projekto sėkmės rodikliams. Ne iki galo išgrynintų reikalavimų rezultatas – viršijamas biudžetas ir terminas, kadangi programuotojams užduoties įgyvendinimo metu tenka analizuoti dviprasmiškus reikalavimus (Palumbo et al., 2024). Be to, dėl netinkamų reikalavimų inžinerijos procesų išpildymo, galutiniame produkte gali atsirasti daugiau klaidų ir problemų, kurioms ištaisyti bus eikvojamas komandos laikas bei kils rizika viršyti projekto biudžetą.

Kai nepasiekiamas bendras supratimas tarp kliento ir projekto komandos apie kuriamą produktą, gerokai padidėja rizika neatitikti kliento lūkesčių (Jarzębowicz & Poniatońska, 2020). Apibendrinant galima teigti, kad teisingas reikalavimų inžinerijos procesų išpildymas leidžia įgyvendinti projektą nustatytu terminu ir su paskirtu biudžetu, taip pat sudaro palankias sąlygas sukurti kliento poreikius atitinkantį produktą. Vadinasi, reikalavimų inžinerija yra kertinis IT projektų akmuo, ant kurio stovi visų kitų projekto etapų sėkmė (Palumbo et al., 2024). Papildant šį teiginį, verta paminėti Jarzębowicz'iaus ir Poniatońska'os (2020) tyrimą, kuriame mokslininkai nustatė, jog klaidos reikalavimų inžinerijos procesuose sukuria apie 30% papildomo darbo. Apibendrinant galima teigti, jog siekiant sumažinti projekto sąnaudas ir išvengti papildomo darbo, svarbu skirti pakankamą dėmesį reikalavimų inžinerijos procesų tobulinimui.

Anot Hoy ir Xu (2023) atliktos sisteminės 78 mokslinių straipsnių analizės, „Agile“ projektuose reikalavimų inžinerijos iššūkius galima suskirstyti į 11 grupių:

1. **Ignoruojami kokybiniai reikalavimai.** Kokybiniai reikalavimai apima: greitaveiką, saugumą, naudojimo patogumą, sistemos plėtimo galimybes, sistemos patikimumą, lengvą palaikymą (Behutiye et al., 2020). Ankstyvoje projekto stadijoje kokybiniai reikalavimai dažniausiai ignoruojami, nes pagrindinis dėmesys skiriamas funkciniais reikalavimams. Bandant integruoti kokybinius reikalavimus projekto eigoje, dažnai aptinkami trūkumai, sukuriantys daug papildomo darbo;
2. **Dokumentavimo trūkumas.** „Agile“ metodologija yra labiau orientuota į veikiančią kodą negu į reikalavimų dokumentaciją (Agile manifesto, 2001), todėl dokumentacija dažnai yra nepilna, netiksli arba netgi visiškai neegzistuoja. Žinių dalijimasis komandoje, pagrįstas asmeniniais susitarimais ir reikalavimų saugojimu atmintyje, ilgainiui praranda vertę, ypač keičiantis projekto komandai. Nepakankama arba prasta dokumentacija apsunkina komunikaciją ir testavimą;
3. **Netinkami prioritetai.** Kai yra daug reikalavimų, prioritetų nustatymas dažnai nėra objektyvus. Tiesioginę vertę verslui kuriantys funkciniai reikalavimai dažnai gauna aukštesnį prioritetą nei kokybiniai reikalavimai. Pavyzdžiui, saugumas arba sistemos praplėtimo galybės nekuria akivaizdžios vertės, tačiau yra ypatingai svarbūs ilgoje perspektyvoje. Taip pat pastebėta, kad tam tikrais atvejais klientai tiksliai nenustato reikalavimų prioritetų;
4. **Prastas reikalavimų pokyčių valdymas.** Verslo poreikiai nuolat kinta, todėl reikalavimus reikia nuolat peržiūrėti ir iš naujo nustatyti prioritetus. Tai sukelia nestabilumą, reikalauja dokumentų atnaujinimo, didina projekto kaštus. Jeigu reikalavimų pokyčiai prieš sprintą nėra apribotai, labai tikėtina, jog bus nukrypta nuo plano (Palumbo et al., 2024);
5. **Prastai aprašyti reikalavimai.** Reikalavimai būna nerealistiški, dviprasmiški, neišsamūs, aprašyti su trūkumais (Palumbo et al., 2024). Dažnai jie yra per daug detalizuoti arba atvirkščiai – per daug bendri, todėl kyla skirtingos interpretacijos tarp komandos narių. Naudotojo istorijos (angl. *User Stories*) didelį dėmesį kreipia į funkcinius reikalavimus, kokybinius reikalavimus paliekant nuošalyje;
6. **Netikslus darbo sąnaudų įvertinimas.** Reikalavimų pokyčiai vėlesniuose projekto etapuose reikšmingai keičia darbo sąnaudų įvertinimus. Tai dažnai pasitaiko, kai trūksta istorinių duomenų ar komandos ekspertiškumo, ypač didelės apimties naudotojo istorijose (angl. *User Stories*);
7. **Kliento nepasiekiamumas.** Dažnu atveju klientas yra sunkiai pasiekiamas ir ypatingai mažai įsitraukia į produkto kūrimo procesą. Šis iššūkis dar stipriau išryškėja, kai projektas rengiamas su užsienio partneriais. Tai pačiai problemai priskiriami klientai, kurie neturi įgaliojimų priimti sprendimus, pavyzdžiui jeigu kontaktiniai asmenys nėra tie patys asmenys, kurie moka pinigus

arba naudoja kuriamą produktą. Tai lemia nepakankamą reikalavimų supratimą, prastos kokybės sprendimus, labai apriboja galimybę gauti grįžtamąjį ryšį iš klientų, atlikti patikslinimus. Atsakomybė už reikalavimų kokybę tampa didele problema, kai atsiranda daug tarpininkų tarp tikrojo sistemos naudotojo ir projekto komandos;

8. **Kliento žinių trūkumas.** Kai klientai neturi pakankamai techninių žinių ar nesupranta „Agile“ metodologijos privalumų, sunku iš jų gauti kokybišką grįžtamąjį ryšį. Be to, ribotas kliento žinių lygis apie domeną sunkina kokybiškų reikalavimų formulavimą;
9. **Netinkama architektūra.** Ankstyvoje projekto stadijoje priimti architektūriniai sprendimai gali tapti nebeaktualūs ar netinkami, atsirandant naujiems reikalavimams. Dėl to atsiranda poreikis perdarinėti sistemos architektūrą, kad būtų išpildyti kliento poreikiai;
10. **Netinkami komunikacijos metodai.** Komunikacija su suinteresuotosiomis šalimis gali būti sudėtinga. Kai kurie metodai yra per brangūs arba neefektyvūs, o tiesioginiai susitikimai pripažįstami efektyviausiais, tuo tarpu momentinis susirašinėjimas – prasčiausias komunikacijos būdas;
11. **Reikalavimų dokumentacijos palaikymas.** Komandos, dirbančios prie kelių projektų vienu metu, dažnai susiduria su sunkumais prižiūrėti ir atnaujinant reikalavimų dokumentaciją.

Apibendrinant galima teigti, jog IT projektų reikalavimų inžinerija yra sudėtingas procesas, pasižymintis dideliu klaidų skaičiumi dėl aukščiau minėtų įvairių reikalavimų inžinerijos iššūkių. Analizuojant mokslinę literatūrą (Borhan et al., 2025; Chari & Agrawal, 2018; Kasauli et al., 2021; Rasheed et al., 2021; Yaseen et al., 2022) pastebėta, jog reikalavimų inžinerijos procesų iššūkiai yra gana plačiai tyrinėjami. Tačiau taip pat pastebima, jog daugeliu atvejų netyrinėjamos prie projekto sėkmės prisidedančios gerosios reikalavimų inžinerijos praktikos. Atsižvelgiant į tai, mokslinių tyrimų autoriai (Ferrari et al., 2022; Kasauli et al., 2021; Rasheed et al., 2021; Money et al., 2021; Nizam, 2022) nurodo reikalavimų inžinerijos procesų tobulinimo kryptį kaip galimybę sekantiems moksliniams tyrimams. Tokio tyrimo rezultatai – rekomendacijos – padėtų verslui įgyti konkurencinį pranašumą, sumažinti klaidų skaičių bei kurti tiksliau kliento poreikius atitinkančius produktus.

2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose teoriniai sprendimai

2.1. Reikalavimų inžinerijos samprata ir svarba

Nuo tada, kai žmonija pradėjo statyti pastatus ar kitus statinius, atsirado poreikis turėti reikalavimus. Be aiškių, nors ir žodinių reikalavimų, nebūtų buvę įmanoma pastatyti tokių statinių kaip piramidės. Vienas iš pirmųjų rašytinių reikalavimų pavyzdžių užfiksuotas hebrajų Biblijoje, Pradžios knygoje. Joje buvo aprašyti aiškūs reikalavimai, kaip pastatyti Nojaus arką. Reikalavimai yra svarbūs bet kokiam projektui – tiek dideliame, tiek mažame. Net vykdydami mažus projektus, kuriuos atliekame patys, turime bent jau idėją, ką norime pasiekti ir kam tai panaudosime. Tokiu atveju, kai neturime rašytinių reikalavimų, vis tiek susidarome suvokimą, kokio rezultato tikimasi. Nuo pat žmonijos istorijos pradžios reikalavimų inžinerija buvo sutinkama visur – net tokia sritis kaip maisto ruošia irgi turi tam tikrus reikalavimus, vadinamus receptais (Crowder & Hoff, 2022).

Reikalavimų inžinerija – tai plačiai skirtingas pramonės šakas apimanti disciplina, turinti lemiamą įtaką projektų sėkmės rodikliams. Reikalavimų inžinerijos procesai įtraukia įvairių sričių specialistus, kurie dirbdami kartu geba suformuluoti išsamius reikalavimus, užtikrinant, kad nei vienos srities aspektai nebūtų praleisti. Toks bendradarbiavimas ypatingai pastebimas, kai kuriamos sudėtingos aviacijos, karinės, automobilių sistemos (Crowder & Hoff, 2022).

Kaip teigia Crowder'is ir Hoff'as (2022) bei Palumbo ir kt. (2024) – reikalavimų inžinerija yra didžiausią įtaką IT projekto sėkmei daranti sritis. Bėgant laikui išaiškėjo, jog IT projektų nesėkmės dauguma atvejų nutinka dėl netinkamai valdomų reikalavimų ir laiku neiškomunikuojamų iššūkių. Šios problemos skatina kurti sisteminius metodus reikalavimų rinkimui, analizavimui ir valdymui visu programinės įrangos kūrimo ciklo laikotarpiu. IT projektus įgyvendinančios organizacijos praktinėje veikloje įsitikino, jog tradicinė krioklio (angl. *Waterfall*) projektų valdymo metodologija duoda prastus rezultatus ir, galiausiai, sukurtos sistemos neatitinka verslo poreikių (Chari & Agrawal, 2018).

Taikant krioklio (angl. *Waterfall*) projektų valdymo metodologiją IT projektams, susiduriama su šiomis problemomis: per daug lėtas reagavimas į pokyčius, ilgas ciklas iki produkto pristatymo klientams, mažai dėmesio skiriama grįžtamajam ryšiui (Dong et al., 2024). Pagal Chari ir Agrawal'ę (2018), reikalavimų inžinerijos procesų problemos taikant krioklio (angl. *Waterfall*) projektų valdymo metodologiją, kyla dėl šių priežasčių:

- **netikslūs ir klaidingi reikalavimai.** Prastai surinkti arba netinkamai interpretuoti reikalavimai sukuria papildomo darbo vėlesniuose projekto etapuose, atliekant įvairius pakeitimus ir taisyms, kurių buvo galima išvengti didesnę dėmesį skiriant reikalavimų surinkimo tikslumui.
- **abstraktūs reikalavimai.** Abstrakčiai surinkti ir tinkamai iki programavimo pradžios neišgryninti reikalavimai gali sukelti vėlavimus, nes sprendžiant neaiškumus prireikia nuolatinių pakeitimų, o tai reikalauja papildomo darbo ir didina klaidų skaičių.
- **reikalavimų kaita.** Dažni reikalavimų pakeitimai sukelia didelį darbo krūvį, nes esami dokumentai turi būti atnaujinami, o pakeitimai gali turėti įtakos projekto kaštams ir grafikui. Tai ypač didelė problema krioklio (angl. *Waterfall*) projektų valdymo metodologijoje, nes pakeitimai vykdomi vėlesnėse projekto stadijose ir turi įtakos anksčiau atliktam darbui.
- **didelis defektų skaičius dėl naujų reikalavimų.** Projekto eigoje besikeičiantys reikalavimai padidina riziką, kad atsirastų klaidų ir defektų. Šių reikalavimų integravimas dažnai būna neefektyvus, kainuoja papildomai finansine ir laiko prasmėmis.

- **apribotas lankstumas reaguoti į pokyčius.** Krioklio (angl. *Waterfall*) modelis yra linijinis ir nuoseklus, todėl jis sunkiai prisitaiko prie greitų pokyčių. Bet kokie pakeitimai reikalauja sugrįžti į ankstesnes stadijas, o tai veda prie papildomų sąnaudų ir vėlavimų.

Pagrindinis ir reikšmingiausias skirtumas tarp krioklio (angl. *Waterfall*) ir „Agile“ metodologijų yra tas, jog krioklio metodologija pasižymi nuoseklumu, kai vienas projekto etapas seka kita etapą. Tuo tarpu „Agile“ metodologijoje procesai vykdomi lygiagrečiai ir apima daugkartines iteracijas (Dong et al., 2024). Šie krioklio metodologijos trūkumai skatina įmones rinktis „Agile“ metodologiją, siekiant išlaikyti konkurencingumą dinamiškoje rinkoje. Sudėtinga IT projektų prigimtis yra pagrindinė priežastis, lemianti pagal krioklio metodologiją valdomų IT projektų nesėkmes. „Agile“ metodologija padeda įveikti trūkumus, su kuriais susiduriama pagal krioklio metodologiją valdomuose projektuose, nes orientuojasi į dažną tarpinių rezultatų pristatymą, efektyvų resursų panaudojimą, prisitaikymą prie rinkos pokyčių. Dėl šių savybių „Agile“ projektai pasižymi aukštesniu sėkmės rodikliu nei tie, kurie valdomi pagal krioklio metodologiją (Ogirri & Idugie, 2024).

Didžioji dalis IT įmonių pakeitė krioklio projektų valdymo metodologiją į „Agile“, kadangi ši metodologija sprendžia daugelį tradicinių projektų valdymo metodų trūkumų (Colleen, 2024). Atlikus krioklio ir „Agile“ metodologijų lyginamąją analizę (Hoy & Xu, 2023; Srivastava et al., 2023; Dong et al., 2024; Rasheed et al., 2021; Yaseen et al., 2022), toliau pateikiami esminiai „Agile“ projektų valdymo metodologijos privalumai:

- **prisitaikymas prie pokyčių.** Reikalavimų inžinerijos procesai pagal „Agile“ metodologiją valdomuose projektuose įgalina prisitaikymą ir nuolatinį grįžtamojo ryšio įtraukimą į vystymo procesą, kuris leidžia lengviau adaptuotis prie besikeičiančių reikalavimų ir kurti realius poreikius atitinkančius produktus;
- **klientus įtraukiantis vystymas.** „Agile“ leidžia vykdyti iteratyvų ir inkrementinį sistemos vystymą nuolat tikslinant reikalavimus. Tai sumažina projekto nesėkmės riziką, kuri gali kilti dėl nebeaktualių ar netikslių reikalavimų;
- **efektyvus komunikavimas su suinteresuotomis šalimis.** Valdant IT projektus pagal „Agile“ metodologiją ypatingas dėmesys skiriamas nuolatiniam bendravimui su klientais ir kitomis suinteresuotomis šalimis, siekiant užtikrinti reikalavimų atitikimą verslo poreikiams ir sumažinti nesusipratimų tikimybę;
- **naudotojų istorijos (angl. *User Stories*).** „Agile“ metodologijoje naudojamos naudotojų istorijos padeda geriau suprasti reikalavimus bei palengvina komunikaciją. Taip išsprendžiamos bendravimo problemos, būdingos tradicinei reikalavimų inžinerijai;
- **nuolatinis tarpinio rezultato validavimas.** „Agile“ praktikos apima nuolatinį tarpinio rezultato validavimą viso projekto metu. Pagal standartines „Agile“ gaires, naujas sistemos funkcionalumas perduodamas kliento testavimui pasibaigus kiekvienam sprintui. Toks požiūris leidžia klaidas pastebėti gerokai anksčiau ir dėka dažno grįžtamojo ryšio jas ištaisyti be didesnių sąnaudų, kol projektas nepasiekęs galutinių etapų.

Apibendrinant aptartus privalumus galima teigti, kad teisingai taikomi „Agile“ metodologijos principai įgalina lengvesnį projekto prisitaikymą prie pokyčių, suteikia daugiau lankstumo. „Agile“ metodologijos esmė yra vykdyti darbus mažomis iteracijomis, nuolat atsižvelgti į grįžtamąjį ryšį iš klientų ir atitinkamai adaptuoti sprendimus (Dong et al., 2024). Pagal „Agile manifestą“ (2001) deklaruojamos pagrindinės „Agile“ metodologijos vertybės:

- žmonės ir bendravimas svarbiau nei procesai ir įrankiai;
- veikiantis produktas svarbiau nei išsami dokumentacija;

- nuolatinis bendravimas su klientu svarbesnis nei derėjimasis dėl sutarčių;
- gebėjimas prisitaikyti prie pokyčių svarbiau nei griežtas pirminio plano laikymasis.

Pagal Dong'ą ir kt. (2024) „Agile“ metodologijos teikiamos naudos yra:

- didesnis kliento pasitenkinimas;
- greitas prisitaikymas prie pokyčių;
- produkto pristatymas mažomis dalimis;
- bendradarbiavimas;
- motyvuota komanda;
- komunikacija;
- veikiančio produkto kūrimas;
- pastovus produkto dalių pristatymo tempas;
- skiriamas dėmesys detalėms ir dizainui;
- paprastumas;
- savarankiškos komandos;
- reguliarius rezultatų ir procesų vertinimas (retrospektyvos).

Nuolatinis reikalavimų prioritetų nustatymas yra esminė „Agile“ reikalavimų inžinerijos praktika, kuria siekiama užtikrinti, kad projektas nuolat prisitaikytų prie verslo poreikių ir besikeičiančių aplinkos sąlygų. Ši praktika skiriasi nuo tradicinių metodų, kuriuose reikalavimai prioritetizuojami tik projekto pradžioje. „Agile“ padeda išvengti per didelio apimties išsiplėtimo (angl. *overscoping*) problemos ir įgyvendinti projektą su turimais resursais. Nuolat detalizuojant ir prioritetizuojant reikalavimus, sumažinama rizika įgyvendinti nereikalingas ar mažai vertės turinčias sistemos funkcijas. Anot Inayat'o ir kt. (2015), išskiriami šie pagrindiniai nuolatinio reikalavimų prioritetų nustatymo bruožai:

- **nuolatinė peržiūra.** Kiekvienos iteracijos metu klientas ir komanda vertina reikalavimus ir nusprendžia, kuriuos iš jų įgyvendinti toliau;
- **lankstumas.** Reikalavimai gali keistis priklausomai nuo naujų įžvalgų ar besikeičiančių rinkos sąlygų;
- **vertė verslui ir rizika.** Prioritetai nustatomi pagal tai, kuris sistemos funkcionalumas suteikia didžiausią vertę verslui arba padeda sumažinti neigiamą rizikos poveikį;
- **kliento įtraukimas.** Klientai aktyviai dalyvauja prioritetų nustatymo procese, taip užtikrinant, kad projektas atitiktų jų lūkesčius.

Pagal „Agile“ metodologiją valdomuose projektuose ypatingas dėmesys skiriamas reikalavimų pokyčių valdymui. „Agile“ metodologija, skirtingai nei kriklio metodologija, leidžia lanksčiai prisitaikyti prie atsiradusių pokyčių kliento versle. Apibendrinant mokslinės literatūros analizę (Anifa et al., 2024; Inayat et al., 2014; Rasheed et al., 2021) sąraše pateikiamos reikalavimų pokyčio valdymo praktikos „Agile“ projektuose:

- **grįžamojo ryšio rinkimas.** Pastovus bendravimas su klientais ir grįžamojo ryšio rinkimas viso projekto laikotarpiu leidžia anksčiau identifikuoti reikalavimų pasikeitimus ir prie jų prisitaikyti. Taikant šią praktiką, sumažėja poreikis atlikti didelius pakeitimus vėlesnėse projekto stadijose ir išvengiama bereikalingo sistemos perdarymo;
- **greita prototipų validacija.** Klientams pateikiami sukurti sistemos prototipai, pavyzdžiui sukurtas interaktyvus sistemos dizainas, kuriame veikia tam tikros navigacijos funkcijos. Prototipavimas leidžia lengviau valdyti tolimesnius pokyčius ir sukuria bendrą supratimą apie kuriamą funkcionalumą tarp kliento ir projekto komandos. Prototipavimas leidžia sumažinti klaidų skaičių, išvengiant netinkamos reikalavimų interpretacijos;

- **lankstumas pridėti ar pašalinti funkcijas.** Dėl didesnio „Agile“ metodologijos suteikiamo lankstumo, projekto eigoje identifikavus naujo funkcionalumo poreikį, galima jį įtraukti į esamą planą, atitinkamai sudedant prioritetus ir atsisakant mažiau būtinų funkcijų;
- **mažesnis poreikis išsamiems pokyčių valdymo dokumentams.** Kadangi komunikacija vyksta tiesiogiai, mažėja poreikis detaliam dokumentavimui ir formalizuotoms pokyčių valdymo procedūroms;

Apibendrinant galima teigti, jog „Agile“ metodologijos pritaikymas išsprendė daugelį problemų, su kuriomis buvo susiduriama IT projektus valdant pagal krioklio metodologiją. Iš esmės, toks rezultatas pasiekiamas dėl iteracijomis grįsto darbo modelio ir nuolatinio grįžtamojo ryšio įtraukimo į sistemų kūrimo procesą. Nepaisant aukščiau išvardintų „Agile“ metodologijos privalumų, atsiranda nauji iššūkiai suderinti „Agile“ metodologijos lankstumą su planavimu ir nuspėjamumu (Inayat et al., 2015). Taip pat pastebėta, kad nedidelis „Agile“ projektų metu parengiamos dokumentacijos kiekis sukuria papildomas problemas projekto perdavimo metu arba keičiantis komandos sandarai, kai sukauptos žinios išeina kartu su žmonėmis.

2.2. Reikalavimų inžinerijos procesai

Procesas yra sudėtingas, įvairialypis reiškinys, kurio supratimas priklauso nuo nagrinėjamo konteksto. Mokslinėje literatūroje dažnai sutinkami skirtingi procesų apibūdinimai, kurie dažniausiai remiasi skirtingomis prielaidomis. Hilmer'is (2016) pateikė išsamią procesų sampratos sisteminimo studiją, kurioje identifikavo net 75 procesų savybes. Hilmer'io išskirti proceso apibrėžimai pagal skirtingus autorius pateikiami 1 lentelėje.

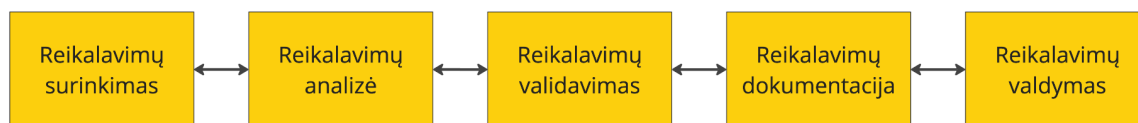
1 lentelė. Proceso apibrėžimai pagal skirtingus autorius (Hilmer, 2016)

Autoriai	Apibrėžimas
Hammer ir Champy (1994)	Veiklų rinkinys, kuris, naudodamas vieną ar kelias įvestis, sukuria vertingą rezultatą klientui. Kaip pavyzdys minimas naujo produkto kūrimas.
Scheer ir Jost (1996)	Išskiria procesų modeliavimą, aprašant funkcijas, susietas su laiku ir įvykiais, ir pabrėžia procesų tarpdisciplininį pobūdį.
Österle	Išskiria informacinių technologijų vaidmenį, apibrėždamas procesus kaip veiklų seką, kuri jungia įmonės strategiją ir informacines sistemas.
Berkau (1998)	Procesus skirsto į techninius ir verslo, pabrėždamas jų dokumentavimo svarbą.
Gehring (1998)	Tikslinga, laike išsidėsčiusi veiklų seka, siejama su organizacijos tikslais ir jų įgyvendinimu, naudojant informacines technologijas.

Tokie įvairūs tos pačios sąvokos apibrėžimai leidžia formuoti išvadą, jog proceso apibrėžimas priklauso nuo konteksto ir tiriančiojo asmens. Šio tyrimo kontekste labiausiai tinka Hammer'io ir Champy'io apibrėžimas, kuris nusako, jog procesas tai yra veiklų rinkinys, kuriam reikia tam tikros įvesties ir yra sukuriamas vertingas rezultatas. Šis apibrėžimas gali būti pritaikomas reikalavimų inžinerijos procesų tobulinime, nes pagrindinis reikalavimų inžinerijos tikslas yra užtikrinti, kad kuriama sistema ar

produktas atitiktų kliento poreikius ir tikslus. Reikalavimų inžinerijos procesai galiausiai sukuria rezultatą – aiškius, struktūrizuotus ir vertę kuriančius reikalavimus.

Šiuolaikinis reikalavimų valdymas yra nenutrūkstantis, grįžtamuoju ryšiu pagrįstas procesas, kuris trunka visą sistemos gyvavimo ciklą. Nors pradinė sistemos kūrėjų komanda gali keistis, sistemos vystymo ar palaikymo darbai gali būti perduoti palaikančiam personalui, o reikalavimų valdymo procesas tęsiasi iki sistemos gyvavimo pabaigos. Reikalavimų valdymas yra reikalavimų inžinerijos pagrindas, atsirandantis pačioje sistemos reikalavimų rinkimo pradžioje. 4 paveiksle pateikiamas abstraktus, pagrindinius procesus atspindintis, reikalavimų inžinerijos procesų srauto modelis pagal Crowder'į ir Hoff'ą (2022), kuris apibendrina ir kitų autorių moksliniuose darbuose pateikiamus reikalavimų inžinerijos procesų aprašymus (Inayat et al., 2015; Alazzawi et al., 2023; Hoy & Xu, 2023; Ferrari et al., 2022). Būtina pabrėžti, jog šie procesai gali vykti abejomis kryptimis.



4 pav. Reikalavimų inžinerijos procesų diagrama (Crowder & Hoff, 2022)

Remiantis moksline literatūra (Rasheed et al., 2021; Crowder & Hoff, 2022; Inayat et al., 2015; Alazzawi et al., 2023; Ferrari et al., 2022) reikalavimų inžinerijos procesai nuo sistemos gyvavimo ciklo pradžios iki pabaigos yra skirstomi į šiuos pagrindinius subprocesus:

- **reikalavimų surinkimas.** Šis procesas apima visų sistemos reikalavimų surinkimą. Tai yra pirmas ir esminis žingsnis reikalavimų inžinerijos procese, kurio įgyvendinimo sudėtingumas priklauso nuo kuriamos sistemos sudėtingumo ir kliento žinių bei supratimo lygio. Neišgrynintus ir abstrakčius (aukšto lygio) reikalavimus dažnu atveju pateikia klientas, o verslo analitikų užduotis yra gautus reikalavimus išgryninti, tinkamai interpretuoti ir apipavidalinti. Rezultatas turi būti toks, jog visos suinteresuotos šalys (klientas, verslo analitikai, programuotojai ir testuotojai) reikalavimus suprastų taip pat. Reikalavimų surinkimo etapas kelia nemažai iššūkių ir dažnu atveju užtrunka mėnesius, kadangi net tarp skirtingų užsakovo atstovų gali atsirasti nesutarimų, kurie sukuria papildomus iššūkius verslo analitikams. Galutinis šio proceso tikslas – prieš pradėdant sistemos dizaino kūrimo darbus, užtikrinti, kad tarp visų suinteresuotųjų šalių būtų sukurtas vieningas reikalavimų suvokimas (Crowder & Hoff, 2022; Ferrari et al., 2022);
- **reikalavimų analizė ir validavimas.** Pagrindinis šio proceso tikslas – išgryninti reikalavimus, kurie buvo surinkti pirmame etape iš kliento. Vis dėlto reikalavimų analizė neapsiriboja vien tik iš kliento surinktais reikalavimais, kadangi dažniausiu atveju tai yra ne iki galo išstobulinti reikalavimai. Analizė reikalinga tam, kad būtų išsiaiškinti kliento poreikiai, kuriuos jis laikė savaime suprantamais, aiškiai neišreiškė pirminiame reikalavimų rinkinyje. Be tiesiogiai surinktų reikalavimų, reikalavimų analizės procesas dažnai reikalauja išvesti papildomus, smulkesnius reikalavimus. Tai vadinama reikalavimų išvedimu (angl. *requirement derivation*), kai aukšto lygio reikalavimai yra suskaidomi į konkretesnius, skirtus tam tikriems sistemos komponentams ar posistemėms. Aiškiai neišreikšti reikalavimai būdingi beveik visiems projektams, todėl ankstyvas jų identifikavimas gali padėti išvengti vėlesniuose etapuose sekančių išlaidų bei didelių sistemos pakeitimų, kurie užtesia projekto terminus. Nors yra numatyta viena pagrindinių taisyklių – nepradėti kodo rašymo, kol reikalavimų analizė nėra baigta, tačiau dažnu atveju jaučiamas spaudimas iš suinteresuotųjų šalių kuo greičiau pereiti prie kodo rašymo, o tai priveda prie naujų reikalavimų atsiradimo eigoje ir reikšmingų sistemos pakeitimų. Reikalavimų analizės

metu siekiama sukurti vieningą terminų ir sąvokų supratimą tarp visų suinteresuotų šalių. Jeigu atsitinka taip, jog reikalavimai šiame etape yra ne iki galo apibrėžti, gali susidaryti situacija, kai galutinis produktas neatitinka kliento lūkesčių dėl skirtingų terminų ir sąvokų suvokimo. Analizės metu siekiama įvertinti, ar surinkti reikalavimai nėra pernelyg priklausomi nuo šiuo metu esamų sistemų ar modelių, kurie laikui bėgant gali pasikeisti ir neigiamai paveikti projektą. Mažesnis priklausomybių skaičius padeda užtikrinti lankstesnę sistemą ir įgalina jos plėtrą ateityje. Apibendrinant literatūros analizę (Inayat et al., 2015; Alazzawi et al., 2023; Hoy & Xu, 2023; Ferrari et al., 2022), galima išskirti šias reikalavimų analizės dalis:

- **aplinkos analizė.** Siekiama visapusiškai suprasti sistemos pritaikymo atvejus ir kontekstą, kuriame bus sistema;
- **reikalavimų klasifikacija.** Kiekvienas identifikuotas reikalavimas turi būti klasifikuojamas, t.y. priskiriamas tam tikrai kategorijai: funkciniai, greitaveikos, saugumo ir panašūs reikalavimai. Nustatoma, ar neatsiranda prieštaravimų jau esantiems reikalavimams;
- **prieštaraujančių reikalavimų derinimas.** Kai aptinkami prieštaravimai tarp reikalavimų, atsirandantys iš skirtingų suinteresuotų šalių ar skirtingų poreikių, nustatinėjamas reikalavimų prioritetas ir iš naujo derinama su klientu, atnaujinama dokumentacija;
- **reikalavimų validavimas.** Kiekvienas identifikuotas reikalavimas patikrinamas bendrų sistemos tikslų atitikimo atžvilgiu. Tai pasikartojantis procesas, kurio tikslas yra užtikrinti, kad kiekvienas klientų ir kitų suinteresuotų šalių sprendimas bei numatomas sistemos pakeitimas būtų suderintas su esminiais sistemos tikslais;
- **reikalavimų dokumentavimas.** Galutiniame etape sukuriami išsamūs reikalavimų specifikacija, kurioje pateikiama kiekvieno reikalavimo apibrėžtis bei jo testavimo metodai. Reikia pabrėžti tai, jog reikalavimų analizė yra iteracinis procesas ir per visus žingsnius yra pereinama daug kartų. Siekiant išlaikyti sprendimų priėmimo atsakumą, svarbu užtikrinti, kad viso sistemos kūrimo gyvavimo ciklo metu reikalavimai būtų tinkamai sekami ir dokumentuojami. Dažnu atveju papildomi reikalavimai išryškėja sistemos kūrimo metu, todėl svarbu juos tinkamai dokumentuoti, kad ateityje būtų aiškus suvokimas, kodėl buvo priimti tam tikri sprendimai;
- **reikalavimų prioritetizavimas.** Sistemos vystymo eigoje nutinka nenumatytų dalykų, pavyzdžiui, apskarpomas biudžetas arba sumažinamas skirtas laikas. Dėl to kyla poreikis reikalavimus prioritetizuoti, aukščiau iškeliant esmines funkcijas, nuošalyje paliekant mažiau svarbius punktus. Prioritetizavimas atliekamas atsižvelgiant į kliento poreikius ir esminių sistemos funkcijų veikimo užtikrinimo kriterijus.
- **reikalavimų valdymas.** Išskiriami keturi pagrindiniai reikalavimų valdymo tikslai: struktūros užtikrinimas, pastovių valdymo principų užtikrinimas, dokumentacijos sukūrimas, pokyčių valdymas.

Apibendrinant galima teigti, jog reikalavimų inžinerijoje tinkamiausias proceso apibrėžimas yra pagal Hammer'į ir Champ'į (1994), kuris nusako, jog procesas tai yra veiklų rinkinys, kuriam reikia tam tikros įvesties ir yra sukuriamas vertingas rezultatas. Atlikus mokslinės literatūros analizę ir susisteminus jos rezultatus, galima teigti, jog reikalavimų inžinerijos procesas gali būti skaidomas į: reikalavimų surinkimą, analizę, validavimą, dokumentavimą ir galiausiai valdymą. Būtina pabrėžti, jog šie procesai vyksta abejomis kryptimis, kadangi projekto metu (ypatingai „Agile“) reikalavimai kinta, iškyla nauji

poreikiai, o taip pat viso proceso metu atsiranda būtinybė reikalavimus tikslinti, todėl per kiekvieną etapą yra pereinama keletą kartų.

2.3. Procesų analizės ir tobulinimo metodai

Verslo procesų tobulinimas (angl. *Business Process Improvement*) šiomis dienomis išlieka ypatingai svarbus. Tobulinimas apibrėžiamas kaip proceso pagerinimas, koreguojant jį taip, kad procesas taptų efektyvesnis, našesnis ir tinkamai adaptuotas esamai situacijai. Pagrindinės verslo procesų tobulinimo priežastys yra poreikis sumažinti verslo išlaidas ir pagerinti našumą, taip pat padidinti klientų pasitenkinimą, atitikti keliamus reikalavimus. Organizacijos didelę dalį energijos ir resursų nukreipia į procesų tobulinimą, kad išliktų konkurencingos dinamiškoje rinkoje. Procesams tobulinti jau yra sukurta nemažai metodologijų ir įrankių, vis dėlto nemažai kompanijų vis dar susiduria su verslo procesų tobulinimo iššūkiais (Casebolt et al., 2020). Siekiant nuolatinio procesų tobulinimo, būtina išanalizuoti esamus procesus ir ieškoti naujų galimybių juos patobulinti (Pratama et al., 2017).

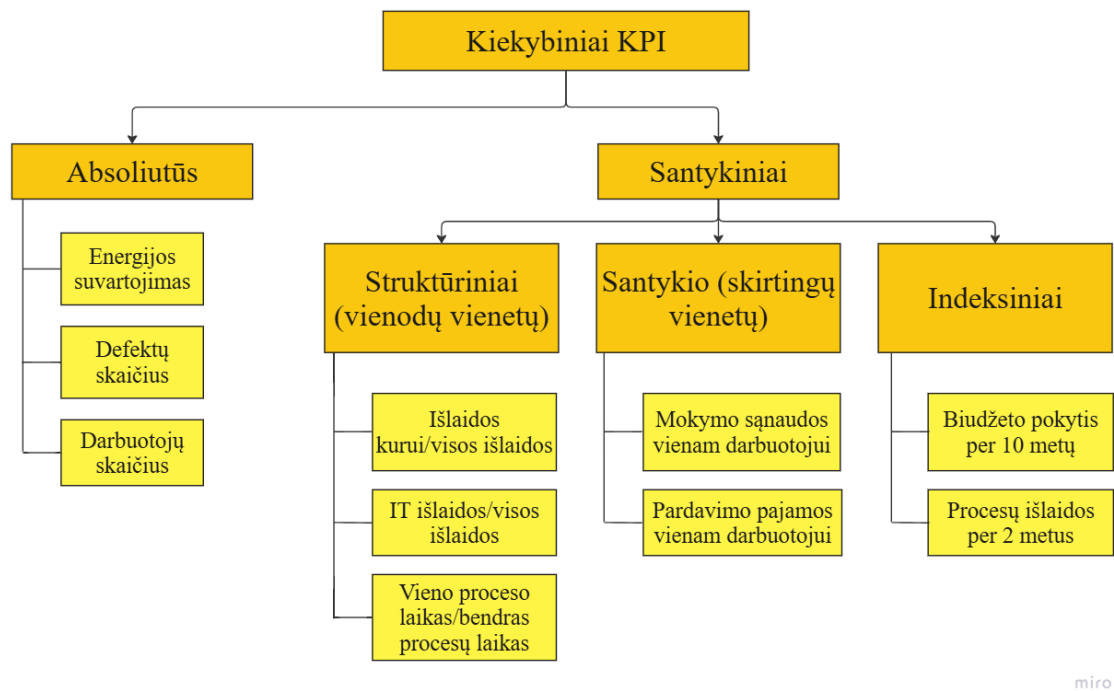
2.3.1. Pagrindiniai veiklos rodikliai

Siekiant stebėti tam tikrų procesų patobulinimų teikiamą naudą būtinos procesų vertinimo metrikos. Proceso trukmės sumažinimas ir efektyvumo bei našumo didinimas yra išskiriami kaip pagrindiniai procesų tobulinimo uždaviniai (Gadatsch, 2023). Tobulinant procesus peržiūrimi atliekami veiksmai ir operacijos, identifikuojamas neoptimalus išteklių panaudojimas ir siekiama identifikuotus trūkumus pašalinti. Po atliktų korekcijų tikimasi, jog procesų vertinimo metrikos įgys geresnes nei prieš tai buvusias vertes (Phokwane & Arnesh, 2021).

Procesų vertinimo metrikos standartizuotai vadinamos pagrindiniais veiklos rodikliais KPI (angl. *Key Performance Indicators*). Pagrindiniai veiklos rodikliai leidžia organizacijoms išmatuojamu dydžiu įsivertinti esamų procesų efektyvumą, išteklių panaudojimą, organizacijos tikslų pasiekimą. Pasitelkiant pagrindinius veiklos rodiklius sukuriamas išsikeltų tikslų pasiekimo progreso matavimas. Pagrindiniai veiklos rodikliai dažniausiai išreiškiami normalizuotais skaičiais, pavyzdžiui nuo 0% iki 100% (Hollender et al., 2016), jie turi būti tiesiogiai suderinti su organizacijos strateginiais tikslais, o KPI dizainas užtikrina, kad jie būtų išmatuojami ir objektyvūs (Řepa, 2019).

Procesų KPI galima suskirstyti į šias grupes:

- **kiekybiniai.** Šie rodikliai yra grindžiami išmatuojamais procesų duomenimis. Kiekybiniai KPI apima našumą, energijos suvartojimą, prastovų laiką, defektų skaičių (Hollender et al., 2016). Kiekybiniai KPI dar skirstomi į du tipus – absoliučius ir santykinius dydžius. Absoliutūs rodikliai nurodo tam tikrą kiekį, pavyzdžiui, darbuotojų skaičių, dalyvaujantį procese. Tačiau šie rodikliai yra ribotos informatyvumo vertės, nes neturi palyginimo bazės. Santykiniai rodikliai nusako santykį tarp dviejų dydžių, pavyzdžiui mokymo išlaidos vienam darbuotojui. Kiekybinių KPI klasifikavimas pateikiamas 5 paveiksle;
- **kokybiniai.** Kai kurie verslo procesai yra kokybinio pobūdžio ir negali būti lengvai įvertinti tradiciniais kiekybiniais rodikliais. Tokiais atvejais naudojami netiesioginiai rodikliai, tokie kaip tikslai, sudėtingumas ir procesų brandos lygis (Pidun & Felden, 2011).



5 pav. Kiekybinių KPI klasifikavimas (Gadatsch, 2023)

Nustatant pagrindinius veiklos rodiklius svarbu tinkamai juos pasirinkti, įvertinant pritaikymo galimybes, kiek sudėtinga juos sekti ir kokią naudą organizacijai suteiks konkrečių rodiklių sekimas. Remiantis moksline literatūra, gali būti išskiriami šie pagrindinių veiklos rodiklių atrankos kriterijai:

- **rodiklio atitikimas organizacijos tikslams.** Pasirinktas rodiklis privalo turėti aiškią sąsają su organizacijos tikslais ir būti suderintas su strategija, atspindėti tikslo siekimo progresą (Wetzstein, 2016);
- **kiekybinių ir kokybinių rodiklių balansas.** Nors KPI dažniausiai orientuojasi į kiekybinius rodiklius, kokybinių rodiklių įtraukimas leidžia gauti išsamesnį veiklos rezultatų vaizdą (Pidun & Felden, 2011);
- **pritaikomumas ir lankstumas.** KPI turi būti pritaikomi prie besikeičiančių verslo procesų ir veiklos sąlygų. Tai galima pasiekti informacinių technologijų pagalba, kurios leidžia sukurti pagal specifinius poreikius adaptuotus KPI (Calabro et al., 2015);
- **galimybė integruoti su IT sistemomis.** KPI turėtų būti integruoti su IT sistemomis, kad būtų užtikrintas sklandus duomenų rinkimas ir analizė. Toks integravimas leidžia realiuoju laiku stebėti rodiklius ir atlikti reikiamus pakeitimus verslo procesuose (Wetzstein, 2016);
- **atitikimas SMART kriterijams.** KPI turi būti konkretūs (angl. *specific*), pamatuojami (angl. *measurable*), pasiekiami (angl. *achievable*), aktualūs (angl. *relevant*) organizacijos tikslams, apibrėžti laike (angl. *time-based*) (Tømmerås et al., 2020);
- **organizacijos galimybė daryti įtaką.** Tam, kad KPI turėtų prasmę ir būtų galima jį pagerinti, turi būti matuojami tokie parametrai, kuriems įtaką daro vykdoma veikla (Tømmerås et al., 2020);
- **aiškūs visoms suinteresuotoms šalims.** KPI prasmę ir matavimo būdą turėtų suprasti visos suinteresuotos šalys, tam, kad būtų išvengta nesusipratimų dėl skirtingų KPI interpretacijų (Rojas et al., 2013).

Kokybiniai pagrindiniai veiklos rodikliai naudojami vertinant verslo procesų aspektus, kurių negalima lengvai išreikšti kiekybiniais duomenimis. Kokybiniais rodikliais siekiama pamatuoti darbuotojų

pasitenkinimo lygį, klientų polinkį rekomenduoti įmonės paslaugas ir kitus panašius įmonės veiklos aspektus. Lyginant su kiekybiniais rodikliais, kokybiniai rodikliai suteikia gilesnes įžvalgas apie įmonės veiklą, subjektyvius aspektus. Kokybiniai veiklos rodikliai pasitarnauja, siekiant įvertinti procesų kokybę ir jų įtaką organizacijos tikslams. Nors kokybiniai duomenys tiesiogiai neišsiverčia į skaitinius duomenis, jie yra itin svarbūs ilgalaikiam augimui ir sėkmei (Pidun & Felden, 2011). Kokybinių KPI matavimas yra sudėtingas dėl tokių rodiklių neskaitinio pobūdžio. Siekiant kokybinius KPI pamatuoti, duomenys gali būti gaunami interviu, anketų, stebėjimo būdais. Pavyzdžiui, darbuotojų pasitenkinimas gali būti pamatuojamas atlikus anoniminę apklausą įmonėje (Sardjono et al., 2020).

Remiantis Hoy ir Xu (2023) nustatytais pagrindiniais reikalavimų inžinerijos iššūkiais, galima išskirti šiuos kokybinių KPI pavyzdžius:

- **pakankamas dokumentacijos kiekis.** Atliekama naujai prie projekto komandos prisijungusių žmonių apklausa, siekiant nustatyti, ar buvo pasiekama pakankamai informacijos apie kuriamą sistemą, kad galėtų atlikti jiems priskirtas užduotis.
- **reikalavimų aiškumas.** Atliekama projekto komandos narių apklausa, leidžiant jiems įvertinti, kaip aiškiai, jų manymu, pateikti reikalavimai. Taip pat, galima atlikti testą, kuris pamatuotų, kokią dalį reikalavimų komanda interpretavo teisingai. Tokiu būdu kokybinis parametras gali būti paverčiamas į skaitinę reikšmę;
- **suinteresuotųjų šalių įsitraukimas.** Naudojant apklausas ir interviu, galima įvertinti, kiek suinteresuotosios šalys dalyvauja reikalavimų rinkimo ir patvirtinimo etapuose, taip pat stebint dalyvavimo susirinkimuose lygį ir komunikacijos dažnumą.

2.3.2. Procesų vizualizavimas

Vizualizavimas verslo procesų tobulinime užima reikšmingą poziciją, kadangi vizualus procesų atvaizdavimas leidžia vienu metu aprėpti didesnę procesų mastą, pamatyti bendrą vaizdą, kuriame lengviau pastebimi įvairūs trūkumai ir neoptimalūs procesų aspektai. Tokiam tikslui pasiekti plačiai naudojama verslo procesų modeliavimo (angl. *Business Process Modeling*) metodologija (Phokwane & Arnesh, 2021).

Procesams vizualizuoti skirti metodai skirstomi į skriptų ir diagramų grupes. Skriptų metodai yra pagrįsti tam tikromis skriptų (programavimo) kalbomis, kurios leidžia aprašyti procesus naudojantis programavimo kalbų standartais. Naudojantis šiuo metodu galima pasiekti aukštą procesų modelio tikslumą ir apibrėžti sudėtingą logiką bei sąsajas. Vis dėlto skriptų metodas turi didelį trūkumą – jo panaudojimas sudėtingas dėl poreikio turėti specifinių žinių, tokių kaip programavimo kalbos mokėjimas. Dėl šios priežasties praktikoje plačiau pritaikomi diagramų metodai, kurie yra ganėtinai paprasti naudoti ir nereikalauja nuodugnaus išankstinio pasiruošimo prieš pradėdant juos taikyti. Diagramų metodai išskiriami kaip tinkamesnė priemonė skirtingų sričių specialistų bendradarbiavimui dėl savo paprastumo (Gadatsch, 2023).

Pagal Gadatsch'ą (2023), diagramų metodai skirstomi į šiuos pogrupius:

- **duomenų srauto.** Šio pogrupio metodai tiesiogiai proceso neatvaizduoja. Tokius metodus tikslinga pasirinkti, siekiant atvaizduoti, kaip duomenys keliauja tarp atskirų proceso veiklų. Iš duomenų srauto diagramos negalima pamatyti procesų žingsnių sekos, todėl tokio tipo diagramos yra mažiau naudojamos šiuolaikinėje procesų vadyboje;
- **valdymo srauto.** Sutelkia dėmesį į veiklų seką, pagrindinis dėmesys skiriamas procesų modeliavimui. Šiuose metoduose svarbiausias aiškus veiksmų sekos vykdymo tvarkos

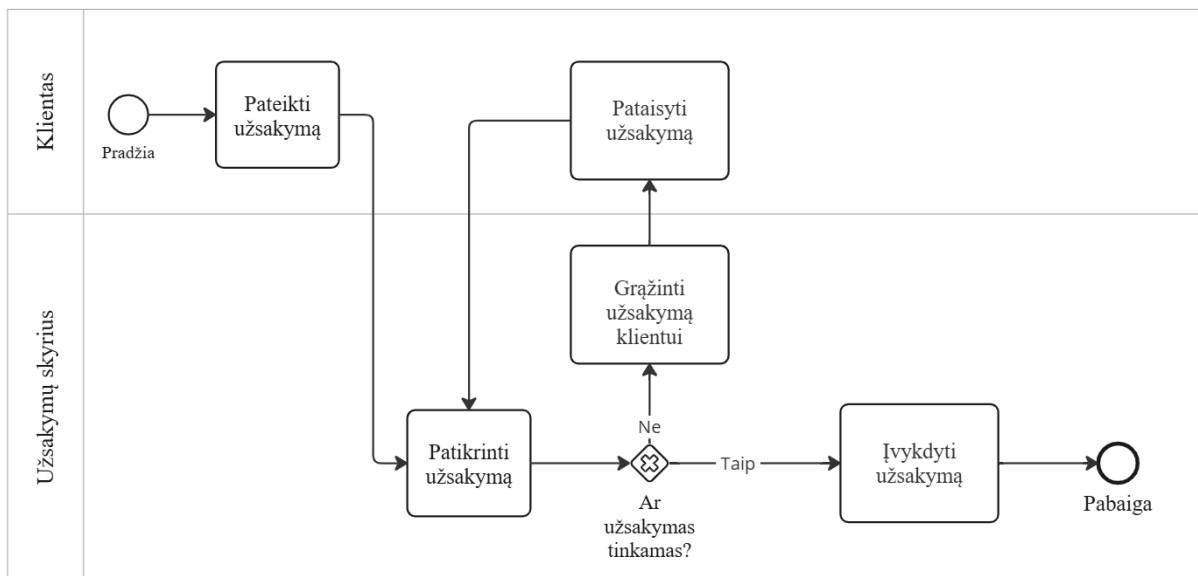
atvaizdavimas. Valdymo srauto diagramos naudojamos praktikoje, kai būtina suprasti procesų logiką, optimizuoti veiklų seką ar procesų našumą. Aiškūs grafikai ir diagramos padeda suprasti procesų eigą tiek techniniams, tiek netechniniams naudotojams. Praktikoje labiausiai paplitę metodai: procesų žemėlapis (angl. *Process Map*), plaukimo takelio diagrama (angl. *Swimlane Diagram*), vertės grandinės diagrama (angl. *Value Chain Diagram*), išplėstinė įvykiais grindžiama procesų grandinė (angl. *Extended Event-driven Process Chain*) ir verslo procesų modeliavimo ir notacijos metodas (angl. *Business Process Modeling and Notation method*);

- **objektų.** Ši metodų grupė atkeliavo iš programinės įrangos kūrimo srities, kurioje funkcijos ir duomenys yra priskiriami objektams. Objektinio modeliavimo metodai, pavyzdžiui unifikauta modeliavimo kalba (angl. *Unified Modeling Language*), yra plačiai taikomi procesų modeliavime;
- **mišrius.** Plačiai žinomos vertės srauto analizės (angl. *Value Stream Analysis*) ir verslo modelio drobės (angl. *Business Model Canvas*) diagramos jungia skirtingus procesų valdymo ir analizės aspektus, siekiant optimizuoti veiklą bei geriau suprasti organizacijos struktūrą ir procesus.

Procesų vizualizacijai dažnai naudojama standartizuota notacija – BPMN (angl. *Business Process Model Notation*). Notacija pateikia tris pagrindinius srauto objektų tipus: įvykiai (angl. *events*), veiklos (angl. *activities*) ir valdymo vartai (angl. *gateways*). Visi šie objektai susiję su tam tikrais proceso taškais laike ir atlieka skirtingas funkcijas procesų modeliavime (Pergl et al., 2019). Toliau apibūdinami pagrindiniai procesų srauto objektų tipai:

- **įvykiai.** Jie žymi, jog vyksta tam tikras atominis (mažas, nedalomas) veiksmas arba laukiama rezultato. Įvykiai gali būti: pradžia, tarpinis įvykis, pabaiga. Pradžia rodo, kur procesas prasideda, gaunamas signalas arba užklausa. Tarpinis įvykis rodo, jog procesas laukia arba reaguoja į kitą įvykį, pavyzdžiui pranešimo gavimą. Galinis įvykis rodo, kur baigiasi procesas, pasiekus tam tikrą rezultatą;
- **veiklos.** Tai procesų darbo elementai, kurie atlieka užduotis arba vykdo tam tikrą veiklą. Veiklos skirstomos į užduotis ir subprocesus. Užduotis – tai vienas žingsnis, kuris nėra detaliau struktūrizuotas, pavyzdžiui, užsakymo patvirtinimas. Subprocesas tai kompleksinė veikla, kuri apima papildomus procesus;
- **valdymo vartai.** Tai elementai, kurie naudojami procesų srautui nukreipti pagal tam tikras sąlygas. Išskiriami šie vartų tipai: išskirtiniai (angl. *exclusive*), imtiniai (angl. *inclusive*), lygiagretūs (angl. *parallel*), kompleksiniai (angl. *complex*). Išskirtiniai vartai nukreipia srautą tik vienu iš galimų kelių pagal sąlygą. Imtiniai vartai nukreipia srautą viena arba keliomis šakomis. Lygiagretūs vartai skirsto srautą į kelias lygiagrečias veiklas, kurios gali būti vykdomos vienu metu. Kompleksiniai vartai naudojami, kai sprendimas priklauso nuo kelių sudėtingesnių sąlygų.

Procesų srautas atvaizduojamas baseinuose (angl. *pool*) ir takeliuose (angl. *lanes*). Baseinai ir takeliai reprezentuoja procesų vykdytojus bei atsakomybes už tam tikrų proceso veiksmų pasiskirstymą. Baseinai apibrėžia organizaciją arba sistemą, kuriai priklauso tam tikri procesai, o juostos rodo atsakingus asmenis arba skyrius tame pačiame baseine (Pergl et al., 2019) (žr. 6 pav.).



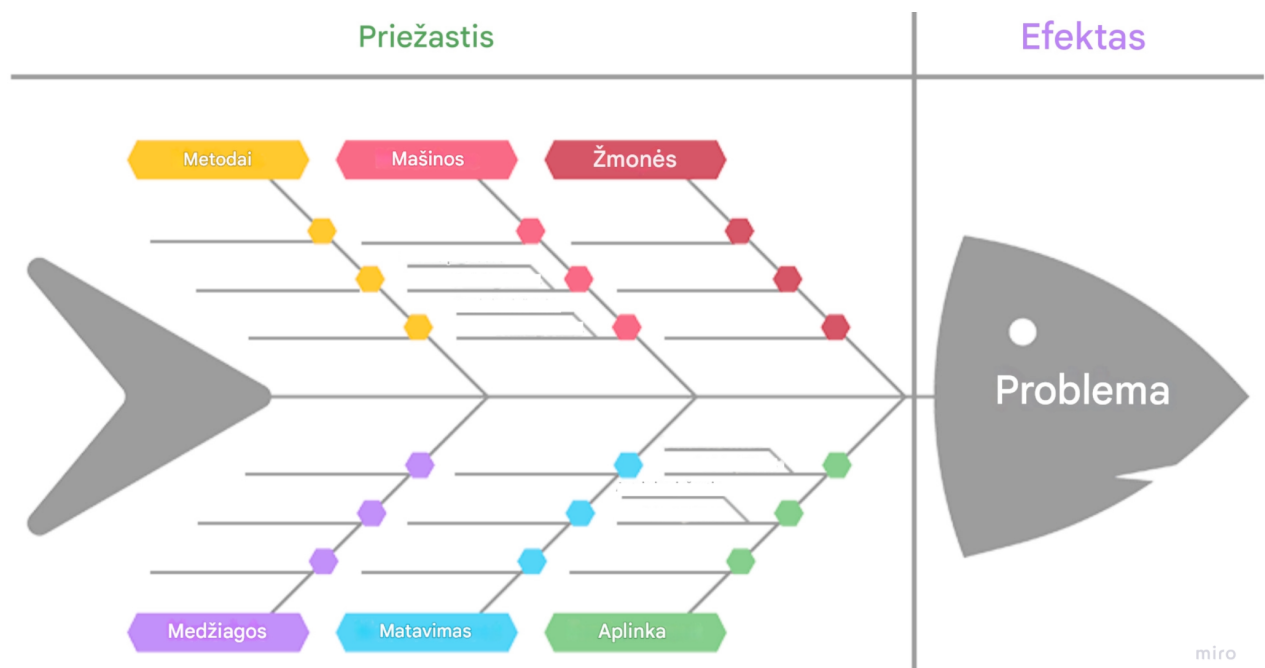
6 pav. Užsakymo vykdymo proceso pavyzdys, naudojant BPMN (Pergl et al., 2019)

6 paveiksle pateiktas pavyzdys, kaip atrodo paprasta BPMN diagrama, vaizduojanti užsakymo įvykdymą elektroninėje parduotuvėje. Procesas prasideda nuo kliento užsakymo pateikimo ir seka į užsakymų skyrių, kuriame užsakymas yra patikrinamas. Jeigu užsakymas pateiktas tinkamai, jis yra įvykdomas, jeigu netinkamas – grąžinamas klientui pataisyti ir vėl grįžtama prie tikrinimo. Procesas baigiasi įvykdžius užsakymą.

2.3.3. Ishikawa diagrama

Remiantis gerosiomis praktikomis, sprendžiant susidariusias problemas, pagrindinis dėmesys turėtų būti skiriamas simptomus sukėlusios šakninės priežasties analizei. Simptomų šalinimas dažnu atveju būna neefektyvus, kadangi nepašalinus šakninės priežasties, simptomai dažnu atveju pasikartoja. Kadangi šakninės problemos yra sudėtingų procesų dalis, todėl jos nėra akivaizdžiai matomos ir jų identifikavimui turi būti pasitelkiami tam tikri metodai (Alrabadi et al., 2023). Analitiniai metodai „5 kodėl“ (angl. *5 Whys*) ir Ishikawa (žuvies kaulo) diagrama gali būti naudojami kartu arba atskirai, siekiant identifikuoti šaknines problemos priežastis.

Ishikawa diagrama, dar žinoma kaip žuvies kaulo arba priežasties ir pasekmės diagrama, yra kokybės valdymo priemonė, naudojama sistemingai identifikuoti, analizuoti ir vizualiai pateikti galimas tam tikros problemos priežastis. Šis įrankis buvo sukurtas 1960 metais japonų profesoriaus Kaoru Ishikawa (Botezatu et al., 2019). Pagal Ishikawa filosofiją kokybės valdymas turi apimti visas įmonės veiklos sritis, o jo darbai turėjo gilų poveikį kokybės valdymo praktikoms visame pasaulyje (Lawless, 2024). Ishikawa diagrama primena žuvies skeletą: pagrindinė problema ar poveikis yra žuvies galva, o galimos priežastys – tai kaulai, išsišakojantys iš stuburo. Diagramos iliustracija pateikiama 7 paveiksle.



7 pav. Ishikawa diagramos šablonas (Lawless, 2024)

Ishikawa diagramos pagrindinis tikslas yra išsiaiškinti ir grupuoti problemos priežastis. Ji vizualiai pavaizduoja, kaip priežastys ir pasekmės yra susijusios. Dažniausiai priežastys grupuojamos pagal 5M+E modelį, į kurį įeina: žmonės (angl. *people*), mašinos (angl. *machines*), metodai (angl. *methods*), medžiagos (angl. *material*), matavimai (angl. *measurements*), aplinka (angl. *environment*) (Siwec et al., 2023). Diagramos taikymas padeda sistemiškai sudėlioti problemos priežastis, atsižvelgiant į problemos priežasties grupę, ir tokiu būdu identifikuoti pagrindines priežastis (Nascimento et al., 2024).

Pagal Lawless'ą (2024), Ishikawa diagramos metodo taikymas išskiriamas į šiuos pagrindinius etapus:

- **problemos identifikavimas.** Tiksliai ir aiškiai suformuluota problema užrašoma diagramos galvoje;
- **priežasčių kategorijos.** Prie žuvies stuburo pridedamos šakos su problemos priežasčių kategorijomis (žmonės, mašinos, metodai, medžiagos, matavimai, aplinka). Komandoje aptariama ir kiekvienoje kategorijoje užrašomos priežastys, galinčios prisidėti prie problemos. Prie kiekvienos kategorijos pridedamos mažesnės šakelės su konkrečiomis priežastimis;
- **analizė.** Peržiūrimos identifikuotos priežastys, analizuojama, kurios iš jų labiausiai tikėtinos kaip pagrindinės priežastys;
- **veiksmai.** Sukuriamas veiksmų planas, kaip bus pašalintos problemos priežastys. Įgyvendinus sprendimus, periodiškai peržiūrima ir atnaujinama diagrama, įvertinant naujas aplinkybes ir išvalgas;

Ishikawa diagrama vizualiai atvaizduoja potencialias problemos priežastis, skatina holistinį požiūrį į problemą, kadangi apima daug (keturias – aštuonias) skirtingų sričių (Korecki & Adámková, 2020). Ji yra universali ir gali būti pritaikoma kone kiekvieno pramonės sektoriaus specifikai. Ishikawa diagrama ypač naudinga siekiant sugeneruoti platų potencialių problemos priežasčių sąrašą. Vis dėlto Ishikawa diagramoje identifikuotoms problemos priežastims gali pritrūkti gylio ir faktais paremtos analizės (Bilsel & Lin, 2012). Siekiant pašalinti šį trūkumą, Ishikawa diagrama turėtų būti naudojama kartu su kitais analizės metodais, tokiais kaip „5 kodėl“ (Wolniak, 2019).

2.3.4. „5 kodėl“ metodas

„5 kodėl“ metodas – tai efektyvi problemų sprendimo technika, kurios esmė yra nuosekliai klausti „kodėl?“ tam, kad būtų nustatyta pagrindinė problemos priežastis. Šį metodą sukūrė Sakichi'is Toyoda, ir jis tapo neatsiejama „Toyota Motor Corporation“ problemų sprendimo praktikos dalimi. Nepaisant paprasto metodo pobūdžio, jo galia slypi šakninės problemos atskleidime. Vietoje to, kad būtų sprendžiami tik matomi problemos simptomai, metodas skatina ieškoti gilesnių priežasčių, kurios slypi už akivaizdžių požymių. Kiekvienas klausimas „kodėl?“ padeda pereiti prie kito problemos sluoksnio, kol pasiekama jos esmė. Metodo pakartojimų skaičius gali būti keičiamas priklausomai nuo situacijos, svarbiausia išlaikyti metodo esmę, klausti „kodėl?“ tol, kol pasiekama šakninė problemos priežastis. Šio metodo taikymo pagrindinis privalumas – išsprendžiamos simptomų priežastys, tokiu būdu užkertant kelią problemos pasikartojimui ateityje (Wolniak & Grebski, 2023).

Šioje pastraipoje pateikiamas „5 kodėl“ metodo taikymo pavyzdys programinės įrangos kūrimo įmonėje. Ši įmonė susidūrė su pasikartojančiomis klaidomis savo programoje, dėl kurių klientai buvo nepatenkinti. Taikant „5 kodėl“ metodą buvo nustatyta, kad klaidos atsiranda dėl nepakankamo sistemos testavimo. Toliau gilinantis buvo išsiaiškinta, kad testavimas buvo skubotas dėl griežtų terminų. Ši įžvalga paskatino įmonę peržiūrėti projektų tvarkaraščius ir skirti daugiau laiko testavimui, taip pat įdiegti griežtesnį kokybės užtikrinimo procesą, siekiant aptikti klaidas prieš joms pasiekiant klientus. Dėl siūlomų naujų priemonių pritaikymo klaidų skaičius žymiai sumažėjo, o klientų pasitikėjimas įmonės produktais pagerėjo (PMWorld 360, 2024). Apibendrinant galima teigti, kad problemų sprendimo procesas, orientuotas į priežasčių analizę, yra neatsiejamas nuo ilgalaikio efektyvumo ir organizacijos tobulėjimo užtikrinimo. Tokiu būdu sukuriama sprendimai, kurie ne tik pašalina esamus trūkumus, bet ir sustiprina sistemų atsparumą ateities iššūkiams.

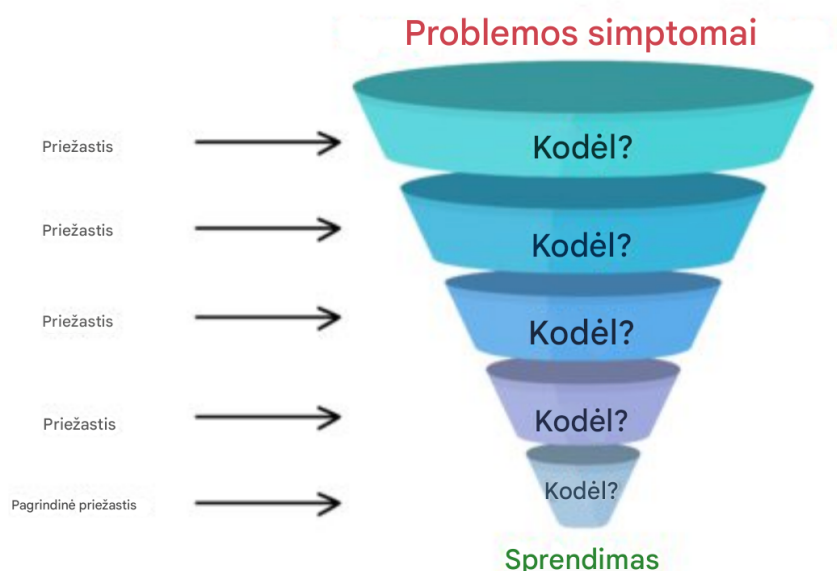
Pagal Wolniak'ą ir Grebski (2023) išskiriami sekantys „5 kodėl“ metodo naudojimo principai:

- **šakninės problemos identifikavimas.** Šis metodas nukreipia dėmesį nuo problemos simptomų į šakninę priežastį, atliekant giluminę problemos analizę;
- **struktūruotas požiūris.** Metodas remiasi struktūruotu požiūriu į problemų sprendimą. Keliant klausimus „kodėl?“, komandos metodiškai analizuoja problemą, identifikuoja tarpusavyje susijusius veiksnius ir geriau supranta situaciją;
- **pakartotinė analizė.** Šis metodas pasižymi pakartotiniu klausimo „kodėl?“ uždavimu, siekiant išryškinti skirtingus problemos aspektus. Kiekvienas sekantis klausimas leidžia geriau suprasti problemos struktūrą ir gilumines priežastis;
- **problemos pasikartojimo prevencija.** Šis metodas ne tik sprendžia esamą problemą, jis taip pat padeda sukurti sprendimus, kurie užtikrintų, kad panašios problemos nepasikartotų ateityje.
- **nuolatinis tobulėjimas.** „5 kodėl“ metodas skatina nuolatinį mokymąsi ir tobulėjimą, analizuojant gilumines problemų priežastis;
- **atviras požiūris.** Šis metodas skatina problemų sprendimą be išankstinių nusistatymų, kadangi gilinamasi į šaknines priežastis. Vengiant išankstinių prielaidų skatinama į problemos esmę pažvelgti nešališkai ir atrasti naujas įžvalgas;
- **komandinis įsitraukimas.** Šis metodas išryškina bendradarbiavimo svarbą, kadangi jo taikymo metu komandos dirba drauge, kartu ieško problemos priežasčių. Bendras darbas leidžia pasiekti geresnių rezultatų, atskleisti paslėptas priežastis, kadangi grupėje dažniausiai būna žmonių, kurie problemą mato iš skirtingų pozicijų;
- **prisitaikymas prie aplinkybių.** „5 kodėl“ metodas yra labai universalus ir gali būti plačiai naudojamas skirtingose pramonės šakose ir problemų sprendimo scenarijuose.

Pagal „PMWorld 360“ (2024), išskiriami keturi pagrindiniai metodo taikymo etapai:

1. **Problemos identifikavimas.** Svarbiausias pirmas žingsnis – aiškiai įvardinti sprendžiamą problemą. Tai gali būti pasikartojanti problema, pavyzdžiui klientų nepasitenkinimas paslaugomis arba silpnoji procesų grandinės dalis;
2. **Klausimas „kodėl?“.** Užduodamas klausimas, dėl ko ši problema atsirado ir užrašoma pirmoji problemos priežastis. Tuomet pradedama gilintis į sekančias priežastis, kiekvienam atsakymui vėl užduodant klausimą „kodėl?“. Procesas tęsiamas tol, kol pasiekama pagrindinė problemos priežastis. Taikant šį metodą dažniausiai pakanka penkių pakartojimų, vis dėlto priklausomai nuo problemos sudėtingumo, gali tekti metodą šiek tiek adaptuoti;
3. **Šakninės priežasties analizė.** Kai šakninė priežastis identifikuota, pradedama jos analizė, siekiant suprasti, kaip ji prisideda prie problemos. Analizė padeda sukurti praktinius sprendimus pagrindinei problemos priežastčiai pašalinti;
4. **Sprendimų įgyvendinimas.** Identifikavus šakninę priežastį, kuriami ir įgyvendinami korekciniai veiksmai, kurie pašalina problemos šaknį ir užkerta kelią jos pasikartojimui.

„5 kodėl“ metodo taikymas vizualiai vaizduojamas 8 paveiksle. Kaip ir aprašyta tekste, pirmiausiai pradedama nuo problemos ir jos simptomų identifikavimo ir užduodamas pirmasis „kodėl?“ klausimas. Taip kartojama penkis kartus, kol randama problemos šakninė priežastis.



8 pav. „5 kodėl“ metodo taikymo iliustracija (PMWorld 360, 2024)

Nors „5 kodėl“ metodas yra plačiai pripažintas šakninės priežasties analizės įrankis, jis turi keletą reikšmingų apribojimų:

- **per didelis supaprastinimas.** Taikant metodą sudėtingos ir kompleksinės problemos dažnai supaprastinamos iki vienos pagrindinės priežasties, nors iš tikrųjų jos kyla dėl keletos veiksnių (Yugay et al., 2021). Kelis organizacijos lygmenis apimančioms problemoms spręsti „5 kodėl“ metodas gali būti nepakankamas – metodas neatspindi tarpusavyje susijusių veiksnių ar sisteminių problemų. Tai gali nulemti ribotus sprendimus ir problemos pasikartojimą ateityje (Barsalou et al., 2024).
- **klaidingas metodo taikymas.** Pastebima atvejų, kai šis metodas naudojamas kaip idėjų generavimo įrankis. Metodas taikomas skirtingai įvairiose pramonės šakose ir situacijose – vienur jis naudojamas tik idėjų generavimui, kitur – sujungiamas su empiriniais duomenimis. To

rezultatas – paviršutiniški atsakymai, kurie neatskleidžia gilesnių priežasčių ir sisteminių problemų. Dėl to labai svarbu kiekvieną atsakymą pagrįsti faktais, ne tik dalyvių nuomone (Barsalou et al., 2024).

- **problemos gylio suvokimas.** Iššūkis ieškant šakninės priežasties – suprasti, kada ji atrandama. Per greitas sprendimo priėmimas gali neatskleisti tikrosios priežasties, vis dėlto per ilgai tęsiant sesiją, galima nukrypti į smulkmeniškumus (Yugay et al., 2021).
- **žmogiškųjų veiksnių įtaka.** Taikant šį metodą gali kilti pasipriešinimas iš komandos narių, ypač jeigu analizuojamos su darbuotojų klaidomis susijusios problemos arba kyla interesų konfliktas. Iš to kyla pasipriešinimas pokyčiams, kurie numatomi siekiant išspręsti problemą (Gangidi, 2019).

Išanalizavus „5 kodėl“ metodo esmę, galima teigti, jog šis metodas yra tinkamas paprastų ir vidutinio sudėtingumo problemų sprendimui, kurios neapima daug kompleksinių ir persipynusių aspektų. Šio metodo pritaikymas itin platus – nuo statybų sektoriaus iki programinės įrangos kūrimo. Statybų projektuose metodas gali būti panaudotas vėlavimo priežastims nustatyti (Shen et al., 2017), gamybos sektoriuje – identifikuojant defektų priežastis arba mažinant švaistymą (Benjamin et al., 2015). „5 kodėl“ metodas pritaikomas trikdžių analizei programinės įrangos kūrimo projektuose, siekiant sutrumpinti gedimų analizės laiką (Kataoka et al., 2011). Galiausiai, šis metodas gali būti taikomas ir buityje sprendžiant kasdienes problemas, pavyzdžiui, aptinkant buitinių prietaisų gedimus (Keyser & Pooyan, 2023). Taigi, remiantis moksline literatūra, „5 kodėl“ metodas gali būti tinkamas reikalavimų inžinerijos procesų tobulinimui dėl sukuriamos galimybės atrasti šaknines problemas priežastis.

2.3.5. SCAMPER metodas

SCAMPER (angl. *Substitute, Combine, Adapt, Modify, Put to other use, Eliminate, Rearrange*) yra kūrybiško mąstymo metodas, skirtas idėjų generavimui ir leidžiantis kūrybiškai pažiūrėti į problemą. Šį metodą sukūrė Alex'as Osborn'as, smegenų šturmo (angl. *brainstorm*) kūrėjas ir vėliau išvystė Bob'as Eberle'is. SCAMPER veikia kaip idėjų generavimo įrankis, skatinantis nukrypti nuo įprasto loginio mąstymo ir generuoti neįprastus sprendimus. Skirtingai nei smegenų šturmas ar minčių žemėlapiai, SCAMPER yra struktūruotas idėjų generavimo metodas. Jis leidžia sistemingai kurti idėjas pagal iš anksto parengtus klausimus. Taikant SCAMPER iš anksto suformuluojamos galimos problemos ar situacijos ir nuosekliai analizuojamos (Boonpracha et al., 2023).

SCAMPER metodo pavadinimas yra akronimas, kurio kiekviena raidė reiškia tam tikrą kūrybinės analizės veiksmą. Pagal Elmansy'į (2015) sudarytame sąrašė pateikiami veiksmai ir į procesų tobulinimą orientuoti klausimai:

- **pakeisti (angl. *substitute*).** Keisti tam tikrą produkto, proceso ar idėjos elementą kita alternatyva. Klausimai:
 - kurią proceso dalį galima pakeisti nepakenkiant visam projektui?
 - Ką galima pakeisti nepakenkiant procesui?
 - Kokios proceso dalys gali būti pakeistos geresnėmis alternatyvomis?
- **Sujungti (angl. *combine*).** Derinti skirtingus komponentus ar idėjas, siekiant naujų rezultatų. Klausimai:
 - ar galima sujungti du proceso žingsnius?
 - Ar galima vykdyti du procesus vienu metu?
 - Ar galima suderinti dvi technologijas?
- **Priderinti (angl. *adapt*).** Pritaikyti esamus sprendimus naujoms aplinkybėms. Klausimai:

- kaip padaryti procesą lankstesnį?
- Kokius kito sektoriaus procesus galima priderinti?
- **Modifikuoti/Padidinti/Sumažinti (angl. *modify/magnify/minify*).** Keisti arba padidinti/sumažinti produkto ar idėjos mastą, savybes ar funkcijas. Klausimai:
 - kaip proceso modifikavimas pakeis rezultatus?
 - Jeigu rinkta padidėtų, kaip atrodytų procesas?
 - Ar galima pakeisti procesą, kad jam reikėtų mažiau resursų?
- **Pritaikyti kitur (angl. *put to other use*).** Panaudoti produktą ar procesą naujam tikslui. Klausimai:
 - ar kiti padaliniai taip pat galėtų naudoti šitą procesą?
 - Ar galime pridėti papildomų žingsnių į procesą, kad jis būtų tinkamas kitoje situacijoje?
- **Pašalinti (angl. *eliminate*).** Atsisakyti nereikalingų komponentų, kad sistema taptų efektyvesnė. Klausimai:
 - ką būtų galima pašalinti iš proceso, kad jis taptų suprantamesnis?
 - Ar šis žingsnis reikalingas procese?
 - Ką reikėtų pašalinti iš proceso, jeigu turėtume dvigubai mažiau resursų?
- **Keisti tvarką (angl. *reverse/rearrange*).** Keisti komponentų tvarką ar struktūrą, ieškant naujų sprendimų. Klausimai:
 - kas būtų, jeigu proceso etapai būtų sukeisti vietomis?
 - Kaip galima sukeisti proceso žingsnius, kad būtų pasiektas geresnis rezultatas?

SCAMPER metodas yra vienas iš būdų efektyviai surasti skirtingus problemos sprendimus, neapsiribojant viena siaura sritimi – problemos sprendimo ieškoma septyniomis skirtingomis priemonėmis (Elmansy, 2015). SCAMPER metodas pasižymi lankstumu ir kūrybiškumo skatinimu – galima naudoti klausimus bet kokia tvarka, pakeisti vieną klausimą kitu (Boonpracha et al., 2023).

Apibendrinant pateiktus mokslinės literatūros analizės rezultatus ir darbo autoriaus įžvalgas apie procesų analizės ir tobulinimo metodus, galima teigti, jog procesų tobulinimas pradedamas nuo tikslo ir pagrindinių rodiklių apibrėžimo. Plačiai procesus pamatyti padeda vizualus atvaizdavimas, naudojant standartizuotus metodus, tokius kaip BPMN. Išdėdčius procesus vizualiai analizė tampa lengvesnė, išryškėja bendra procesų sąveika ir jų silpnosios vietos. Identifikavus silpnąsias procesų vietas, gali būti taikomi tokie metodai kaip „5 kodėl“ kartu su Ishikawa diagrama, siekiant nustatyti gilumines problemos priežastis. Problemų sprendimo paieškai gali būti pritaikytas SCAMPER metodas, kurio tikslas struktūruotai sugeneruoti daug idėjų, galinčių padėti išspręsti nustatytą problemą. Pašalinus šaknines priežastis, išnyskta ir problemos simptomai, taip pat užkertamas kelias problemos pasikartojimui. Siekiant valdyti procesus, kiekvienas procesas turi būti pamatuojamas jam pritaikytomis metrikomis – pagrindiniais veiklos rodikliais – KPI. Kadangi reikalavimų inžinerijoje skaitinių parametrų sutinkama nedaug, todėl tikslingiausia naudoti kokybinius KPI, kurie tam tikru būdu gali būti paversti skaitinėmis reikšmėmis, siekiant stebėti pažangą.

2.4. Reikalavimų surinkimo proceso tobulinimas

Analizuotuose moksliniuose straipsniuose ir reikalavimų inžinerijos iššūkių tyrimuose (Inayat et al., 2015; Rasheed et al., 2021; Yaseen et al., 2022) reikalavimų surinkimo procesas identifikuojamas kaip pats sudėtingiausias etapas, kuriame padaroma daugiausiai klaidų. Pagrindiniai iššūkiai siejami su neaiškiais, dviprasmiškais reikalavimais, skirtingais suinteresuotųjų šalių lūkesčiais bei komunikacijos trūkumu tarp klientų ir projekto komandos. Užsakovams jų pačių darbo procesai atrodo savaime

suprantami, dėl to dalis reikalavimų lieka neišsakyti, to rezultatas – sistemos trūkumai ir papildomos išlaidos. Atsižvelgiant į tai, tolesniame darbe didžioji dėmesio dalis skiriama reikalavimų surinkimo proceso analizei ir tobulinimo sprendimams. Šiame skyriuje nagrinėjami reikalavimų surinkimo metodai ir praktikos, išryškinami jų privalumai ir trūkumai. Siekiama nustatyti metodus, galinčius padėti sumažinti klaidų tikimybę, užtikrinti, kad surinkti reikalavimai kuo tiksliau atspindėtų esminius naudotojų poreikius.

2.4.1. Tradiciniai reikalavimų surinkimo metodai

Mokslinėje literatūroje įvardijami ir kaip reikalavimų inžinerijos pagrindas pripažįstami šie tradiciniai reikalavimų surinkimo metodai: dokumentų analizė, naudojimo atvejai (angl. *Use Cases*), interviu (Mesquita et al., 2023).

Dokumentų analizė – tai struktūruotas reikalavimų rinkimo metodas, pasižymintis dokumentų ir duomenų peržiūrėjimu, analizavimu, siekiant suprasti ir apibrėžti sistemos reikalavimus. Šis metodas naudingas projektuose, kuriuose yra sukaupta daug dokumentų, tekstinės medžiagos, tokių kaip ataskaitos, sistemos vadovai, specifikacijos. Dokumentų analizė gali būti naudojama tiek kuriant naują sistemą, kai keliama reikalavimai atitikti tam tikrus teisės aktus, tiek keičiant seną sistemą nauja ir siekiant užtikrinti, kad naujos sistemos funkcionalumas nepalikdamas spragų visiškai pakeistų senąją (Nicholas). Anot Mesquita'o ir kt. (2023), dokumentų analizė susiduria su tokiais iššūkiais kaip analizuojamų dokumentų nepatikimumas ir išsamumo trūkumas, nerašytos/užslėptos informacijos neįvertinimas, laikui imlus tinkamų dokumentų atrinkimas, šališkumas atrenkant dokumentus.

Naudojimo atvejai (angl. *Use Cases*) yra scenarijais pagrįsta technika, skirta rinkti funkciniais reikalavimams iš naudotojų perspektyvos – taikant šią techniką aprašoma vartotojų sąveika su sistema. Ši metodika dažnai taikoma norint aiškiai ir suprantamai aprašyti funkcinis reikalavimus (Ochodek et al., 2016). Kiekvienas atvejis pateikia žingsnius iš naudotojo perspektyvos, nurodant, kokius veiksmus jis atlieka ir kaip į tai reaguoja sistema, todėl taikant naudojimo atvejus dažnai atrandami užslėpti reikalavimai (Visual Paradigm, 2023a). Kuriant naudojimo atvejus naudojama paprasta, mažai techninių terminų turinti kalba, kurią lengvai supranta visos suinteresuotosios šalys. Naudojimo atvejų privalumas išryškėja sistemos testavimo metu, kadangi testuotojai gali identišškai atkurti realius sistemos naudojimo scenarijus (Visual Paradigm, 2023b). Naudojimo atvejų kūrimas mažina poreikį atskirai rašyti sistemos dokumentaciją, kadangi naudojimo atvejai išsamiai aprašo sistemos panaudojimo galimybes. Vis dėlto Mesquita'o ir kt. (2023) pabrėžia, jog kuriant sudėtingas sistemas naudojimo atvejų kūrimas ir palaikymas reikalauja neproporcingai daug laiko ir jų naudingumas mažėja.

Interviu metodas – tai pagrindinė, dažniausiai naudojama priemonė reikalavimų surinkimui, suteikianti galimybę perimti naudotojų skausmus, patirtis ir lūkesčius (Hegade, 2020). Interviu metodo pritaikymas gali būti įvairus – tiek kalbinant naudotojus individualiai, tiek kalbinant fokus grupes. Interviu procesas apima kelis pagrindinius etapus: pasirengimą, vykdymą bei surinktos informacijos patikrinimą. Šių etapų laikymasis padeda užtikrinti kokybišką reikalavimų surinkimą ir tinkamą jų interpretaciją (Nicholas; Test Bytes, 2024). Interviu pagal pobūdį skirstomi į struktūruotus, pusiau struktūruotus ir nestruktūruotus. Struktūruoti interviu turi iš anksto paruoštą klausimų rinkinį, kuris suteikia griežtą struktūrą. Atliekant pusiau struktūruotus interviu remiamasi paruoštais klausimais, tačiau taip pat leidžiama pokalbiui vystytis natūraliai, klausimus naudojant daugiau kaip gaires. Mažiausiai formalūs yra nestruktūruoti interviu – pokalbio metu palaikoma tik nustatyta interviu kryptis. Anot Mesquita'o ir kt. (2023) atlikto tyrimo, taikant interviu metodą kyla šios pagrindinės problemos:

- **neaiškūs klientų poreikiai.** Neretai klientai nėra visiškai tikri, ko jiems reikia, dėl to interviu metu pateikta informacija gali būti nepilna ir netiksli. To pasekoje reikalavimų surinkimas apsunksta ir suprojektuoti sistemą tampa sudėtingiau;
- **temos nukrypimas.** Interviu metu klientai yra linkę aptarti nereikšmingas temas, kurios nesusijusios su projekto tikslais ir dėl to apsunksta esminių reikalavimų surinkimas;
- **nepilni reikalavimai.** Klientai dažnai nenusako alternatyvių scenarijų, nupasakoja per siaurai. Dėl to vėlesniuose projekto etapuose tenka atlikti sistemos pakeitimus. Klientų prašymai neretai neatitinka tikrųjų poreikių, todėl kyla problemų identifikuojant, ką klientas iš tikrųjų nori pasiekti;
- **kliento techninių žinių stoka.** Daugelis klientų neturi pakankamų žinių apie informacines sistemas, todėl negali tinkamai išreikšti savo poreikių techniniu lygiu. Tai apsunkina bendravimą ir gali lemti neteisingai interpretuotus reikalavimus.

Apžvelgus tradicinius reikalavimų surinkimo metodus apibendrinant galima teigti, kad jie visi turi tiek trūkumų, tiek privalumų. Pavyzdžiui, mokslinės literatūros analizės metu išryškėjo, jog interviu metu surenkami reikalavimai gali būti nepilni, kliento poreikiai ne iki galo aiškūs. Vis dėlto mokslinėje literatūroje pabrėžiama, jog interviu metodas yra tinkamas išsamiam naudotojų patirties ir skausmų surinkimui. Tad galima daryti prielaidą, jog derinant interviu metodą su kūrybiniais reikalavimų surinkimo metodais, galima panaudoti skirtingų metodų privalumus, tokiu būdu sukuriant aiškesnius reikalavimus. Ši prielaida skatina empiriškai ištirti skirtingų metodų derinimo naudą reikalavimų surinkimo procese.

2.4.2. Kūrybiniai reikalavimų surinkimo metodai

Kūrybinis reikalavimų rinkimas – tai procesas, kurio metu naudojamos įvairios kūrybiškumą skatinančios technikos, praplečiančios mąstymo ribas ir skatinančios į problemą pažvelgti iš skirtingų pusių. Kūrybinės technikos leidžia išnaudoti žmogaus kūrybiškumą, sprendžiant sudėtingas, kelių lygių problemas. Kūrybinių reikalavimų rinkimo tikslas – išnaudojant grupinių sesijų privalumus į problemą pažvelgti iš įvairių perspektyvų (Mich, 2021).

Smegenų šturmas (angl. *brainstorming*) – plačiai žinoma technika, naudojama siekiant sugeneruoti platų idėjų ir potencialių reikalavimų sąrašą. Šis metodas skatina kūrybiškumą ir idėjų srautą, suteikiantį naują požiūrio kampą (Ribeiro et al., 2014). Smegenų šturmas skatina įsitraukti visus grupinės sesijos dalyvius ir apsvarstyti skirtingas perspektyvas. Grupinis įsitraukimas leidžia sukurti išsamesnius ir įvairesnius reikalavimus (Ribeiro et al., 2014). Greta metodo privalumų, pažymimi šie smegenų šturmo trūkumai: dėl metodo struktūros trūkumo praleidžiami kai kurie esminiai aspektai, sugeneruotos nepamatuotos idėjos gali būti sunkiai įgyvendinamos (Khade et al., 2022).

Šešios skrybėlės (angl. *Six Thinking Hats*) – tai Edward'o de Bono sukurtas kūrybiškumą skatinantis metodas, kuris remiasi lygiagrečiu mąstymu. Šis metodas struktūrizuoja mąstymo procesą, suskirstydamas jį į šešias skirtingas perspektyvas, atstovaujamas skirtingų spalvų skrybėlių. Diskusijos metu dalyviai užsideda arba nusiima tam tikros spalvos skrybėlę, taip nurodydami, kokį mąstymo tipą naudoja. Kiekvienos skrybėlės spalvos reikšmės pateikiamos sąrašė (Ribeiro et al., 2014):

- **balta skrybėlė.** Orientuota į faktus ir skaičius, skatina informaciją perteikti objektyviai, be išankstinio nusistatymo ar interpretacijų. Jos tikslas – išgauti neutralią informaciją;
- **raudona skrybėlė.** Orientuota į emocijas ir jausmus, leidžia išsakyti nuomonę be argumentų ar pasiteisinimų, priešingai nei balta skrybėlė;

- **juoda skrybėlė.** Pabrėžia galimus trūkumus ir rizikas, naudojama galimų klaidų, pavojų ir nesėkmių išryškinimui;
- **geltona skrybėlė.** Orientuojasi į teigiamus aspektus, skatina optimizmą ir analizuoja idėjos teikiamą naudą – atsveria juodos skrybėlės pesimizmą;
- **žalia skrybėlė.** Pabrėžia kūrybinį mąstymą, ragina ieškoti naujų idėjų ir alternatyvų, skatina peržengti įprastų sprendimų ribas;
- **mėlyna skrybėlė.** Orientuojasi į bendrą problemos vaizdą apibendrinant diskusiją, priimant sprendimus ir struktūrizuojant procesą.

Šešių skrybėlių metodas ypač naudingas diskusijoms, kai siekiama išvengti konfliktų, užtikrinti visų nuomonių išsklausymą. Anot Ribeiro ir kt. (2014) atlikto tyrimo, šešių skrybėlių metodo taikymo rezultatai reikalavimų rinkimo procese parodė šio metodo efektyvumą – padidėjo dalyvių įsitraukimas, motyvacija ir buvo gauti geresnės kokybės reikalavimai. Diskusijos leido dalyviams išsamiai analizuoti idėjas iš įvairių perspektyvų: nuo teigiamų ir neigiamų aspektų iki kūrybiškų sprendimų, o surinkti reikalavimai pagal kokybę prilygo arba viršijo renkamus tradiciniais būdais. Toks formatas taip pat padėjo stiprinti dalyvių ryšius ir gerinti santykį.

Dirbtuvės (angl. *workshop*) – tai plačiai žinoma, efektyvi technika reikalavimų surinkimui. Dirbtuvėse sukuriamas bendradarbiavimo aplinka, kurioje susitikę suinteresuotos šalys gali išdiskutuoti ir apibrėžti kuriamos sistemos reikalavimus. Dirbtuvėse dažnai rengiamos interaktyvios sesijos, kuriose naudojamos įvairios technikos, tokios kaip smegenų šturmas, prototipavimas. Lipnių lapelių (angl. *sticky note*) diagramos yra dažnas įrankis vizualizavimui ir informacijos sisteminiui. Tam naudojamos tokios technikos kaip įvykių modeliavimas (angl. *Event Storming*) ir naudotojo patirties žemėlapis (angl. *User Story Mapping*), kurios skirtos fasilituoti bendras diskusijas (Gardner et al., 2020). Grupinio darbo technikos yra dažnai naudojamos reikalavimų surinkimui dėl suteikiamų bendradarbiavimo privalumų, tačiau siūloma tokias priemones derinti su tradiciniais reikalavimų surinkimo metodais, tokiais kaip interviu (Mesquita et al., 2023).

Įvykių modeliavimas (angl. *Event Storming*) – tai vizualus ir bendradarbiavimu pagrįstas domeno modeliavimas, kurį sukūrė italas, programinės įrangos ekspertas Alberto'as Brandolini's (Brandolini, 2021). Metodo autorius yra stiprus domenu grįsto dizaino (angl. *Domain-Driven Design*) ir „Agile“ metodologijų šalininkas, kurios padėjo pagrindą įvykių modeliavimo metodo sukūrimui 2013 metais. Iš pradžių jis buvo skirtas greitai modeliuoti sudėtingus verslo procesus remiantis domenu grįsto dizaino principais, tačiau vėliau šis metodas itin plačiai paplito procesų ir programinės įrangos modeliavimui. Šis metodas prisidėjo prie domenu grįsto dizaino atgimimo ir padėjo išpopuliarinti modeliavimo bendradarbiaujant (angl. *Collaborative Modelling*) judėjimą.

Analizuojant mokslinę literatūrą (Kasauli et al., 2021; Mendez & Moreira, 2024; Rasheed et al., 2021; Copei et al., 2022; Grimm & Rubart, 2021; Gardner et al., 2020; Brandolini, 2021; Palumbo et al., 2024; Alazzawi et al., 2023) nustatyta, kad **įvykių modeliavimo** metodas skirtas efektyviam verslo procesų supratimui ir sprendžia dažnai pasitaikančias reikalavimų inžinerijos problemas:

- **skirtingas domeno supratimas.** Suinteresuotosios šalys dažnai turi skirtingą supratimą apie sistemą ir reikalavimus jai (Kasauli et al., 2021; Mendez & Moreira, 2024; Rasheed et al., 2021). Įvykių modeliavimo (angl. *Event Storming*) metodas suburia skirtingas šalis ir leidžia kartu apibrėžti visą procesą, tokiu būdu suvienodinant visų dalyvių suvokimą (Copei et al., 2022);
- **ilgas ir neefektyvus reikalavimų rinkimas.** Esant neapibrėžtiems reikalavimams, tradiciniai reikalavimų rinkimo metodai gali būti lėti ir neefektyvūs. Įvykių modeliavimo (angl. *Event*

Storming) metodas leidžia greitai išgryninti esminius reikalavimus interaktyviose sesijose (Grimm & Rubart, 2021);

- **vizualizavimo stoka.** Įvykių modeliavimo (angl. *Event Storming*) metodas suteikia priemones sudėtingų sistemos procesų vizualizavimui. Tai leidžia sukurti holistinį supratimą apie visą sistemą, nepraleidžiant svarbių detalių (Gardner et al., 2020; Grimm & Rubart, 2021);
- **bendradarbiavimo trūkumas.** Neretai IT projektuose kyla komunikacijos iššūkių tarp klientų ir projekto komandos. Įvykių modeliavimo (angl. *Event Storming*) metodas skatina visų suinteresuotų šalių bendradarbiavimą, suteikiant galimybę dalyvauti modeliuojant ir išsakant pastabas (Copei et al., 2022);
- **netinkamai interpretuoti verslo reikalavimai.** Verslo taisyklės ir logika neretai būna ne iki galo išgryninamos. Įvykių modeliavimo (angl. *Event Storming*) metodas leidžia identifikuoti ir išsiaiškinti verslo taisykles sesijos metu diskutuojant su domeno ekspertais (Brandolini, 2021; Palumbo et al., 2024);
- **problemos identifikuojamos per vėlai.** Dažnai problemos ir nesusipratimai išaiškėja tik vėlesniuose projekto etapuose, kai pataisymai kainuoja brangiau. Įvykių modeliavimas (angl. *Event Storming*) padeda anksti identifikuoti problemas ir spragas, kol dar nepradėtas sistemos kūrimas. Tokiu būdu sumažinama rizika ir galimos išlaidos (Alazzawi et al., 2023; Brandolini, 2021);
- **dėmesys dokumentacijai.** Per daug detali dokumentacija reikalavimų rinkimo pradžioje gali apsunkinti darbą ir sistemos suvokimą. Įvykių modeliavimas (angl. *Event Storming*) sukuria vizualų ir lengvai suprantamą proceso žemėlapi, kuris iš pradžių gali veikti kaip pagrindinė dokumentacija (Gardner et al., 2020).

Išanalizavus mokslinę literatūrą identifikuoti pagrindiniai įvykių modeliavimo (angl. *Event Storming*) metodo privalumai:

- **greitis.** Taikant šį metodą gerokai sumažinamas laikas, per kur sukuriamas išsamus verslo domeno supratimas – tam gali pakakti ir keletos valandų interaktyvios sesijos (Brandolini, 2021);
- **intuityvumas.** Vietoje sudėtingų UML diagramų, įvykių modeliavimo (angl. *Event Storming*) metodas procesą suskaido į aiškias ir suprantamas dalis, kurias gali lengvai perprasti tiek techniniai, tiek netechniniai dalyviai. Dėl tokio neformalaus proceso visi dalyviai atsiduria tame pačiame lygyje, todėl suinteresuotųjų šalių dalyvavimas ir bendradarbiavimas tampa sklandesnis (Gardner et al., 2020; Grimm & Rubart, 2021);
- **įsitraukimas ir komunikacija.** Anot metodo kūrėjo, vienas iš įvykių modeliavimo (angl. *Event Storming*) tikslų – padaryti sistemos modeliavimą įdomų, skatinti bendradarbiavimą. Taikant šį metodą nelieka neįsitraukusių dalyvių – skatinamos diskusijos, o jų metu sukuriamas bendras supratimas. Fasilitatorius skatina dalyvius aktyviai dalintis savo mintimis ir požiūriu, todėl gaunamos vertingos įžvalgos, kurios fiksuojamos ant lipnių lapelių ir dedamos į bendrą paveikslą – tokiu būdu išryškėja sistemos dalių sąsajos (Copei et al., 2022; Gardner et al., 2020);
- **efektyvumas.** Įvykių modeliavimas (angl. *Event Storming*) vienos sesijos metu sukuria pilną sistemos veikimo principo modelį. Šis sistemos modeliavimo metodas labai gerai dera su domenu grįsto dizaino principais (Brandolini, 2021).

Įvykių modeliavimo (angl. *Event Storming*) metodas taikomas bendruose klientų ir sistemos vystytojų susitikimuose. Nors šiuo metu yra keletas įvykių modeliavimo (angl. *Event Storming*) metodo atšakų, tačiau pasak metodo autoriaus Brandolini'o (2021), tradicinė veiklų seka yra tokia:

- **pasiruošimas.** Į patalpą be kėdžių susirenka sistemos užsakovai ir vystytojai. Toks formatas pasirenkamas dėl kūrybiškumo ir aktyvumo skatinimo. Ant vienos iš patalpos sienų ištiesiamas popieriaus rulonas, kurio ilgis kartais gali siekti net 18 metrų, priklausomai nuo sistemos sudėtingumo. Ant ištiesto popieriaus klijuojami atitinkamas reikšmes turintys lapeliai su užrašais;
- **domeno įvykių fiksavimas.** Pagal eilės tvarką identifikuojami sistemos įvykiai ir užrašomi ant oranžinės spalvos lapelių būtuoju laiku, pavyzdžiui – prekė įdėta į krepšelį, užsakymas priimtas, mokėjimas atliktas;
- **riziką keliančių vietų identifikavimas.** Nustatomi riziką keliantys aspektai, kuriuose dar trūksta informacijos. Dažnu atveju tai būna išorinės sistemos, kurių veikimui negalima padaryti įtakos, atsiradus jų veikimo trikdžiams, sutriktų ir kuriama sistema (Brandolini, 2021). Pavyzdys galėtų būti elektroninė parduotuvė – dažniausiu atveju atsiskaitymas už prekes ar paslaugas internetu būna paliktas kaip trečiosios šalies atsakomybė dėl saugumo užtikrinimo. Taigi, neveikianti atsiskaitymo sistema neleidžia vartotojams atsiskaityti už prekes ar paslaugas internetinėje parduotuvėje ir sukuria blogą naujos sistemos reputaciją;
- **komandų ir veikėjų identifikavimas.** Nustatoma, kas ir kokius veiksmus atlieka kuriamoje sistemoje. Identifikuoti veikėjai ir jų komandos padeda lengviau nustatyti, kokie verslo procesai vyksta ir kas juos inicijuoja sistemoje. Visų pirma identifikuojamos komandos – sesijos dalyviai ant mėlynų lapelių užrašo, kokios komandos yra atliekamos. Komanda užrašoma bendratimi, pavyzdžiui – pateikti užsakymą, atšaukti prenumeratą, išsiųsti patvirtinimą. Šalia kiekvienos komandos priklijuojamas geltonos spalvos lapelis su veikėju, kuris tą komandą iškviečia – klientas, administratorius, išorinė mokėjimų sistema;
- **taisyklių nustatymas.** Apibrėžiamos taisyklės kaip sistema turėtų reaguoti į tam tikrus įvykius ir situacijas. Taisyklės yra reikalingos tam, kad sistemos veikimas būtų nuoseklus ir atitiktų verslo poreikius ir reikalavimus. Ant rožinių lapelių užrašytos taisyklės užrašomos priklijuojamos šalia apibrėžiamų įvykių, o sudėtingesnės vietos pažymimos violetiniais lapeliais.
- **struktūrizavimas ir agregatų nustatymas.** Apibrėžiami pagrindiniai sistemos duomenų modeliai, kurie apjungia duomenis ir veiksmus. Iš kelių modelių susidedantys agregatai padeda išlaikyti atskiras dalis, tuo pačiu jas apjungiant ir valdant sudėtingumą. Agregatų pavyzdžiai – užsakymas, klientas, produktas, atsiskaitymas. Kiekvienam agregatui priskiriama atsakomybė už tam tikrų įvykių ir veiksmų valdymą, jie pažymimi geltonais lapeliais. Tuomet identifikuojami ryšiai ir priklausomybės tarp agregatų. Šio etapo rezultatas – galutinis sistemos žemėlapis .
- **diskusija ir išvados.** Aptariami rezultatai, problemos, numatomi sekantys veiksmai.

Anot įvykių modeliavimo metodo autoriaus Brandolini'o (2021), sėkmingas įvykių modeliavimo (angl. *Event Storming*) taikymas tiesiogiai priklauso nuo sesijos fasilitatoriaus įgūdžių. Šis asmuo turi mokėti pakreipti diskusijas konstruktyvia kryptimi, valdyti laiką, užtikrinti, kad nebūtų pernelyg nukrypta į smulkmeniškumus. Kaip teigia autorius, metodas labiausiai pritaikomas mažiems ir vidutinio dydžio projektams, išlaikant valdomą komandos dydį.

Išanalizavus mokslinėje literatūroje aptariamo įvykių modeliavimo metodo esmę, apibendrinant galima teigti, kad įvykių modeliavimas – tai metodas, kuris padeda greitai išskaidyti sudėtingas sistemas į elementarius įvykius ir sukurti bendrą sistemos žemėlapi, kuris paaikšina sistemos funkcionalumą. Šis metodas įtraukia pagrindines suinteresuotąsias šalis ir pasitelkia vizualinius metodus, kad pagerintų komunikaciją ir užtikrintų išsamią reikalavimų analizę. Interaktyviose sesijose dalyvauja tiek verslo žmonės, tiek sistemų kūrėjai, todėl įgalinama tiesioginė komunikacija tarp pagrindinių suinteresuotųjų

šalių. Neformalus procesas ypač veiksmingai įtraukia netechninius dalyvius ir suburia skirtingų sričių ekspertus. Kadangi pagrindiniai suinteresuotieji asmenys kartu kuria sistemos modelį, probleminės vietos identifikuojamos gerokai anksčiau – bendradarbiavimas padeda užtikrinti, kad reikalavimų rinkimas būtų išsamus ir optimalus.

Apibendrinant tradicinių ir kūrybinių reikalavimų surinkimo metodų analizės rezultatus galima teigti, jog šie metodai pasižymi skirtingais privalumais, pavyzdžiui, interviu metodas yra orientuotas į problemos gylį suvokimą, tuo tarpu įvykių modeliavimo metodas – į bendro paveikslo sudarymą. Tad galima daryti prielaidą, jog šių metodų derinimas duotų geresnius rezultatus nei metodų taikymas atskirai.

2.5. Reikalavimų inžinerijos procesų tobulinimo IT projektuose teorinis modelis

Remiantis mokslinės literatūros analizės rezultatais buvo sudarytas teorinis konceptualus reikalavimų inžinerijos procesų tobulinimo IT projektuose modelis (9 pav.), kuris užduoda gaires tolesniam empiriniam tyrimui. Modelio centre atvaizduoti reikalavimų inžinerijos procesai per visą sistemos kūrimo gyvavimo ciklą – nuo reikalavimų surinkimo iki reikalavimų valdymo. Šie procesai vaizduojami su grįžtamosiomis rodyklėmis, kadangi jie nors ir turi tekmės kryptį, tačiau yra iteraciniai, pavyzdžiui iš reikalavimų analizės gali būti grįžtama į surinkimą ir panašiai.



9 pav. Reikalavimų inžinerijos procesų tobulinimo IT projektuose konceptualus teorinis modelis

Iš reikalavimų inžinerijos procesų kyla problema, kuri suskaidyta į dvi dalis: iššūkiai ir poreikis tobulinti reikalavimų inžinerijos procesus. Esminis tobulinimo poreikis – pagerinti projektų sėkmės rodiklius ir sumažinti sąnaudas. Remiantis atlikta mokslinės literatūros analize, išskiriamas reikalavimų inžinerijos procesų tobulinimas, kuris susideda iš kūrybinių ir tradicinių reikalavimų surinkimo metodų derinimo, pagrindinių veiklos rodiklių (angl. *KPI*) nustatymo ir procesų tobulinimo metodų. Kaip ir bet kokio proceso tobulinime, ypatingai svarbu nustatyti pagrindines metrikas, kurios nusako, ar teisinga linkme daromi pakeitimai, ar gaunamas norimas rezultatas. Modelyje atvaizduoti literatūros analizėje paminėti procesų tobulinimo metodai, tokie kaip „5 kodėl“, Ishikawa diagrama, SCAMPER, vizualizavimas, kurie galėtų būti panaudoti reikalavimų inžinerijos procesų tobulinimui. Atliekant literatūros analizę buvo pastebėta, jog keliuose šaltiniuose paminėtas kūrybinių ir tradicinių reikalavimų rinkimo metodų derinimas kaip tolesnė tyrimų kryptis, todėl daroma prielaida, jog tai padėtų tobulinti reikalavimų inžinerijos procesus. Modelyje paminėti atitinkami kūrybiniai reikalavimų surinkimo metodai: smegenų šturmas, dirbtuvės, įvykių modeliavimas, šešios skrybėlės bei tradiciniai, tokie kaip dokumentų analizė, interviu, naudojimo atvejai.

3. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo metodologija

Tyrimo klausimas

Kaip tobulinti reikalavimų inžinerijos procesus IT projektuose?

Tyrimo objektas

Reikalavimų inžinerijos procesų tobulinimas.

Tyrimo tikslas

Atskleisti reikalavimų inžinerijos procesų tobulinimo metodų taikymą praktinėje veikloje.

Uždaviniai

1. Identifikuoti reikalavimų inžinerijos procesų iššūkius IT projektuose;
2. Nustatyti reikalavimų inžinerijos procesų vertinimo metrikas IT projektuose;
3. Atskleisti reikalavimų inžinerijos procesų tobulinimo metodus, taikomus IT projektuose;
4. Ištirti ekspertų taikomus reikalavimų surinkimo metodus IT projektuose.

Pasirinktas tyrimo metodas – pusiau struktūruotas kokybinis interviu. Kokybinis interviu – tai metodas, kuris suteikia galimybę giliau pažvelgti į tyrimo dalyvių patirtis, išgirsti jų nuomones ir suprasti jausmus. Pusiau struktūruotas interviu turi gerokai daugiau lankstumo nei kiekybiniuose tyrimo metoduose taikomas struktūruotas interviu, todėl yra labiau tinkamas surinkti dalyvių patirčiai. Pusiau struktūruotas interviu pasižymi šiais bruožais: interaktyvus dialogas, teminis požiūris, tyrimo kontekstas. Tyrėjas interviu veda pagal iš anksto paruoštus klausimus, tačiau, priklausomai nuo susiklosčiusių aplinkybių, kai kurie klausimai gali būti praleidžiami. Pavyzdžiui, tyrimo dalyvis atsakydamas į pirmą klausimą, gali pateikti atsakymą ir į kitus tyrėjo paruoštus klausimus – tokiu atveju nebūtų prasmės klausimus užduoti pakartotinai. Taip pat interviu metu pagal dalyvių atsakymus, tyrėjui gali kilti tikslinamųjų klausimų, kurie nebuvo numatyti iš anksto. Dėl to svarbu interviu metu išlaikyti tam tikrą struktūrą, tuo pačiu išlaikant ir lankstumą (Edwards & Holland, 2013). Pusiau struktūruoto interviu metodo pasirinkimo tikslas – surinkti IT projektuose dirbančių verslo analitikų patirtį, susijusią su reikalavimų inžinerijos procesais. Siekiama išsiaiškinti, kokie reikalavimų procesų tobulinimo metodai pasiteisina praktinėje veikloje, kokios gerosios praktikos padeda sėkmingai užbaigti projektą. Taikant kiekybinius tyrimo metodus siekiama apklausti statistiškai reikšmingą respondentų skaičių, kad būtų pasiektas numatytas imties dydis ir būtų galima išvesti tam tikrą statistiką (Edwards & Holland, 2013). Tuo tarpu šio tyrimo, kaip ir kitų kokybinių tyrimų, tikslas – nuodugnus ir išsamus reiškinio įvertinimas, o tyrimo dalyvių atranka – patogioji tikslinė.

Pakankama tyrimo imtis kokybiniuose ir kiekybiniuose tyrimuose suprantama skirtingai. Šiuo atveju, kokybiniame tyrime, pakankama tyrimo imtis laikoma tada, kai pasiekiamas surinktų duomenų saturacija ir atsakoma į tyrimo klausimą. Kokybiniuose tyrimuose imties dydis labai glaudžiai susijęs su tyrimo kontekstu, nes nuo jo priklauso, kas bus tyrimo dalyviai ir kiek jų reikės, siekiant išsamiai atsakyti tyrimo klausimą. Kai kurie tyrėjai teigia, jog imties terminas nelabai tinka kokybiniams tyrimams, kadangi dėmesys skiriamas ne į kiekybinį rezultatą, o į duomenų generavimo procesą. Pagrindinis imties formavimo kokybiniuose tyrimuose tikslas – užtikrinti, kad surinkti duomenys leistų atsakyti į iškeltą tyrimo klausimą. Kokybiniuose tyrimuose dalyviai atrenkami pagal jų atitikimą tyrimo teoriniam pagrindui, kadangi tai leidžia atrinkti dalyvius, kurie turi realios, tyrimui naudingos patirties. Tyrėjas gali papildyti imtį ir interviu klausimus tyrimo eigoje, remiantis tarpine tyrimo duomenų analize, kylančiomis įžvalgomis. Tokia imtis vadinama tiksline, kadangi atrenkami tik tyrimo tikslą atitinkantys atvejai. Vienas iš būdų suformuoti tyrimo imtį – sniego gniūžtės metodas. Tyrimo pradžioje tyrėjas

užmezga ryšį su keletu tyrimo dalyvių per turimus ryšius ir kanalus, o po to pirmieji dalyviai rekomenduoja kitus asmenis, kurie turi tyrimui reikalingą patirtį ir galėtų papildyti tyrimo imtį. Akademiniuose tyrimuose dažniausiai naudojama patogioji atranka, kai tyrėjas pasirenka per turimus kanalus prieinamus tyrimo dalyvius (Edwards & Holland, 2013).

Tyrimo etika

Interviu respondentai atsakinėja laisva valia ir turi teisę bet kuriuo metu nutraukti pokalbį. Pašnekovai išlieka anonimais, o surinkta informacija nuasmeninama, kad pagal ją nebūtų įmanoma identifikuoti konkretaus asmens ar įmonės. Tyrėjas užtikrina tyrimo sąžiningumą, vengia šališkumo ir nemanipuliuoja duomenimis ar dalyviais.

Tyrimo imtis ir populiacija

Tyrimo populiaciją sudaro brandžių, IT projektus vystančių, turinčių padalinius Lietuvoje įmonių atstovai. Brandžios įmonės, tikėtina, turi nusistovėjusius darbo procesus ir skiria didelę dalį dėmesio procesų tobulinimui ir darbuotojų kvalifikacijos kėlimui, lyginant su jaunomis, ką tik įkurtomis įmonėmis. Tyrimo dalyviai – verslo analitikai – tai profesionalai, kurie IT projektuose palaiko ryšius su klientu ir daugiausiai dalyvauja reikalavimų inžinerijos procesuose. Tyrimo dalyvių atranka – patogioji netikimybinė, remiasi tikslingumo principu. Tyrėjas tikslingai pasirenka tyrimo dalyvius iš prieinamos aplinkos. Sekantiems interviu respondentams surasti naudojamas sniego gniūžtės metodas. Tokiu būdu siekiama gauti kuo įvairesnius duomenis, kurie padės atsakyti, kaip tobulinti reikalavimų inžinerijos procesus IT projektuose. Tyrimo imties dydis, kaip ir būdinga kokybiniais tyrimams, numatomas toks, jog iš gautų rezultatų būtų galima atsakyti tyrimo klausimą ir būtų pasiektas rezultatų prisotinimas. Interviu respondentų duomenys pateikiami 2 lentelėje.

2 lentelė. Interviu respondentų duomenys

Eksperto kodas	Organizacijos kodas	Eksperto pareigos
E1	O1	Vyr verslo analitikas
E2	O1	Vyr.verslo analitikas
E3	O1	Testavimo praktikų vadovas
E4	O1	Vyr. verslo analitikas
E5	O2	Vyr. verslo analitikas
E6	O3	Vyr. verslo analitikas

Vidutinė interviu trukmė 1 valanda.

Tyrimo dalyvavo ekspertai iš trijų skirtingų organizacijų, vykdančių IT projektus. Siekiant išlaikyti dalyvių ir įmonių anonimiškumą, nepateikiami organizacijų pavadinimai ir ekspertų vardai.

Pirmoji organizacija (O1) – daugiau kaip 30 metų veikianti, globali IT konsultacinių paslaugų įmonė, turinti virš 300 000 darbuotojų visame pasaulyje. Tyrimo dalyvavo Lietuvos padalinyje dirbantys ekspertai.

Antroji organizacija (O2) – Lietuvoje veikianti gamybinė įmonė, turinti IT projektų padalinį. Įmonė kuria savo programinę įrangą ir parduoda ją kaip galutinio produkto dalį. Įmonėje programinė įranga kuriama projektiniu principu.

Trečioji organizacija (O3) – globali IT konsultacinė įmonė, kuri specializuojasi valstybinės ir ekonominės skaitmeninės infrastruktūros kūrimo.

Tyrimo eiga

Tyrimo eiga susideda iš pasiruošimo tyrimui, interviu atlikimo, surinktų duomenų analizės ir išvadų pateikimo.

Pasiruošimo etape sudaromas interviu klausimynas (1 priedas) pagal atliktą mokslinės literatūros analizę ir darbo autoriaus parengtą konceptualų modelį (9 pav.). Paruošus klausimyną, pradedamas ryšių mezgimas su atrinktais tyrimo dalyviais, jie kviečiami dalyvauti tyrime, sutariama abiem pusėms tinkama data ir vieta. Kviečiant dalyvauti tyrime, dalyviai iš anksto informuojami apie tyrimo tikslą, eigą, preliminarines interviu temas, tyrimo anonimiškumą. Toks dalyvių informavimas leidžia jiems geriau pasiruošti ir nusiteikti.

Interviu atliekami su kiekvienu respondentu atskirai, naudojantis internetinėmis pokalbių platformomis, kurios suteikia galimybę pokalbį įrašyti. Interviu transkribuojami pasitelkiant transkribavimo programas ir aktualios citatos koreguojamos rankiniu būdu, siekiant gauti aukštos kokybės duomenis.

Duomenų analizė seka po interviu atlikimo ir pradedama dar nepasibaigus interviu atlikimo etapui, kadangi įžvalgos ir daliniai rezultatai gali padėti praturtinti sekančius interviu ir surinkti dar išsamesnius duomenis. Analizuojami interviu transkriptai kokybinės turinio analizės būdu, naudojantis programa „MAXQDA“, kuri leidžia grupuoti informaciją pagal temas, pažymėti reikšmingas citatas ir galiausiai duomenis susisteminti diagramomis.

Išvadų pateikimo etape koncentruotai pateikiami duomenų analizės rezultatai ir įžvalgos. Išvadose išryškinami pagrindiniai tyrimo rezultatai, nurodant metodus, kurie gali būti panaudoti, siekiant patobulinti reikalavimų inžinerijos procesus, remiantis ilgametę profesinę patirtį turinčių tyrimo dalyvių nuomone.

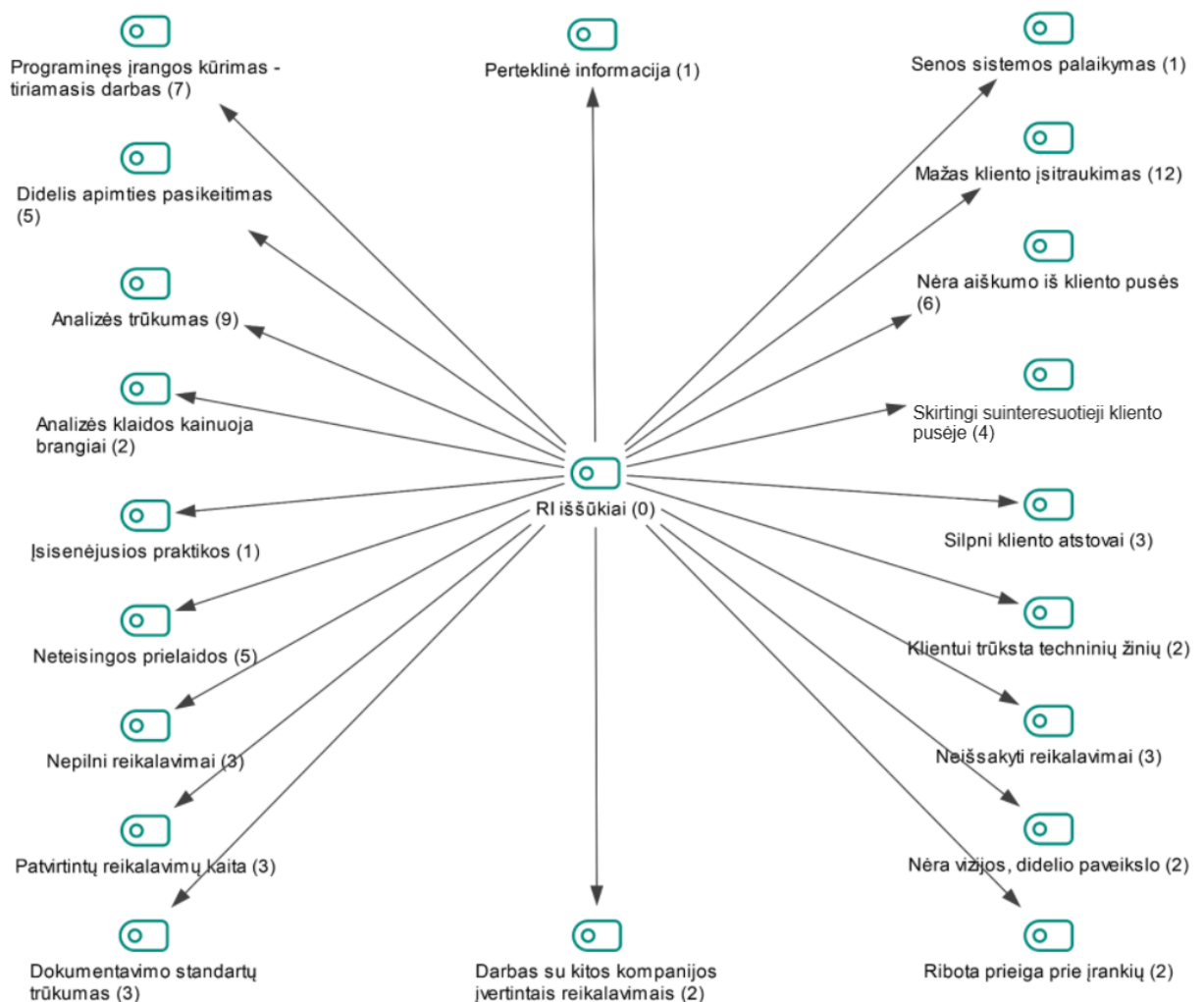
4. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo rezultatai ir diskusija

4.1. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo duomenų analizė

Empirinio tyrimo metu sudaryti interviu transkriptai buvo analizuojami kompiuterine programa „MAXQDA“, koduojant gautą informaciją ir grupuojant kodus į kategorijas. Kategorijos buvo struktūrizuojamos pagal išsikeltus empirinio tyrimo tikslus, remiantis tyrimo konceptualiuoju modeliu. Rezultatai atvaizduojami kodų–subkodų hierarchijos diagramomis, siekiant struktūruotai pateikti tyrimo metu surinktus duomenis. Tyrimo rezultatai skaidomi į keturis skyrius: reikalavimų inžinerijos iššūkiai IT projektuose, reikalavimų inžinerijos procesų vertinimo metrikos IT projektuose, reikalavimų inžinerijos procesų tobulinimo metodai IT projektuose ir ekspertų taikomi reikalavimų surinkimo metodai IT projektuose.

4.1.1 Reikalavimų inžinerijos iššūkiai IT projektuose

Šiame skyrelyje pateikiami empirinio tyrimo metu identifikuoti reikalavimų inžinerijos iššūkiai IT projektuose, su kuriais susiduria tyrime dalyvavę ekspertai ir jų komandos. 10 paveiksle pateikiamas vienas iš tyrimo artefaktų – reikalavimų inžinerijos iššūkių IT projektuose vizualinis modelis. Modelyje atvaizduojami 20 iššūkių, su kuriais susiduria reikalavimų inžinerijos ekspertai.



10 pav. Reikalavimų inžinerijos iššūkių IT projektuose vizualinis modelis

3 lentelėje pateikiamos reikalavimų inžinerijos iššūkių IT projektuose subkategorijos ir jas pagrindžiančios citatos. Kiekviena subkategorija išskiria identifikotą iššūkį.

3 lentelė. Reikalavimų inžinerijos iššūkių IT projektuose subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Nėra aiškumo iš kliento pusės	„ <i>Stakeholder'iai</i> <...> nežino, ko nori, ir <...> reikia <...> pasižiūrėti šitą procesą, kuris <...> primena <...> žaidimą vaikų darželyje, kai tu turi pats kažkokiu būdu sužaisti ir gauti atsakymą, kuris <...> neaišku, ar bus tinkamas.“ (E4); „<...> gali būti neaiškūs reikalavimai. Patys nesupranta, ko nori <...>, bet <...> vis tiek tai išsiverčia į tai, kad tavo tikslas yra ateiti ir išsiaiškinti. <...> klientai <...> galbūt neaiškiai <...> iškomunikuos, nes jie nežino, ko nori. <...>“ (E5); „<...> kiek žmogus pratęs kalbėti ir leisti į detales. Nes vieni tikrai būna žmonės, kurie labai aukštai idėjiniam lygy gali kalbėti <...>, ir tau reikia iš to <...> nusileisti žemiau, kad galėtum <...> įgyvendinti. <...> Tu gali interviu daryti su vienu žmogumi, kuris <...> visą laiką padėbesiais skraidžioja, o tau reikia jį <i>sugroundinti</i> .“ (E5); „Klientas <...> labai neapsisprendžiantis ir tamposi pirmyn atgal <...>. <...> Reikia bendro sutarimo <...>, kad, žiūrėk, mes deginam čia pinigus.“ (E6); „Būna, kad klientai nežino atsakymų arba neapsisprendžia labai ilgai. <...> Didelis iššūkis – suvaldyti <...> juos kažkaip gražiai, apriboti laike <...> ir iškaulyti atsakymą, kuris reikalingas, kad galėtum judėti toliau <...>.“ (E6); „<...> Jie negali duoti aiškaus atsakymo, kuris <...> reikalingas, kad galėtum suprogramuoti dalykus.“ (E6).
Mažas kliento įsitraukimas	„<...> Vizionieriai vadovai <...> transliuoja labai gerai tą žinutę, bet dingsta, nepasiekiami patampa. <...> Reikia turėti pastoviai pasiekiamus <i>stakeholderius</i> <...>.“ (E1); „Jeigu <i>mindset'as</i> vadovų <...> pasidarom greičiau, čia nėra laiko susitikti, <...> tai <...> visi jaus tą patį, <...> trūkumą, kad <i>engagement'as</i> <...> iki galo neužkurtas <...>.“ (E1); „Reikalavimų surinkimas gali strigti, nes klientas tiesiog neatsakinėja į klausimus, laiškus <...>. Gali būti žmogiškas faktorius, techninis faktorius ar žinių trūkumas.“ (E2); „ <i>Response rate, response time</i> <...> reikia juos sudominti, kad rastų tam laiko.“ (E2); „Įsitraukimas mažas <...>. Tik tada, kai suranda laiko, pasirodo, kad oi, mes tikėjomės ne to. <...> Reikia juos dažnai <i>būzinti</i> <...>, kad kuo greičiau patikrintų ir pasakytų, kas negerai.“ (E2); „Kliento pusėje gali būti žmonių, kurie nelinkę bendrauti <...>. Užsivertę darbais ir čia dar kažkas lenda iš šono <...>.“ (E2); „Kartais gali būti <...>, kad tau nėra norima suteikti konteksto <...>. Tada <...> nesuprantama, kas iš tikrųjų norima pasiekti <...>, atsiranda daug komunikacinių problemų <...>.“ (E3); „ <i>Deploy'inant</i> dažniau <...> klientas galėtų <...> nuolatos stebėti, kas vyksta. <...> Bet dažnai problema yra, kad neįsitraukia.“ (E3); „Jeigu negauni pastabų <...>, tai gali būti ženklas, kad niekas nevyksta.“ (E3); „Negu po penkių mėnesių sužinoti, kad važiuom ne į tą pusę. <...> Reikia rast būdų tai išaiškinti, kol dar nėra blogų pasekmių.“ (E3); „Didelėse korporacijose neprileidžia tiesiogiai prie programuotojų <...>. Galiausiai, kai pavyko susisiekti, per vieną dieną viską išsprendė <...>.“ (E4); „Jeigu nesulauki atsakymo, eskaluoji <...>. Jei sakys, kad neturi laiko, <...> tai paveiks projekto laikotarpį. <...> Čia vėl rizikos apibrėžimas.“ (E4).
Klientui trūksta techninių žinių	„<...> gali būti žinių trūkumas pas klientą pusėje, kur jis net nesugeba įvardinti, su kuo susiduria <...>. Tada turi pats <i>brainstorminti</i> , siūlyti variantus ir žiūrėti, ar pataikai.“ (E2); „Kiek jie techniškai išprusę ir gali įvardinti <...>. Gal nebūtinai techniškai, bet kiek patys gali įvardinti problemą, kurią reikia išspręsti.“ (E2).
Didelis apimties pasikeitimas	„<...> žiūri į sprinto <i>burnout'ą</i> ir matai, kad 30 <i>story point'ų</i> atsirado. Nu jau čia yra <i>scope creep'as</i> <...>. Jie dažniausiai įtakoja visą planą ir resursus <...>. Pokytis iš tikrųjų kenkia – žmonės patiria stresą, perdegimą, nepasitenkinimą.“ (E1); „Labai blogai <i>scope creep</i> , jį būtina valdyti <...>. Jeigu neturim stipresnio PM, tada nugula ant žmonių <...>. Nenormali situacija, kurią tikrai galima sutvarkyti <...>.“ (E1); „Viena dažniausių problemų projektuose – <i>scope creep</i> .“ (E3); „<...> siekiam suprasti ir atliepti <...>. Ar tas <i>scope creep</i> nesukompromituoja mūsų ilgalaikių tikslų? Jei ne, gal galim tą padaryti.“ (E3); „Mačiau, kaip metų projektas pavirsta į dviejų ar dviejų su puse metų projektą, o pinigai tai juk fiksuoti.“ (E6).
Analizės trūkumas	„Dirbo, kad greičiau padaryti ir pritraukti klientus, o dabar jie tapo brandūs <...>. Vadovų <i>mindset'as</i> pasikeitė <...>. Dabar galvoja apie ilgalaikį produkto palaikymą <...>, todėl kyla poreikis labiau planuoti, labiau analizuoti.“ (E1); „<...> Jei analizė nebuvo padaryta, negali pradėti <i>sprint'o</i> <...>. Jei <i>sprint'o</i> viduryje <i>story point'ų</i> padvigubėja, vadinasi, analizė nebaigta <...>. Greičiausiai dirba, bet nespėja į visus reikalavimus atsižvelgti.“ (E1); „Būtų geriau, jei kiekvienoje

	komandoje būtų BA <...>, nes visa analizė nugula ant programuotojų ir trukdo jų darbui <...>. O mes jau vėluojam.“ (E4); „Daug klausimų kyla per vėlai, negu turėtų.“ (E4); „Jei reikalavimų kokybė prasta, tai labai skaudės <i>development</i> 'ui <...>. Jei analizė nepadaryta iš anksto, <i>developer</i> 'iai turės patys daryti analizę, o tai stabdys visą tempą.“ (E5); „Blogi reikalavimai reiškia neaiškius <i>timeline</i> 'us, tada visi pyksta, daugiau laiko skiriama aiškinimuisi, kodėl vėluojama, o ne tikram darbui.“ (E5); „Jei analitiko klaidos prasilenkia su realybe, jos <i>waterfall</i> 'e nuvilnija per visą procesą <...>.“ (E6); „<...> esu mačiusi reikalavimų, kur pusė sakinio – ir še tau suprogramuok <...>. Net jei <i>acceptance criteria</i> detalūs, vis tiek yra tikimybė prasileisti ir padaryti kitaip. O jei reikalavimai beveik neaprašyti – tragedija.“ (E6).
Nepilni reikalavimai	„Jie turi reikalavimų rinkimo procesą, bet jis per mažas – neužtenka atsakyti visiems klausimams, kylantiems <i>development</i> 'o metu. Klausimai dažnai yra iš verslo pusės: kaip jie daro? Koks jų procesas? <...> <i>development</i> 'o metu vėl reikia grįžti pas klientus, vėl tartis.“ (E1); „Jei bendrauji tik su vienu žmogumi kliento pusėje, dažnai gauni tik ledkalnio viršūnę. Reikia atsikasti iki pačios problemos esmės, o tai gali būti sudėtinga <...>. Reikia rasti reikiamus žmones, kurie turi tikrą informaciją.“ (E2); „IT dalyje <i>Product Owner</i> dažnai nebūna techniškai žmonės <...>. Būtina aiškiai nurodyti, kaip aprašyti ir pateikti reikalavimus <...>. Kartais reikia būti atkakliu ir užduoti net elementarius klausimus – nes durnų klausimų nėra.“ (E4).
Neišsakyti reikalavimai	„Klientai dažnai mano, kad jų procesai yra savaime suprantami, nes su jais dirba metų metus. Pirmiausia reikia jiems išaiškinti, kad mes nežinome, kaip viskas vyksta, ir mums reikia aiškios dokumentacijos ar medžiagos analizei ir <i>discovery</i> procesui.“ (E2); „Kai kurie klientai netraktuoja neaiškumų kaip pakeitimų – sako, kad reikėjo skaityti tarp eilučių. Yra atvejų, kai net teko rinkti įrodymus, nes reikalavimų aprašymai buvo pakeisti jau po to, kai <i>development</i> 'as prasidėjo.“ (E4); „Interviu metu gali kalbėti su žmogumi, kuris yra labai idėjiniame lygyje ir skraido padebesiais. Didelis iššūkis perversi jį į konkretų, žemišką lygį, kad išgautum realią, naudingą informaciją.“ (E5).
Skirtingi suinteresuotieji kliento pusėje	„Yra nemažai didelių ligoninių, kurios dirba su dabartine sistema. Kiekviena jų turi savo specifiką – vieni orientuoti į vaikus, kiti į kalėjimus <...>. Procesai bei įpročiai labai skirtingi.“ (E1); „Jo, turbūt, aišku, ten, kai nesutampa reikalavimai.“ (E1); „Niekas negirdi viso to – tu atneši tik santrauką į komandą, <...> dažniausiai pasikartojančius reikalavimus. Jei nepakvieti svarbaus žmogaus, gali būti, kad kažkoks reikalavimas liks neprioritetinis.“ (E1); „Jei papuolei į tokią situaciją, turi surasti tinkamus žmones kliento pusėje. Dažniausiai turi tik vieną ar du kontaktus, bet reikia klausti kitų skyrių žmonių. <...> Kadangi negali tiesiog vaikščioti ir klausinėti, svarbu, kad kas nors nukreiptų pas tinkamą žmogų.“ (E2).
Neteisingos prielaidos	„<...> kad pats pasidarei prielaidą, kad mes viską žinom jau dabar, mums tik reikia implementuoti, ir čia labai neteisinga prielaida.“ (E3); „Visuomet reikia prasivalduoti. <...> pasitikėk tuo, ką žinai, bet vis tiek patikrink.“ (E4); „Aš mačiau, šitam darbe buvo paruoštas vienas projektas – 5 lapų <i>Word</i> dokumentas. Buvo mergina, kuri mėgsta tuos klasikinius dokumentus, kur viską sudėlioja, <...> gaunasi, kad jei tau pirmame žingsnyje nepaeis, visas dokumentas gali iškelti.“ (E5); „Nes tikrai neretu atveju yra tekę, žinai, kalbėti, diskutuoti apie kažkokį <i>feature</i> 'ą, ir tu galvoji, kad jis veiks taip, darys tą, na ir... <...> tu supranti, kad čia darai klaidas.“ (E5); „Ir tada gali ten <i>overengineer</i> 'inti ir padaryti kažkokį fancy sprendimą, kuris nebus nei tai, ko galbūt reikėjo, nei tai, ko klientas norėjo. Tai pirmiausia, susirink problematiką, tą problemą, tada eini link vizijos, <...> prasideda nuo problemos identifikavimo, ką mes šiandien bandome išspręsti.“ (E5).
Patvirtintų reikalavimų kaita	„Pastoviai keičiasi <...>“ (E2); „Būdavo, kad reikėjo rinkti įrodymus, kada buvo pakeisti reikalavimai, nes kartais pakeisdavo <i>description</i> 'ą po <i>development</i> 'o pradžios.“ (E4); „Tuo metu, kai pereinama į <i>dev in progress</i> statusą, pakeitimas gali įvykti ir po to, kai užbaigiamas darbas.“ (E4).
Darbas su kitos kompanijos įvertintais reikalavimais	„Būna sąrašas su šimtu reikalavimų, bet nesi visiškai susipažinęs su kliento veikla, todėl sunku nustatyti, kaip jie pasiskirsto <i>backlog</i> 'e. Ne visi reikalavimai yra vienodi – vienas gali būti 3 <i>story point</i> 'ai, o kitas – 5, todėl <i>backlog</i> 'as nėra tolygus.“ (E6); „<...> <i>discovery</i> fazę atliko kita organizacija, turinti savo patirtį, ir tu jau esi įsiūtęs į šios fazės reikalavimus.“ (E6).
Perteklinė informacija	„<...> dažnai gaunama daug perteklinės informacijos – apie 80% yra nereikalingos.<...> Iš 3 valandų pokalbio tik po 2 mėnesių supranti, kad tik 10 minučių buvo naudinga <...>.“ (E1).
Analizės klaidos	„Bet jeigu pagalvoji vėliau, gali užsisukti ciklas: nepagalvojom, reikėjo kažką taisyti, o taisymas

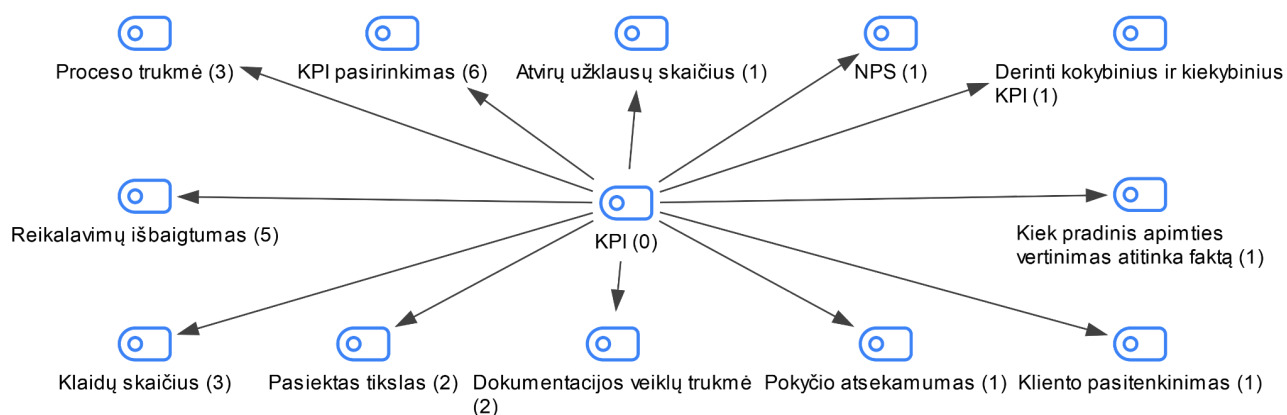
kainuoja brangiai	gali būti brangus – kartais lengvas, kartais reikia perdaryti viską.“ (E3); „Blogo reikalavimų etapo kokybė stabdo visą <i>development</i> ’ą ir neleidžia turėti patikimo darbo tempo <...>.“ (E5).
Ribota prieiga prie įrankių	„Analizės įrankiai <...> dažnai būna riboti dėl finansų arba organizacijos noro investuoti į tinkamą infrastruktūrą. Be tinkamo įrankio, sunku <...>.“ (E1); „Įrankių prieiga yra problema, nes brangūs įrankiai dažnai pasidaro nepasiekiami <...>.“ (E1).
Programinės įrangos kūrimas - tiriamasis darbas	„<...> tai būtų labiau vertės hipotezė, tai ji turi eksperimentą, kaip patikrinsime, ir pamatuojamą vertę, kad prieš įgyvendinant žinotume, kaip matuoti vertę.“ (E3); „ <i>Softo</i> kūrimas yra tiriamasis darbas, kur pradžioje nežinai, kas atsiras vėliau. Estimavimas ir kontraktų rašymas gali būti sudėtingi dėl kompleksiško.“ (E3); „Sunkumai kyla, nes dažnai tikimės, kad viską žinome iš anksto, tačiau realybėje reikalavimai gali nesutapti su tikrais poreikiais.“ (E3); „Verifikavimas ir validavimas yra esminiai procesai, nes reikia pasiekti užtikrintumą ir laipsnišką perėjimą į realizavimą, priešingai nei manyti, kad viską žinome ir padarysime iš karto.“ (E3); „Reikalavimai turėtų būti vertinami kaip poreikiai, kuriuos vartotojas išreiškia per juos, o tikslas – suprasti, kodėl jam to reikia.“ (E5); „Įvertinimas, ar sprendimas geriausiai išsprendžia problemą ir atneša daugiausiai vertės, yra esminis sprendimo priėmimo aspektas.“ (E5); „Sėkmė priklauso nuo greito sprendimų validavimo ir pasirinktos strategijos įgyvendinimo. Taip pereiname nuo projektinių mąstymo prie produkto kūrimo požiūrio, kuriame vertė yra svarbiausia.“ (E5).
Dokumentavimo standartų trūkumas	„Labai daug detalių, nėra vieno šablono pavadinimų, kartais to vieno kaip ir negali būti, bet kažkoks tai atominis dalykas galėtų egzistuoti.“ (E4); „Reikėtų geresnės kokybės aprašų <...>.“ (E4); „Jeigu dokumentacija nepilna, gali atsirasti spragų <...>.“ (E4).
Senos sistemos palaikymas	„<...> incidentas atsiranda, tai reiškia, visi meta darbus ir eina dirbti. <...> Labai sunku palaikyti. Kiekvienas pakeitimas neaišku kuo baigsis <...>“ (E1).
Įsisenėjusios praktikos	„<...> mažai yra lankstumo, žinai, pakeisti net kitą kartą, tu mažiau turi lankstumo įvesti tą gerą praktiką, kur labai daug turi tų savo senų praktikų.“ (E1).
Silpni kliento atstovai	„<...> šiandien dienai pas juos labai silpna yra <i>product owner</i> ’ų komanda, jiems labai sunku patiems <...>.“ (E1); „būna situacija, kada ateina tikrai labai aukštos kvalifikacijos inžinieriai, bet tie vedantys žmonės, kad ir analizė, nu, jie turi labai mažai tų praktikų.“ (E1); „jie dažniausiai nebūna techniškai žmonės. Negali tiesiog pasakyti, kaip reikia aprašyti tuos dalykus, ar kaip tai pateikti BA, jeigu ten BA viską dokumentuoja.“ (E4).
Nėra vizijos, didelio paveiklo	„<...> trūksta to tokio, žinai, kad būtų ir procesai <i>high level</i> , tokia <i>big picture</i> ir daugiau tokių scenarijų aprašytų iš įvairiausių pusių. <...> kiekvieną kartą tas grįžimas labai trukdosi ir tie vėlavimai atsiranda.“ (E1); „<...> bet kaip tie visi reikalavimai susiriša į vieną grandinę, eilutę, to nepamatsi per lentelės prizmę, nes ten jos išsiskirsto į penkis dar kitas.“ (E5).

Tyrimo metu nustatyti šie reikalavimų inžinerijos iššūkiai: aiškumo trūkumas iš kliento pusės, mažas kliento įsitraukimas į programinės įrangos vystymo procesą, kliento techninių žinių trūkumas, didelis apimties pasikeitimas projekto eigoje, analizės trūkumas, nepilni reikalavimai, neišsakyti reikalavimai, skirtingi suinteresuotieji kliento pusėje, neteisingos prielaidos, patvirtintų reikalavimų kaita, darbas su kitos kompanijos įvertintais reikalavimais, perteklinė informacija, brangios analizės klaidos, ribota prieiga prie įrankių, programinės įrangos kūrimas – tiriamasis darbas, dokumentavimo standartų trūkumas, senos sistemos palaikymas, įsisenėjusios praktikos, silpni kliento atstovai, pilno paveiklo nebuvimas. Iššūkiai dažniausiai susiję su neapibrėžtumu, neaiškumu, informacijos trūkumu. Daugelis respondentų minėjo, jog „*stakeholder'iai* <...> nežino, ko nori“, neretai „reikalavimų surinkimas gali strigti, nes klientas tiesiog neatsakinėja į klausimus“ ir dėl to sekančiuose projekto etapuose gali paaškėti, jog „važiavom ne į tą pusę“. Problemos taip pat kyla dėl skubėjimo, stengiantis tilpti į projekto terminus, kai komanda „nespėja į visus reikalavimus atsižvelgti“ ir praleidžiamos esminės detalės. Dėl to nukenčia tiek rezultato kokybė, tiek ir projekto komanda, nes dėl laiko spaudimo „žmonės patiria stresą, perdegimą, nepasitenkinimą“. Keletas tyrimo dalyvių prabrėžė, jog „klientai <...> mano, kad jų procesai yra savaime suprantami“ ir aiškiai neišreiškia lūkesčių, tačiau įgyvendinus

tam tikrą funkcionalumą paaiškėja, jog buvo praleisti tam tikri aspektai ir „reikėjo skaityti tarp eilučių“. Situacija tampa dar sudėtingesnė dėl komunikacijos problemų ir neteisingų prielaidų, kadangi „pats pasidarai prielaidą, kad mes viską žinom jau dabar, mums tik reikia implementuoti“, tačiau tik vėliau paaiškėja, jog „čia labai neteisinga prielaida“. Visi šie paminėti iššūkiai yra itin svarbūs, nes „bloga reikalavimų etapo kokybė stabdo visą *development*’ą ir neleidžia turėti patikimo darbo tempo“, o tai daro tiesioginę įtaką projekto sėkmės rodikliams. Apibendrinant, IT projektų reikalavimų inžinerijos procesuose susiduriama su dideliu neapibrėžtumu, kuriame ypatingai svarbu ne tik formalios procedūros, tačiau ir nuolatinis dialogas su klientu visu projekto įgyvendinimo laikotarpiu.

4.1.2 Reikalavimų inžinerijos procesų vertinimo metrikos IT projektuose

Šiame skyrelyje pateikiami tyrimo metu nustatyti pagrindiniai veiklos rodikliai, kuriuos reikalavimų inžinerijos ekspertai taiko IT projektuose, siekdami įvertinti procesų efektyvumą ir kokybę. 11 paveiksle pateikiamas pagrindinių reikalavimų inžinerijos procesų veiklos rodiklių IT projektuose vizualinis modelis, kuriame vaizduojama pagrindinių veiklos rodiklių kategorija ir 12 subkategorijų, iš kurių 10 yra ekspertų taikomos metrikos. Likusios dvi subkategorijos apima ekspertų įžvalgas apie pagrindinių veiklos rodiklių pasirinkimą ir kokybinių bei kiekybinių rodiklių derinimą.



11 pav. Pagrindinių reikalavimų inžinerijos procesų rodiklių IT projektuose vizualinis modelis

Pagrindinių reikalavimų inžinerijos procesų rodiklių IT projektuose subkategorijos ir jas pagrindžiančios citatos pateikiamos 4 lentelėje.

4 lentelė. Pagrindinių reikalavimų inžinerijos procesų rodiklių IT projektuose subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Kiek pradinis vertinimas atitinka faktą	„<...> seki, kiek tas tavo pradinis vertinimas atitinka vėliau faktą ir tokiu būdu tobulinti savo veiklas <...>.“ (E6).
Kliento pasitenkinimas	„<...> nes siekis būtų vis tik pagrindinis, kad klientas būtų patenkintas rezultatu.“ (E6).
Dokumentacijos veiklų trukmė	„Tai paėmus, tarkim, dokumentacijos dalį tai neturėtų sudaryti daugiau kaip dvidešimt procentų visų veiklų.“ (E6).
Pasiektas tikslas	„<...> vertinsim tuos tikslus, kiek mums pavyko išspręsti ar neišspręsti <...>.“ (E2); „Ar tau pavyko pakeisti tai, ką tu nori pakeisti, ar nepavyko.“ (E5).

Atvirų užklausų skaičius	„<...> turim KPIs, kiek yra atvirų klientų užklausų <...>.“ (E5).
Reikalavimų išbaigtumas	„<...> tas vadinamas <i>completeness</i> , ar visi reikalavimai yra surašyti, ar visos funkcijos padengtos ir yra funkciniai ir nefunkciniai reikalavimai.“ (E4); „<...> tai vėlgi tas aprašas turėtų būti pilnas, išbaigtas ir kiek galima vienodžiau suprantamas.“ (E4); „Kiek tu <i>backlog</i> ’o turi <i>ready</i> pasidaręs, visuomet buvo noras turėti du <i>sprint</i> ’us į priekį.“ (E5); „<...> jeigu žiūrėti dabar reikėtų, tai <i>DEV Ready task</i> ’ų kiekis, kiek kartų <i>task</i> ’as nuėjo į <i>DEV Ready</i> būseną.“ (E5); „<...> tokius suvedinėčiau, kad užtikrinčiau kokybę gerai aprašytų darbų, ko pasekoje reikėtų, kad ir mūsų reikalavimų analizės procesas kažkaip yra stipresnis ir tvirtesnis.“ (E5).
Pokyčio atsekamumas	„<...> ar yra atsekamumas kiekvieno reikalavimo, nėra tokių, kurie tiesiog pasiklydę.“ (E4).
Klientų rekomendavimo indeksas (<i>angl. NPS</i>)	„<...> daryti visokias apklausas, kaip yra <i>NPS</i> ’as. <i>NPS</i> ’ai visokie, tai kiek klientai yra patenkinti tam tikrais produktais ir, vėlgi, tai gali būti irgi vienas iš KPIs, ką naudojome.“ (E2).
Proceso trukmė	„Jeigu problema sako apie tai, kad kažkas negerai su greičiu, kokybe, kažkokiais duomenimis, ir tu tai gali pamatuoti, tai tada tu apsibrėži tai kaip KPIs.“ (E2); „<...> nuo <i>bug escape ratio</i> , iki kažkokių ten <i>cycle times</i> , <i>feedback times</i> , kiek ten <i>review</i> užtrunka procesai.“ (E3); „Atskirų proceso dalių <...> kiek <i>hand’off</i> užtrunka, kiek procesai jų užtrunka ir tada visas <i>lead time</i> iš ko susideda <...> turi pradėti stebėti nuo produkto pusės ir eiti iš produkto per sistemą, per kodą, į procesus.“ (E3).
Klaidų skaičius	„<...> nuo <i>bug escape ratio</i> , iki kažkokių ten <i>cycle times</i> , <i>feedback times</i> , kiek ten <i>review</i> užtrunka procesai.“ (E3); „Kiek <i>bug</i> ’ų išlenda, kiek kartų iš kliento gavai pastabų – tai tau pasakys apie tavo reikalavimų kokybę.“ (E5); „Aš manau, kad galima pamatuoti pagal incidentų kiekį. Kai naują sukurtą sistemą paleidi į gyvenimą, per pirmą ketvirtį reikėtų stebėti, kiek incidentų atsiranda.“ (E6).
Derinti kokybinius ir kiekybinius KPI	„Aš balansuočiau <i>feedback</i> ’ą, vadinamą <i>qualitative</i> , su <i>quantitative</i> . Tai gali būti <i>monitoring</i> as, <i>dashboard</i> ’ai ar metrikos, kurias seki. Svarbu turėti būdus pamatuoti, kaip keičiasi tam tikri dalykai.“ (E3).
KPI pasirinkimas	„KPI gali būti apie gerus dalykus, kuriamą vertę, ir apie blogus dalykus. <...> Svarbu, kad rodiklis kistų norima linkme.“ (E3); „Kartais nežinau, ar blogi dalykai nevyksta, nes sekasi, ar tiesiog jų nematai. Pavyzdžiui, ar sistema saugi, sužinau tik po incidento.“ (E3); „ISO standartai gali padėti rinktis KPI temas.“ (E3); „Eksperimentas turi būti pamatuojamas dar prieš įgyvendinimą.“ (E3); „ <i>SPACE framework</i> ’as padeda matuoti komandos patirtį ir efektyvumą.“ (E3); „KPI neturi tapti siekiamybe, nes tada praranda savo vertę.“ (E3).

Tyrimo metu identifikuota 10 pagrindinių veiklos rodiklių, kuriuos taiko reikalavimų inžinerijos ekspertai IT projektuose: kiek pradinis įvertinimas atitinka faktą, kiek klientas yra patenkintas rezultatu, dokumentavimo veiklų trukmė, pasiektas tikslas, atvirų užklausų skaičius, reikalavimų išbaigtumas, pokyčio atsekamumas, klientų rekomendavimo indeksas (*angl. NPS*), proceso trukmė, klaidų skaičius. Tyrimo dalyviai paminėjo, jog stebint pagrindinius veiklos rodiklius, jie „neturi tapti siekiamybe, nes tada praranda savo vertę“. Pagrindinius veiklos rodiklius išskyrė kaip priemonę pamatuoti „gerus dalykus, kuriamą vertę“ – kliento pasitenkinimas, pasiektas tikslas, klientų rekomendavimo indeksas, „ir <...> blogus dalykus“, tokius kaip klaidų ar incidentų skaičius, kuriuos stengiamasi sumažinti. Ekspertai išskyrė tiek kiekybinius, tiek kokybinius veiklos rodiklius. Kiekybiniai rodikliai nesudėtingai išreiškiami skaitine verte, pavyzdžiui klaidų skaičius, pradinio vertinimo atitikimas faktiniam rezultatui, dokumentacijos veiklų trukmė, atvirų užklausų skaičius, proceso trukmė. Kiekybiniai rodikliai yra sudėtingiau pamatuojami ir tam tikru būdu paverčiami skaitine verte, siekiant stebėti pokytį ir imtis veiksmų proceso tobulinimui. Pavyzdžiui „*SPACE framework*’as padeda matuoti komandos patirtį ir efektyvumą“, o klientų rekomendavimo indeksas (*angl. NPS*) padeda įvertinti, „kiek klientai yra

patenkinti tam tikrais produktais“. Apibendrinant galima teigti, jog siekiant sukurti plačiai apimančią veiklos stebėjimo sistemą svarbiausia atsakingai parinkti ir derinti kokybinius ir kiekybinius pagrindinius veiklos rodiklius.

4.1.3 Reikalavimų inžinerijos procesų tobulinimo metodai IT projektuose

Šiame skyrelyje pateikiami tyrimo metu nustatyti reikalavimų inžinerijos procesų tobulinimo metodai, gerosios praktikos, kurios teigiamai prisideda prie projekto sėkmės.

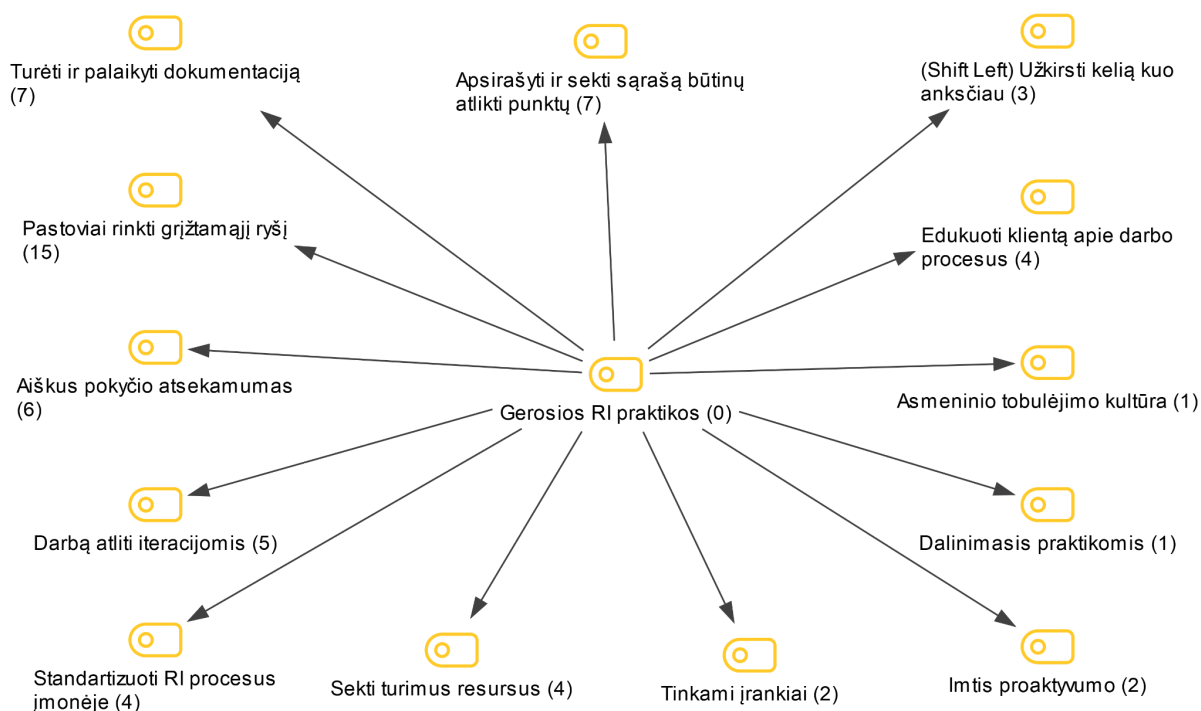
5 lentelėje pateikiama tyrimo metu nustatyta reikalavimų inžinerijos procesų tobulinimo IT projektuose kategorija – šakninės priežasties analizės taikymas – ir ją pagrindžiančios citatos.

5 lentelė. Šakninės priežasties analizės taikymą IT projektų reikalavimų inžinerijoje pagrindžiančios citatos

Kategorija	Citatos
Šakninės priežasties analizės taikymas	„pa'challenge'inti tavo pagrindinį stakeholder'į <...> tai jis šitą brutalų metodą pilnai atlaikys.“ (E1); „Kai žmogus nežino, negadini santykių, bet per projektą bandai sukepti tą <i>project brief</i>'ą, kuris turėjo būti padarytas prieš.“ (E1); „Jei atsiranda nepasitenkinimas, žiūrim, ką planavom, ką darėm, ar supratom teisingai, ar nuomonė tiesiog pasikeitė.“ (E2); „Gali būti ir <i>root cause</i> analizė ar <i>postmortem</i> – atsiverti istorinius reikalavimus ir žiūri, kur įvyko klaida.“ (E3); „Kazkoks incidentas – tada aiškinies, dėl ko kilo, gal reikalavimai nebuvo aiškūs arba apie tai visai nepagalvota.“ (E3); „Visi metodai atsimuša į ciklą – stebi, interpretuoji, kuri hipotezė, eksperimentuoji, tada vėl stebi.“ (E3); „Reikalavimus kartais reikia ištraukti ne iš <i>stakeholder</i> 'ių, o iš programuotojų, kurie turi daugiau patirties.“ (E4); „<...> darėm sesiją išsiaiškinti <i>root causes</i>.“ (E4); „Jei bandai imti interviu, bet nesigauna <...>, ieškai kitų būdų, kaip gauti informaciją.“ (E4); „Pirmas žingsnis – <i>step back to the problem</i>, tik tada galvoti apie galimus sprendimus.“ (E5); „<i>Five Whys</i> galima daryti bet kada – interviu metu ar <i>workshop</i> 'e, žmonės vieni kitus pataiso.“ (E5); „Jei matai, kad kažkas nevažiuoja, bandai suprasti kodėl <...> ir atrasti geriausią būdą, kad tai važiuotų.“ (E5); „Reikalavimus reikia kvestionuoti, nes ne visada tas, kas sugalvojo, gerai įsivertino.“ (E5); „<i>Five Whys</i> metodika: eini gilyn, klausi kodėl <...>, žmonės pradeda nervuotis, bet tik taip iškasi tikrąją problemą.“ (E6);

5 lentelėje pateikti duomenys atskleidžia, kad šakninės priežasties analizė yra svarbus metodas, tobulinant reikalavimų inžinerijos procesus IT projektuose. Tyrimo dalyviai pabrėžia, jog vien tik paviršinis problemų sprendimas dažnai yra neefektyvus, todėl būtina gilintis į esmines priežastis – „pirmas žingsnis – *step back to the problem*, tik tada galvoti apie galimus sprendimus“. Tam taikomos įvairios praktikos, tokios kaip „*Five Whys* metodika: eini gilyn, klausi kodėl <...>, žmonės pradeda nervuotis, bet tik taip iškasi tikrąją problemą“, taip pat incidentų analizė, istorinės informacijos peržiūra. Tokios refleksijos, analizės padeda ne tik spręsti konkrečias problemas, bet ir formuoti gilesnį supratimą apie procesų silpnąsias vietas, o tai leidžia priimti geriau pagrįstus sprendimus ateityje, „atrasti geriausią būdą, kad tai važiuotų“.

Siekiant tobulinti reikalavimų inžinerijos procesus, tyrimo metu buvo surinktos ekspertų taikomos gerosios praktikos IT projektų reikalavimų inžinerijoje. Analizuojant tyrimo duomenis, išryškėjo 13 gerųjų praktikų, kurios, ekspertų teigimu, yra pačios svarbiausios. Vizualinis gerųjų praktikų modelis pateikiamas 12 paveiksle.



12 pav. Gerųjų reikalavimų inžinerijos praktikų IT projektuose vizualinis modelis

6 lentelėje pateikiamos gerųjų reikalavimų inžinerijos praktikų IT projektuose subkategorijos ir jas pagrindžiančios citatos.

6 lentelė. Gerųjų reikalavimų inžinerijos praktikų IT projektuose subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Dalinimasis praktikomis	„Kai vienas <i>stream</i> 'as neapsirašo reikalavimų, sakai – pažiūrėkit, kaip kitas jau pasidarė <...> dalinamės gerosiomis praktikomis.“ (E1).
Asmeninio tobulėjimo kultūra	„Kaip analitikas turi tobulėti, ką tu daugiau nori padaryti, ar tavęs paprašo komanda kažko daugiau. Tu tada tiesiog jau iš savęs mokaisi tą daryti ir darai.“ (E1).
Tinkami įrankiai	„Turim labai didelę problemą su įrankiais.“ (E1); „Įrankis, kuriame gali susivesti reikalavimus kaip <i>backlog</i> 'q, iš esmės išsprendžia daug problemų.“ (E1).
Pastoviai rinkti grįžtamąjį ryšį	„Po <i>sprint</i> 'o gaunam <i>feedback</i> 'q, iš karto keičiam. Tai klientui yra ženklas, kad į jo nuomonę reaguojama.“ (E1); „Grįžti pas tuos pačius žmones su prototipu, tai leidžia patikrinti, ar jie supranta sprendimą – tai naudingas <i>feedback</i> 'o momentas“ (E2); „ <i>Demo</i> metu klientas tikrina, ar viskas veikia taip, kaip įsivaizdavo <...> duoda komentarus.“ (E2); „Idealu gauti <i>feedback</i> 'q dar nelaukiant <i>sprint</i> 'o pabaigos, kad būtų galima greičiau reaguoti.“ (E2); „Bet, iš esmės, jeigu tu negauni pastabų, jeigu tu negauni komentarų, tai galimai yra ženklas, kad tas nevyksta.“ (E3); „<...> Jūs turit dalyvauti tame ir mums duoti nuolatos pastabas, kad mes esam teisingam kelyje ir tik tada mes tą kelią nueisim.“ (E3); „<...> norime, kad tai būtų toks pakeitimas dėl priežasčių A, B, C.“ (E5); „Tikriausiai pati problema labai daug neturėtų keistis, gali keistis nebent kaip jie įsivaizdavo kažką, bet vėl čia jeigu einam per <i>feedback</i> 'o prizmę.“ (E5); „Taip pat procesas buvo, kad visas <i>feedback</i> 'as realiai eina į <i>feedback</i> 'o backlog 'ą, jis buvo atskirtas nuo mūsų <i>development</i> 'o backlog 'o.“ (E5); „Tą <i>feedback</i> 'q reikia kaip galima greičiau procesinti, kad jie galėtų naudoti.“ (E5); „ <i>Fail fast</i> , tai yra <i>feedback</i> 'o loop 'as leidžia tau įsitikinti, kad jie kokybiškai surinkti ir tinkamai dokumentuoti.“ (E5); „Kai tu paleidi tą sprendimą į gyvenimą, tai iš principo tu turėtum būti šalia ir <i>support</i> 'inti klientą ir mokytis iš to, kokie iššūkiai kyla galutiniam sistemos naudotojui.“ (E6).
Apsirašyti ir sekti sąrašą būtinų atlikti	„Greitai pasitikrinam per checklist 'ą – ar backlogas paruoštas, ar reikalavimai aprašyti <...> tai padeda komandai greičiau susivokti“ (E1); „Checklistas leidžia įsivertinti, ar projektas teisingame

punktų	kelyje. Pavyzdžiui, ar atlikti sutarti veiksmai.“ (E1); „ <i>Definition of Done</i> ir <i>Definition of Ready</i> – sąrašas, kokia informacija turi būti paruošta, kad komanda galėtų dirbti“ (E2); „ <i>Checklist'ai</i> padeda nepamiršti <i>edge case'ų</i> .“ (E3); „ <i>Checklist'ą</i> galima naudoti ir <i>refinement'e</i> – jis padeda pastebėti, kurie dalykai dar neapgalvoti.“ (E3); „Reikia turėti aiškius kriterijus, kada <i>user story</i> pakankamai paruoštas sprintui.“ (E6); „Netgi tokius kaip checklist'us pradėjom taikyti ne tik testavime, bet ir reikalavimuose, kad nepamirštume svarbių niuansų.“ (E3).
(<i>Shift Left</i>) Užkirsti kelią kuo anksčiau	„ <i>Shift left</i> principas – geriau numatyti rizikas reikalavimų juodraštyje, nei vėliau jas atrasti testavimo ar validacijos metu.“ (E3); „ <i>Fail fast</i> – tai <i>feedback'o</i> ciklas <...>.“ (E5); „Klasikinis požiūris – laukti 9 mėnesius, kol paaiškės, kad sprendimas neveikia, <...> reikia lankstumo reaguoti anksčiau.“ (E5).
Darbą atlikti iteracijomis	„Jei komanda užduoda daug klausimų – reiškia, reikalavimų surinkimui reikės dar kelių iteracijų.“ (E2); „Pakeitimo toks yra iš esmės iteracinis modelis, kai tu padarei, parodei, susirinkai <i>feedback</i> .“ (E2); „Jei yra rizikų, saugiau eiti palaipsniui – atlikti vidinę validaciją prieš paleidžiant į produkciją.“ (E3); „Tai įgyvendini juos tam tikromis iteracijomis ir klientui darai pristatymus <...>.“ (E6); „Klientas mato, kad vyksta, jis mato rezultatą ir tai juos džiugina ir įkvepia.“ (E6).
Turėti ir palaikyti dokumentaciją	„Įrankis, kuriame reikalavimai suvedami kaip <i>backlog'as</i> , iš esmės sprendžia daug problemų.“ (E1); „Strateginė dokumentacija turi atsakyti ne tik kas padaryta, bet kodėl – tam svarbūs ir <i>decision logs</i> .“ (E3); „Svarbiausia – dokumentaciją palaikyti, nes žmonės keičiasi, o pasenusios žinios sukelia klaidas.“ (E4); „Visa dokumentacija kruopščiai aprašoma, suderinama su klientu, o iš jos gimsta <i>backlog'as</i> .“ (E6); „Specifikacija turi būti tikslinama, kad atitiktų esamą sistemos būklę. Palikti „gyventi sau“ nėra gera praktika.“ (E6); „Kai sistema sudėtinga, dokumentacija būtina – be jos gali būti labai liūdna.“ (E6); „Tikroji vertė yra dokumentacijoje, bet jei užsiimi tik jos rašymu – kažką darai ne taip.“ (E6).
Standartizuoti RI procesus įmonėje	„Kai dirba keli BA, svarbu susitarti dėl vienodos struktūros ir priemonių.“ (E4); „Būdavo tam tikri <i>template'ai</i> , bet jie būdavo tiesiog kaip šablonas pradžiai.“ (E4); „Kai visa įmonė laikosi tų pačių taisyklių, tampa lengviau perimti projektus, pavaduoti kolegas ir mažėja neaiškumų.“ (E6); „Šablonai padeda susieti <i>backlog'o item'us</i> su dokumentacijos skyriais <...> įneša aiškumo ir sutaupo daug laiko.“ (E6).
Imtis proaktyvumo	„Norisi pereiti iš reaktyvaus užsakymų gavimo į proaktyvų modelį, galvoti, kur sistema eis ir ko reikės.“ (E5); „Noras yra susidėlioti struktūrą, kad galėtume proaktyviai spręsti, ką daryti, ko nedaryti ir kokiomis gairėmis vadovautis.“ (E5).
Sekti turimus resursus	„Jeigu <i>sprint'o</i> viduryje smarkiai keičiasi <i>story'ai</i> , reiškia analizė nebuvo padaryta laiku. Tai rodo ir komandų apkrovimą.“ (E1); „Norime įvesti įrankį, kad matytųsi, kurios komandos kiek apkrautos ir ar galime imtis naujų projektų.“ (E5); „Turint geresnį matomumą, galima realistiškiau planuoti <i>roadmap'ą</i> , išvengti pažadų, kurių negalima įgyvendinti.“ (E5); „Geriau planuojant atsiranda didesnis <i>predictability</i> ir lengviau valdyti lūkesčius.“ (E5).
Aiškus pokyčio atsekamumas	„Reikalavimai turi aiškų šaltinį, kad kilus klausimams galėtum greitai patikrinti.“ (E1); „Kiekvienas reikalavimas siejamas su konkrečiu žmogumi ir jo poreikiu – tai padeda jį detalizuoti.“ (E1); „Pokyčiai strategijoje turi būti dokumentuoti. Tai svarbu norint suprasti, kodėl keičiasi reikalavimai.“ (E3); „Svarbu užtikrinti, kad nė vienas reikalavimas nepasiklystų, reikia aiškaus atsekamumo.“ (E4); „Reikalavimų palaikymas tiesiogiai susijęs su pokyčių valdymu, jei jis neveikia, atsiranda daug spragų.“ (E4); „Atsekamumo matrica leidžia matyti, kaip vieno reikalavimo pokytis paveiks kitus.“ (E6).
Edukuoti klientą apie darbo procesus	„Jeigu klientas supranta, kaip veikia <i>Agile</i> <...>, jisai supranta, kad kiekvienas jo pakeitimas reiškia tam tikrą pinigų išmetimą į balą.“ (E2); „Vienas iš pagrindinių dalykų – apšviesti klientą apie darbo procesus <...>, ką tai kainuoja ir ką reiškia įsitraukti į dabartinį planą.“ (E2); „Reikia klientui duoti suprasti, kokį impulsą <...> tai turi bendrai projekto eigai.“ (E2); „Tu turi edukuoti klientą, kad suprastų to vertę <...>, kad mes esam teisingam kelyje.“ (E3).

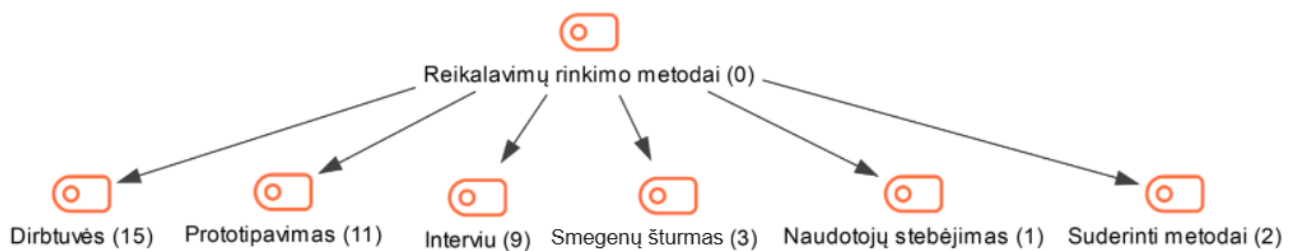
6 lentelėje pateikti duomenys atskleidžia gerąsias reikalavimų inžinerijos ekspertų praktikas, kurios leidžia užtikrinti sklandžius procesus ir gerus projekto sėkmės rodiklius. Pastebima, jog išskiriami tiek techniniai aspektai, tokie kaip kokybiškų ir tinkamų įrankių turėjimas „iš esmės išsprendžia daug problemų“, tiek procesiniai, pavyzdžiui nuolatinis grįžtamojo ryšio rinkimas, skatinantis klientus

„dalyvauti tame ir mums duoti nuolat pastabas, kad mes esam teisingam kelyje“ ar aiškus pokyčio atsekamumas, kadangi „jei jis neveikia, atsiranda daug spragų“, kurios vėliau atveria galimybes klaidoms ir netinkamoms interpretacijoms. Grįžtamojo ryšio rinkimas ypatingai akcentuojamas ekspertų, kadangi „*feedback'o loop'as* leidžia tau įsitikinti, kad jie kokybiškai surinkti ir tinkamai dokumentuoti“ – tik tokiu būdu galima įsitikinti, jog einama teisingu keliu, kliento ir komandos interpretacijos sutampa. Su grįžtamojo ryšiu siejamas ir „*Shift Left*“ principas, sakantis, jog „geriau numatyti rizikas reikalavimų juodraštyje, nei vėliau jas atrasti testavimo ar validacijos metu“, kadangi „klasikinis požiūris – laukti 9 mėnesius, kol paaiškės, kad sprendimas neveikia“ sukelia problemų ateityje. Siekiant užkirsti kelią klaidoms kuo anksčiau, taikomas iteracinis darbo modelis, todėl „klientas mato, kad vyksta, jis mato rezultatą ir tai juos džiugina ir įkvepia.“ Neretai ekspertai pabrėžė, jog „vienas iš pagrindinių dalykų – apšviesti klientą apie darbo procesus“, komunikacijos svarbą, „kad suprastų to vertę <...>, kad mes esam teisingam kelyje“ dėl to, jog neretai klientai būna įpratę dirbti kitokiais principais ir atsiranda tarpusavio nesupratimas.

4.1.4 Reikalavimų surinkimo metodai IT projektuose

Šiame skyrelyje pateikiami tyrimo metu nustatyti ekspertų dažniausiai taikomi reikalavimų surinkimo metodai, identifikuoti jų privalumai ir trūkumai bei tinkamo parinkimo aspektai.

Tyrimo metu identifikuoti reikalavimų surinkimo metodai skirstomi į 6 subkategorijas: dirbtuvės, prototipavimas, interviu, smegenų šturmas (angl. *brainstorming*), naudotojų stebėjimas bei skirtingų metodų derinimas.



13 pav. Ekspertų taikomų reikalavimų rinkimo metodų IT projektuose vizualinis modelis

7 lentelėje pateikiamos reikalavimų surinkimo metodų subkategorijos ir jas pagrindžiančios citatos.

7 lentelė. Reikalavimų surinkimo metodų IT projektuose subkategorijos ir pagrindžiančios citatos

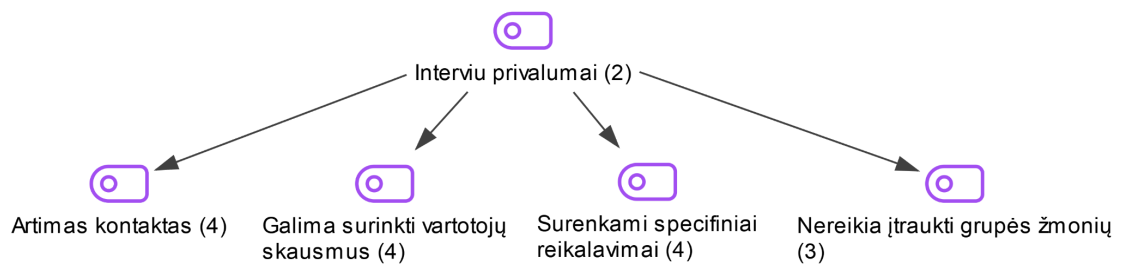
Subkategorija	Citatos
Interviu	„Kartais užtenka vieną vartotoją pasisodinti, gerai iškratyti <...>, bet visumoj jau tie kiti vartotojai tik užtvirtina.“ (E1); „Tu darai daug interviu <...>, matai, kad iš dešimt aštuoni turi tą patį <i>pain point'ą</i> .“ (E1); „Kai pradedi tą interviu daryti, 3 valandas kalbi, o paskui paaiškėja, kad tik 10 minučių buvo naudinga.“ (E1); „Turi klausimus, siunti iš anksto <...>, pagal tų klausimų turinį parenka tinkamus žmones.“ (E1); „Interviu tikslas mūsų atveju – surinkti skausmus, ir jis taip gerai veikia, kad jų nereikia net <i>brainstorm'uose</i> ieškoti.“ (E1); „Nuvažiuojame pas juos <...>, jie papasakoja, kokie tai yra skausmingi taškai.“ (E2); „Po susitikimo nusiunčiam <i>summary</i> <...> tai pirmas momentas, kai klientas gali pasakyti, kad neteisingai supratom.“ (E2); „Interviu galima daryti tiek didesniems dalykams, tiek mažesniems – pasiskambinti keliolikai minučių dėl konkrečios problemos.“ (E2); „Realiai paprasčiausias yra interviu – gyvas kontaktas su klientu <...>, dažnai naudojamas ir 5 <i>whys</i> įrankis.“ (E5); „Interviu priklauso nuo konteksto ir tikslo <...>, mažiau žmonių – paprastesnis darbų <i>scope</i> .“ (E5);
Dirbtuvės	„Jeigu <i>stakeholder'ų</i> nedaug, užtenka valandos ar dviejų <i>workshop'o</i> – vienas pasisako, kiti užtvirtina.“ (E1); „Mintis yra suvesti visus į vieną vietą ir pasitvirtinti reikalavimus – ar tai LEAN <i>workshop'as</i> , ar <i>story mapping'as</i> , ar <i>event storming'as</i> .“ (E1); „ <i>Workshop'as</i> keliom dienom,

	skirtingom temom, veikia geriausiai – ko neatsako pirmą dieną, atsako antrą.“ (E1); „Kai jau žinai pirminius reikalavimus, galima su <i>end-useriais</i> per workshopą pažaisti – pabraižyti <i>view'ą</i> ir pasižiūrėti, kaip supranta.“ (E1); „<i>Workshop'as</i> su kliento ir mūsų atstovais padeda išsikalbėti apie situaciją, problemas, kaip dirba, ko norėtų.“ (E2); „Projekto pradžioje darom kelių dienų <i>workshop'us</i>, kad suprastume kliento verslo procesus ir <i>know-how</i>.“ (E2); „<i>Workshop'as</i> padeda surinkti informaciją apie verslo modelį, bet tai ilgas ir brangus procesas, daromas prieš didelį projektą.“ (E2); „Per <i>workshop'us</i> ir <i>brainstorm'us</i> dažnai pagaunam tas vietas, kurių klientas šiaip nepaminėtų.“ (E2); „Modernesni metodai, kaip <i>story mapping</i>, <i>event storming</i> ar <i>context mapping</i>, padeda struktūruoti reikalavimus.“ (E3); „Naudoju <i>event storming'ą</i>, kad vizualizuočiau procesą ir parodyčiau klientui, kaip supratau, ką jis pasakoja.“ (E5); „Kai dalyvauja daugiau žmonių, gali daryti mini <i>workshop'us</i> su klausimais, idėjomis, <i>story mapping'u</i>.“ (E5); „<i>Workshop'as</i> padeda greičiau surinkti reikalavimus ir suvienodinti visų supratimą apie projekto <i>scope</i>.“ (E5); „Kai reikia bendro matymo tarp skirtingų skyrių, natūraliai darom <i>workshop'ą</i>.“ (E5); „<i>Event storming'as</i> suteikia <i>alignment'ą</i> tarp skirtingų skyrių – visi pamato, kaip procesas atrodo visumoje.“ (E5); „Jeigu gerai fasilituoji <i>workshop'ą</i>, visi įsitraukia. Net jei neįpratę kalbėti, nuteikus tinkamai, veikia.“ (E5);
Prototipavimas	„Duodi kelis <i>screen'us</i> ir prašai atlikti užduotį. Tada žiūri, kuris variantas suprantamiausias <...>, ar be mokymų galėtų naudotis.“ (E1); „Po interviu grįžti su sprendimu, duodi pakliškinti prototipą ir stebi, ar supranta funkcionalumą.“ (E2); „Kartais duodam aprašytą <i>story</i> paskaityti, ar teisingai supranta, kol nepradėjom daryti.“ (E2); „Paruošiam dizainą ar prototipą, grįžtam pas klientą – jei jis nesupranta, galbūt mes nesupratom problemos.“ (E2); „Prototipas paduodamas vartotojams išsibandyti, o po testavimo grįžtama prie dizaino ar interviu.“ (E4); „Viena iteracija pakeitė sprendimą 180 laipsnių, kai jau buvo pradėtas <i>development'as</i>.“ (E4); „<i>Mock-up'ą</i> gali padaryt per porą valandų – <...> pigiau ir greičiau nei taisyti po <i>development'o</i> ir <i>QA</i>.“ (E5); „Prototipai padeda pašalinti dviprasmybes – viską daryti per pavyzdžius. Jei taip – tada taip; ar tikrai tokio rezultato tikėtės?“ (E5); „Viskas prasideda nuo interviu, <i>five whys</i>, tada – <i>mock-up'ai</i> ir pavyzdžiai.“ (E5); „Prototipas gali būti viskas – nuo piešinuko iki <i>click'able</i> versijos ar <i>proof of concept</i> <...>, kad patikrintum prielaidą.“ (E5); „Galima prototipuoti sistemos langus, procesus – tai padeda išsigryninti, ar einam teisingu keliu.“ (E6);
Smegenų šturmas (angl. <i>brainstorming</i>)	Kai paruošiam uždavinį, pasidarom su inžinieriais <i>brainstorm'o</i> sesiją <...>, kad visus atlaisvinti, panaudoti idėją.“ (E1); „Dirbant su inžinieriais, tas paskatina sprendimus greičiau priimti iš techninės pusės.“ (E1); „Kai klientas nesugeba įvardinti problemos, turi pats <i>brainstorm'inti</i> savo pusėje ir siūlyti variantus.“ (E2);
Naudotojų stebėjimas	„<i>Usability</i> testavimo sesijoje duodi žmogui užduotį ir stebi, kaip jis naudoja sistemą <...>, kurios vietos patogios, kurios ne.“ (E3);

7 lentelėje pateikiami tyrimo metu identifikuoti penki ekspertų naudojami reikalavimų surinkimo metodai, leidžiantys geriau suprasti vartotojų poreikius, surinkti aiškesnius ir labiau pagrįstus reikalavimus. Interviu pažymimas kaip vienas iš pagrindinių reikalavimų surinkimo metodų, kuris leidžia tiesiogiai iš klientų „surinkti skausmus, ir jis taip gerai veikia, kad jų nereikia net *brainstorm'uose* ieškoti.“ Interviu metodas taip pat išskiriamas kaip plačiai naudojamas dėl paprastumo ir efektyvumo. Antras išskirtas metodas – dirbtuvės (angl. *workshop*), kurios padeda ne tik greičiau surinkti reikalavimus, bet ir pasiekti bendrą supratimą tarp skirtingų suinteresuotųjų šalių. Dirbant grupėje išryškėja niuansai, kurių individualiuose pokalbiuose nepavyktų užfiksuoti: „per *workshop'us* ir *brainstorm'us* dažnai pagaunam tas vietas, kurių klientas šiaip nepaminėtų“. Naudojami tokie šiuolaikiniai metodai kaip įvykių modeliavimas (angl. *Event Storming*), naudotojų patirties žemėlapis (angl. *User Story Mapping*), konteksto žemėlapis (angl. *Context Mapping*), kurie padeda vizualizuoti procesus ir supaprastina komunikaciją – „*event storming'as* suteikia *alignment'ą* tarp skirtingų skyrių – visi pamato, kaip procesas atrodo visumoje.“ Dar vienas labai svarbus reikalavimų surinkimo ir išgryninimo metodas – prototipavimas. Šis metodas leidžia pastebėti „ar supranta funkcionalumą“, tokiu būdu renkant įžvalgas ir tobulinant, kol sistema tampa aiški ir intuityvi. Taikant prototipavimą galima greitai identifikuoti misinterpretaciją, pastebėti jog jei „klientas nesupranta, galbūt mes nesupratom problemos“. Ekspertai pažymi šio metodo efektyvumą, kaip pavyzdžiui „viena iteracija pakeitė

sprendimą 180 laipsnių“ ir parodė, jog sprendimą reikia iš esmės keisti. Tai leidžia užkirsti kelią nesusipratimams tolimesniuose projekto etapuose, kadangi „*mock-up*‘ą gali padaryti per porą valandų – <...> pigiau ir greičiau nei taisyti po *development*’o ir *QA*.“ Esant poreikiui išgryninti ar atrasti tam tikrus sprendimus taikomas smegenų šturmo metodas, kuris „paskatina sprendimus greičiau priimti iš techninės pusės.“ ir padeda atrasti sprendimus, kai „klientas nesugeba įvardinti problemos“. Naudotojų stebėjimas naudojamas siekiant patikrinti, kaip realūs vartotojai naudoja sistemą: „duodi žmogui užduotį ir stebi, kaip jis naudoja sistemą <...>, kurios vietos patogios, kurios ne“. Šis metodas leidžia identifikuoti vartotojo patirties spragas, kurių nebūtų galima nustatyti vien kalbantis ar remiantis prielaidomis. Apibendrinant, siekiant surinkti išsamius, kokybiškus ir realius klientų poreikius atspindinčius reikalavimus sistemai, turi būti taikomi ir derinami įvairūs metodai – nuo individualių pokalbių iki grupinių sesijų, prototipavimo. Kiekvienas metodas padeda skirtingais aspektais išgryninti reikalavimus ir sumažinti klaidų tikimybę.

Interviu ir dirbtuvės – ekspertų dažniausiai minimi reikalavimų surinkimo metodai, kurie derinami tarpusavyje duoda labai gerus rezultatus. Dėl šios priežasties interviu ir dirbtuvių privalumai bei trūkumai išskiriami atskiromis kategorijomis ir subkategorijomis. Interviu privalumų vizualinis modelis pateikiamas 14 paveiksle.



14 pav. Interviu privalumų vizualinis modelis

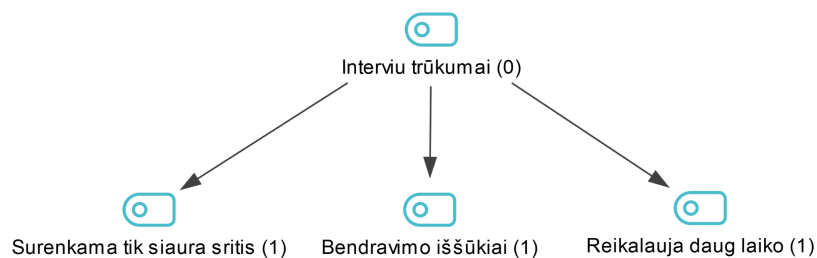
8 lentelėje pateikiamos interviu privalumų subkategorijos ir jas pagrindžiančios citatos.

8 lentelė. Interviu privalumų subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Artimas kontaktas	„Po susitikimo nusiunčiam <i>summary</i> <...>, tai pirmas momentas, kai klientas gali pasakyti, kad neteisingai supratom.“ (E2); „Pasikalbini žmones, kurie naudoja produktą, ir susirenki focus grupę, kuri duoda <i>feedback</i> .“ (E2); „Rašytiniai atsakymai sukelia papildomų klausimų, kuriuos galėtum iš karto paklausti per interviu.“ (E4); „<...> tai gyvas kontaktas su klientu.“ (E5);
Galima surinkti vartotojų skausmus	„Darai daug interviu <...> ir matai, kad iš dešimt aštuoni turi tą patį <i>pain point</i> ‘ą.“ (E1); „Jeigu interviu tikslas yra surinkti skausmus, tai jis labai gerai veikia.“ (E1); „Nuvažiuojame pas juos <...>, jie papasakoja, kokie tai yra skausmingi taškai, kodėl tai skauda.“ (E2); „<...> kai pasiimi vieną vartotoją ir nori jo <i>experience</i> ‘ą susirinkti.“ (E5);
Nereikia įtraukti žmonių grupės	„Matai, kad iš dešimt aštuoni turi tą patį <i>pain point</i> ‘ą <...>, tada neįtrauki grupės, tiesiog daugiau apklausi.“ (E1); „Pliusas tas, kad nereikia visų įtraukti – po kelias valandas su skirtingom grupėm pasiklausai, kaip jie dirba.“ (E1); „<...> daug tų vartotojų, su skirtingais profiliais, tai vat tada darai tuos interviu.“ (E1);
Surenkami specifiniai reikalavimai	„Kai labai specifinė vartotojų grupė, užtenka vieną pasisodinti, gerai iškratyti <...>, kiti tik užtvirtina.“ (E1); „Galima <i>remote</i> ‘u pasiskambinti keliolikai minučių ir pasišnekėti apie konkrečią problemą.“ (E2); „Jeigu mažas <i>feature</i> ‘as, gali kalbėt su vienu <i>power user</i> ‘iu, daug žmonių nereikės.“ (E5); „Intervas mažesnis iš savęs – mažiau žmonių, paprastesnis darbų <i>scope</i> .“ (E5);

8 lentelėje pateiktos pagrindinės interviu metodo stiprybės apima: artimą kontaktą, skausmų surinkimą, kliento laiko taupymą ir galimybę surinkti specifinius reikalavimus. Artimas kontaktas pažymimas kaip vienas didžiausių interviu metodo privalumų, kadangi „rašytiniai atsakymai sukelia papildomų klausimų, kuriuos galėtum iš karto paklausti per interviu.“ Be to, interviu suteikia galimybę surinkti vertingą grįžtamąjį ryšį ir padeda klientui reflektuoti: „po susitikimo nusiunčiam *summary* <...>, tai pirmas momentas, kai klientas gali pasakyti, kad neteisingai supratom.“ Vartotojų skausmų identifikavimas – tai dar viena interviu stiprioji pusė, leidžianti susidaryti išsamų problemos suvokimą, kai „pasiimi vieną vartotoją ir nori jo *experience’q* susirinkti“, klientai „papasakoja, kokie tai yra skausmingi taškai, kodėl tai skauda.“ Interviu metodo „pius tas, kad nereikia visų įtraukti“, o tai taupo kliento atstovų laiką, ypač kai siekiama surinkti atskirų naudotojų įžvalgas, nesukeliant didelio organizacinio apkrovimo. Galiausiai, interviu metodas leidžia surinkti specifinius reikalavimus, gilintis į konkrečią problemą ar funkcionalumą, kalbant „su vienu *power user’iu*“. Apibendrinant, interviu metodas tinkamas surinkti naudotojų skausmams, gilintis į specifines problemas ar funkcionalumus ir nesukuria didelio organizacinio apkrovimo, nes nereikia įtraukti daug žmonių.

Siekiant nustatyti interviu metodo pritaikymo sritis, interviu trūkumai išskirti kaip atskira kategorija su subkategorijomis. Trūkumų identifikavimas padeda atpažinti situacijas, kuriose šis metodas nebūtų pats tinkamiausias. 15 paveiksle pateikiamas interviu trūkumų vizualinis modelis.



15 pav. Interviu trūkumų vizualinis modelis

9 lentelėje pateikiamos interviu trūkumų subkategorijos ir jas pagrindžiančios citatos.

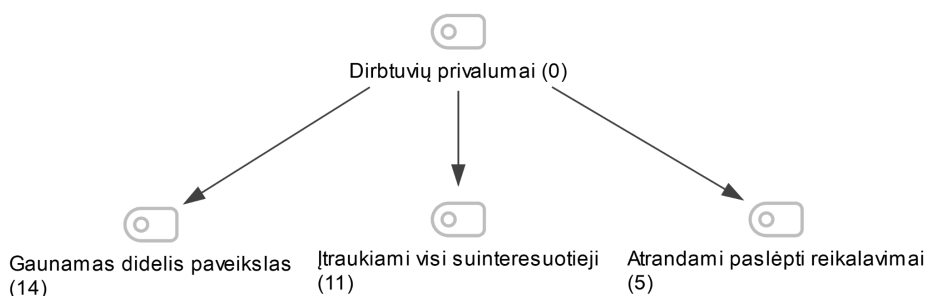
9 lentelė. Interviu trūkumų subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Surenkama tik siaura sritis	„Jei darai <i>one on one</i> , tai gauni tik jų skyriaus požiūrį <...>, bet vėl gali išsiskirti.“ (E5);
Bendravimo iššūkiai	„Jeigu nelabai išeina bendrauti, tai žmogus tik apsidžiaugs pasiūlyti kitą. Sakys, šitas gali ateiti ir atsakyti.“ (E4);
Reikalauja daug laiko	„Kai pradedi daryti interviu, 3 valandas kalbi, o tik po 2 mėnesių supranti, kad naudinga buvo tik 10 minučių.“ (E1);

9 lentelėje išskiriami ekspertų įvardinti pagrindiniai interviu metodo trūkumai. Vienas iš dažniausiai pasitaikančių iššūkių – siaura surenkamos informacijos sritis. Atliekant interviu dažnai gaunamas tik vieno padalinio ar naudotojo perspektyvos fragmentas: „jei darai *one on one*, tai gauni tik jų skyriaus požiūrį“. Tai gali lemti tik dalinį problemos matymą ar net prieštaringus reikalavimus, jei nėra papildomos koordinacijos tarp skirtingų interesų grupių. Taip pat pastebimi bendravimo sunkumai, ypač kai pašnekovas nesijaučia komfortiškai ar neturi pakankamai žinių: „jeigu nelabai išeina bendrauti, tai žmogus tik apsidžiaugs pasiūlyti kitą.“ Tokiais atvejais kyla rizika negauti kokybiškos informacijos arba uždelsti sprendimų priėmimą, jei tenka keisti kliento atstovus. Trečias aspektas – laiko sąnaudos. Nors

interview dažnai išskiriamas kaip vertingas įrankis, jis reikalauja nemažai laiko ir sukuria daug perteklinės informacijos: „kai pradedi daryti interview, 3 valandas kalbi, o tik po 2 mėnesių supranti, kad naudinga buvo tik 10 minučių“. Tai rodo, kad norint pasiekti maksimalią naudą, būtina taikyti tinkamus informacijos analizės metodus, taip pat nuolat vertinti interview efektyvumą projekto kontekste.

Dirbtuvės – tai reikalavimų surinkimo metodas, įtraukiantis didesnes žmonių grupes, leidžiantis surinkti aukštesnio lygio reikalavimus, apjungiantis visas sistemos dalis ir suteikiantis bendrą projekto vaizdą. 16 paveiksle pateikiamas tyrimo metu nustatytų dirbtuvių metodo privalumų vizualinis modelis.



16 pav. Dirbtuvių privalumų vizualinis modelis

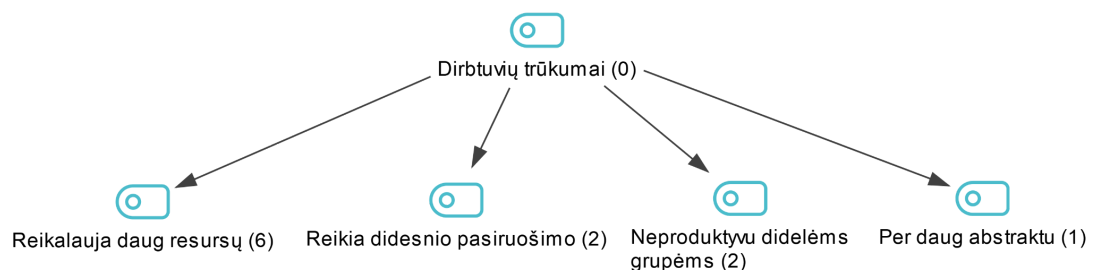
10 lentelėje pateikiamos dirbtuvių privalumų subkategorijos ir jas pagrindžiančios citatos.

10 lentelė. Dirbtuvių privalumų subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Atrandami paslėpti reikalavimai	„Per <i>brainstorm</i> 'o sesijas ar <i>workshop</i> 'us pagaunam tas vietas, kurių klientas šiaip gal ir nepaminėtų.“ (E2); „Per <i>workshop</i> 'q išsidėlioji visą <i>flow</i> 'q ir sugaudai mažas detalėles, už kurių gali slėptis svarbi logika.“ (E2); „ <i>Workshop</i> 'as padeda užkabinti siūlo galą <...> ir išsitraukti nežinomą informaciją.“ (E2); „Kai turi <i>user story map</i> 'q, gali kvestionuoti, ar viską pasiekėm, ir <i>slice</i> 'int iki minimalaus <i>deliverable</i> .“ (E5); „ <i>User story mapping</i> 'as padeda galvoti ne per sistemą, o per tai, ko reikia vartotojui ir kur jo vertė.“ (E5);
Gaunamas didelis paveikslas	„Suvedi visus į vieną vietą <...>, ar tai <i>LEAN workshop</i> 'as, ar <i>event storming</i> as.“ (E1); „ <i>Workshop</i> 'as keliom dienom, skirtingom temom. Pirmą dieną neatsako, antrą atsako, įsijungia visi.“ (E1); „Pabraižai <i>view</i> 'sq ir pažiūri, kaip supranta, ką nori perteikti – tai padeda matyti visumą.“ (E1); „<...> kaip dirba ir kaip norėtų, kad veiktų, susidaro bendras paveikslas.“ (E2); „ <i>Workshop</i> 'uose einam iš <i>high level</i> 'io, kaip jie dirba, ką daro, kaip vysto gamybą ir kas vyksta įmonėje.“ (E2); „Per <i>workshop</i> 'q su klientu dėliojam <i>story map</i> 'q – atsiranda bendras vaizdas, kaip vyksta verslo procesas.“ (E2); „Surenkam informaciją ne tik apie problemą, bet ir apie verslo modelį, kaip klientas dirba.“ (E2); „Pradedam nuo aukštesnio abstrakcijos lygio <...>, sąmoningai paliekam detales vėlesniam etapui.“ (E3); „Naudoju <i>event storming</i> 'q ir <i>story mapping</i> 'q – piešiu procesą, parodau klientui, kaip supratau.“ (E5); „ <i>Workshop</i> 'o metu apsirėži ribas – <i>scope</i> , <i>problem statement</i> , viziją.“ (E5); „ <i>Event storming</i> 'as suteikia <i>alignment</i> 'q tarp skyrių – pamato, kaip procesas atrodo visumoje.“ (E5); „ <i>Story mapping</i> 'as leidžia pamatyti, ar nueisi iki savo norimo rezultato.“ (E5); „Per <i>flow</i> 'us gali vesti dialogą, koks greičiausias kelias iki norimo rezultato.“ (E5); „Pirmiausia važiuoji pas klientą surinkti bendrinių reikalavimų iš aukšto lygio <...>, kad nusistatytum bendrą projekto apimtį.“ (E6);
Įtraukiami visi suinteresuotieji	„Jeigu mažai <i>stakeholder</i> 'ių, užtenka valandos ar dviejų <i>workshop</i> 'o – vienas pasisako, kiti užtvirtina.“ (E1); „Suvedi visus į vieną vietą <...>, pasitvirtini reikalavimus <i>workshop</i> 'o metu.“ (E1); „Pirmą dieną neatsako, antrą atsako, įsijungia visi.“ (E1); „<...> po kavos pertraukos grįžta visi su mintimis – labai veikia.“ (E1); „ <i>Engagement</i> 'as padidėja, kai įtrauki visiškai visus.“ (E1); „Susirenka žmonės iš skirtingų skyrių ir pasakoja, kaip jie dirba.“ (E2); „Gali daryti mini <i>workshop</i> 'us <...>, rinkti mintis, idėjas.“ (E5); „Kai nori bendro suvokimo, turi visus viename kambary, kad vienu metu ateitų tas suvokimas.“ (E5); „Kai didesnė apimtis, natūraliai dalyvauja daugiau žmonių iš skirtingų skyrių, tada darai <i>workshop</i> ą.“ (E5); „ <i>Event storming</i> 'as padeda skyriams matyti visą procesą – kur jungiasi jų dalys, ateina „aha“ momentas.“ (E5); „Gyvos diskusijos veikia geriau nei pristatymai – kai tik pristatinėji, visi tyli, bet nesupranta iki galo.“ (E6);

10 lentelėje išskirti ekspertų įvardinti pagrindiniai dirbtuvių, kaip reikalavimų surinkimo metodo, privalumai. Dirbtuvės išskirtos kaip tinkama priemonė atrasti paslėptus, tiesiogiai neįvardijamus reikalavimus: „sugaudai mažas detalėles, už kurių gali slėptis svarbi logika.“ Ekspertai pabrėžia, jog dirbtuvių metu „pagaunam tas vietas, kurių klientas šiaip gal ir nepaminėtų“ dėl to, jog į dirbtuves susirenka grupė kliento atstovų, kurie turi skirtingą matymą, savas procesų interpretacijas, tad tokiu būdu dirbtuvės padeda „užkabinti siūlo galą <...> ir išsitraukti nežinomą informaciją.“ Kaip dar vienas dirbtuvių privalumas išskiriamas bendro, visą projektą apimančio vaizdo, sudarymas: „susidaro bendras paveikslas.“ Anot ekspertų, toks ir yra dirbtuvių tikslas: „pradedam nuo aukštesnio abstrakcijos lygio <...>, sąmoningai paliekam detales vėlesniam etapui.“ Dirbtuvių metu surinktas bendras paveikslas vėliau yra gryninamas atskiruose interviu su atitinkamos srities kliento atstovais. Didelį paveikslą surinkti padeda pagrindinių suinteresuotųjų šalių įtraukimas, suvedimas, dėl to „*engagement*‘as padidėja, kai įtrauki visiškai visus.“ Suinteresuotųjų šalių įtraukimas vienu metu taip pat suteikia naudą klientui, nes „turi visus viename kambary, kad vienu metu ateitų tas suvokimas“, kuris sukuria bendrą suvokimą tarp skirtingų kliento atstovų. Įvykių modeliavimo (angl. *Event Storming*) metodas pažymimas, kaip priemonė, padedanti „skyriams matyti visą procesą – kur jungiasi jų dalys, ateina „aha“ momentas.“ Dirbtuvių metu vyksta gyvos diskusijos, skatinami įsitraukti visi dalyviai, lyginant su paprastais pristatymais, tai yra geriau, nes „kai tik pristatinėji, visi tyli, bet nesupranta iki galo“, tuo tarpu diskusijų metu galima identifikuoti interpretacijų skirtumus.

Siekiant nustatyti dirbtuvių metodo pritaikymo sritis, dirbtuvių trūkumai išskirti kaip atskira kategorija su subkategorijomis. Trūkumų identifikavimas padeda atpažinti situacijas, kuriose šis metodas nebūtų pats tinkamiausias. 17 paveiksle pateikiamas dirbtuvių trūkumų vizualinis modelis.



17 pav. Dirbtuvių trūkumų vizualinis modelis

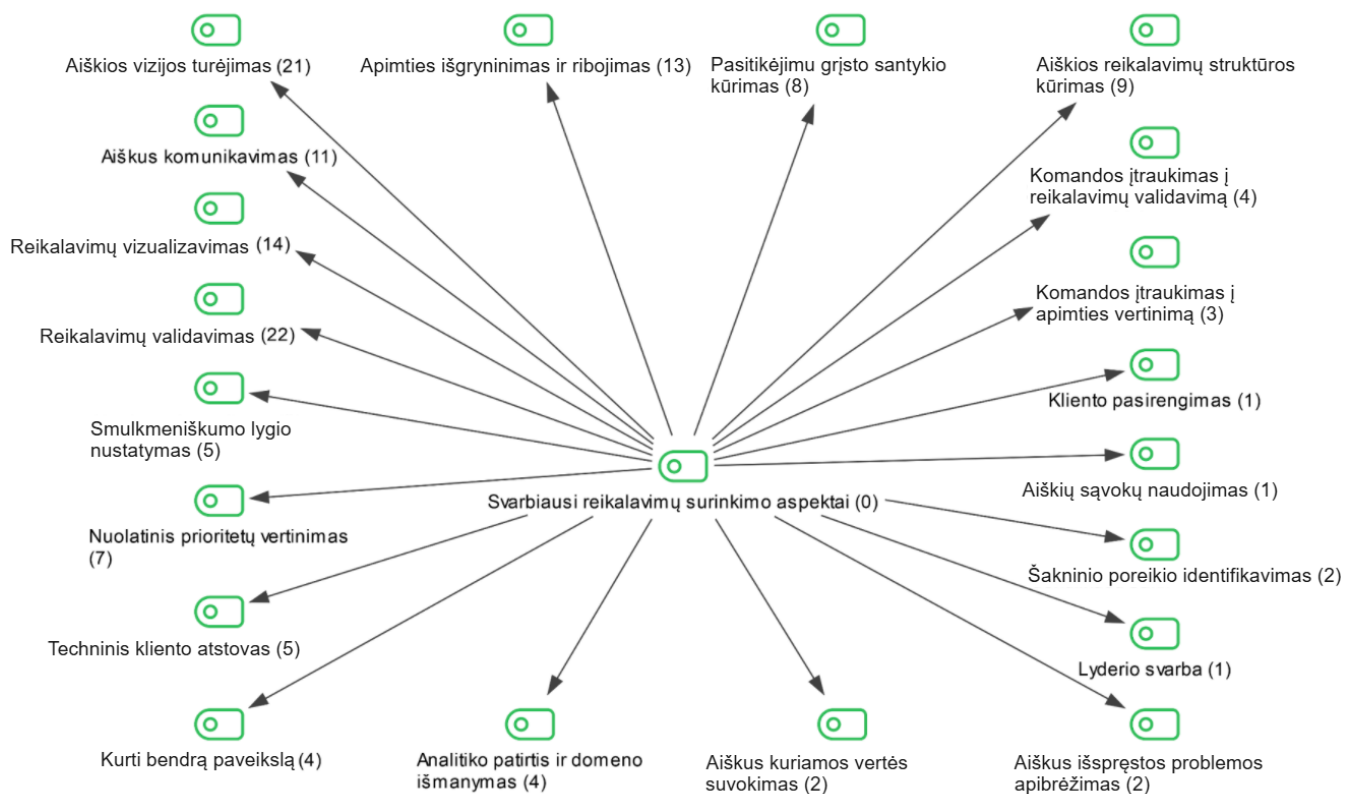
11 lentelėje pateikiamos dirbtuvių trūkumų subkategorijos ir jas pagrindžiančios citatos.

11 lentelė. Dirbtuvių trūkumų subkategorijos ir pagrindžiančios citatos

Subkategorija	Citatos
Per daug abstraktu	„Sritis sudėtinga, jie patys nežino konkrečiai, ko nori <...>, bet tu turi vertinti <i>story point'ais</i> – man tokie dalykai nepatinka.“ (E4);
Reikia didesnio pasiruošimo	„Kiekvieną kartą važiuodama derini, klausinėji <...>, bet vis tiek įvyksta netikėtų situacijų.“ (E1); „ <i>Workshop'ą</i> padaryti daug sunkiau <i>online'u</i> <...>, kai riboja laikas.“ (E1);
Reikalauja daug resursų	„Neturi resursų nuskristi pas klientą <...>, turi pusę valandos tą ir tą dieną, tą ir tą valandą <i>online</i> – viskas“ (E1); „Tam reikia resursų – pakviesti visus į vieną vietą <...>, dabar per tris šalis dirbam.“ (E1); „Kitur nėra priemonių – kambarys suspaustas, net atsistoti nėra kur, kompiuterio prijungti negali.“ (E1); „Galima padaryti gana retai ir tam reikia įsitraukimo gana didelės žmonių grupės.“ (E2); „Tai gana ilgas ir gana brangus procesas <...>, daromas tik prieš didelį projektą.“ (E2); „Kad visi rastų laiko <i>workshop'ui</i> , gali užtrukti kelias savaites – jų laikas brangus, visi užsiėmę.“ (E2);
Neproduktyvu didelėms grupėms	„Kai didelės grupės <...>, nėra produktyvus tas vienas <i>workshop'as</i> .“ (E1); „Reikia sužiūrėti, kokio tipo žmonės <...> kad aukštesnio rango žmogus nenumenkintų specialisto.“ (E4);

11 lentelėje pateikti rezultatai atskleidžia iššūkius, su kuriais tyrimo dalyviai susiduria rinkdami reikalavimus, taikydami dirbtuvių metodą. Vienas iš jų – per didelis abstraktumas: „sritis sudėtinga, jie patys nežino konkrečiai, ko nori <...>, bet tu turi vertinti *story point'ais* – man tokie dalykai nepatinka.“ Taip pat išryškėja pasiruošimo būtinybė: „kiekvieną kartą važiuodama derini, klausinėji <...>, bet vis tiek įvyksta netikėtų situacijų“, o nuotoliniu būdu organizuoti dirbtuves dar sudėtingiau, nes „*workshop'ą* padaryti daug sunkiau *online'u* <...>, kai riboja laikas“. Reikšmingas iššūkis – didelis poreikis resursams: „neturi resursų nuskristi pas klientą <...>, turi pusę valandos tą ir tą dieną, tą ir tą valandą *online* – viskas“, „tam reikia resursų – pakviesti visus į vieną vietą <...>, dabar per tris šalis dirbam“ bei „tai gana ilgas ir gana brangus procesas <...>, daromas tik prieš didelį projektą“. Galiausiai, dirbtuvės ne visuomet pasiteisina dirbant su didelėmis grupėmis, nes „kai didelės grupės <...>, nėra produktyvus tas vienas *workshop'as*“ ir reikia apgalvoti dalyvių kombinacijas, kad „aukštesnio rango žmogus nenumenkintų specialisto.“

Siekiant nustatyti, kokios priemonės ekspertams padeda kokybiškai ir tiksliai surinkti reikalavimus, buvo sudarytas svarbiausių reikalavimų surinkimo aspektų subkategorijų modelis, apimantis 20 svarbiausių punktų. Svarbiausių reikalavimų surinkimo aspektų vizualinis modelis pateikiamas 18 paveiksle.



18 pav. Svarbiausių reikalavimų surinkimo aspektų vizualinis modelis

12 lentelėje pateikiamos ekspertų išskirtos svarbiausių reikalavimų surinkimo aspektų subkategorijos ir jas pagrindžiančios citatos.

12 lentelė. Svarbiausių reikalavimų surinkimo aspektų subkategorijos ir jas pagrindžiančios citatos

Subkategorija	Citatos
Nuolatinis prioritetų vertinimas	„Tu atneši santrauką, pagal ją prioritetizuojiesi ir dirbi.“ (E1); „Jei kažko nepakviete, gali būti, kad reikalavimas nebus pasirinktas kaip prioritetas.“ (E1); „Ar tikrai ta dalis yra pati svarbiausia? <...> bet klientui dabar svarbiausia visai kitas dalykas yra.“ (E2); „Yra metodologijos, kaip vertinti reikalavimus – RICE. Jos padeda suprasti, kas su mažai pastangų duos didelę vertę.“ (E3); „Jei klientas prigeneravo daug <i>change request</i> ’ų, mes viską metame į vieną krūvą ir darome prioritetus.“ (E4) „Kai kurio <i>feedback</i> ’o net nereikia daryti.“ (E5); „Reikalavimai ir projekto apimtis nuolat kinta. Kai kurie gali tiesiog išnykti.“ (E6).
Smulkmeniškumo lygio nustatymas	„Dar kartą žiūrime didįjį paveikslą, kas pasimeta, kur nesutampa, o tada einame į detales.“ (E1); „Smulkmenose kažkas visada bus praleista, bet dažniausiai klientas to net nepastebi.“ (E1); „Ant tiek nusismulkinti neverta, nes tada pametame, ko iš tikrųjų siekiame, ir sukuriame galimybę suklysti.“ (E1); „Yra riba – kiek <i>edge case</i> rašyti reikalavimuose, o kiek palikti testavimui.“ (E3); „Word dokumente viskas išdėliota, bet tiek daug prielaidų, kad jei pirmame žingsnyje nepaeis, viskas sugrius.“ (E5).
Aiškus komunikavimas	„70 procentų vis tiek tai vyksta komunikacija atvira tokia. Ar kalba, ar kažkokios čatai, ar laišškai. Ir tai, ką tu padarai kaip analitikas, kuri griaučius, kad sudėti į struktūrą, žinai, kažkokią“ (E1) „Tai tą bendravimą, žinai, bendrauti, kad galėtum, nu, pakankamai turi empatišką būti. Empatią, žinai, ką duoda, kad jeigu tu bendrauti mėgsi, reiškia, jis tau rūpi, nu, visas to, žinai, ir iš tikrųjų, analitika gerą daro, žinai, va, tas rūpėjimas. <...> 70 procentų tavo darbe, žinok, vien tik bendravimas yra.“ (E1); „<...> reikia surasti reikiamus žmones, kuriuos tu galėtum paklausti <...>“ (E2); „Darom, nusiunčiam jiems kažkokį <i>summary</i> , ką mes apsitarem.“ (E2); „Komunikacija tai yra labai svarbu tam, kad užmegzti gerą ryšį su klientu ir kad, kaip pasakyti, jį prakalbinti. <...>“ (E2); „Kad tiek surasti teisingą būdą, kaip surinkti, tiek surasti teisingą būdą, kaip perduoti komandai <...>“ (E3); „Jeigu tu matai, kad yra perviršis <...> visuomet reikia sakyti iš anksto <...>“ (E4); „Faktiškai jau

	aiškinimosi metu pasakom, kad mes neskaitom tarp eilučių, gražia forma.“ (E4); „Bet čia ne kiek pačio pratimo, čia ir šiaip, kiek žmogus pratęs kalbėti ir leisti į detales.“ (E5); „Sako, reikia ta daryti, ten keist, keiti, informuoti, kad galbūt, žinai, įsivertinsim tau, galbūt nusistums tas <i>deadline'as</i> <...>“ (E5); „Komunikavimo įgūdžiai. Bet čia mes šnekame ne tik apie tai, kaip tu kalbi, bet ir apie tai, kaip tu nori klausyti ir kaip tu moki klausyti ir kaip tu moki ne tik klausyti, bet ir išgirsti. <...>“ (E6).
Aiškios reikalavimų struktūros kūrimas	„70 procentų vis tiek tai vyksta komunikacija atvira <...> Ir tai, ką tu padarai kaip analitikas, kuri griaučius, kad sudėti į struktūrą.“ (E1); „<...> mes jau braižomės <i>journey map</i> <...> tada gali jau apjungti tą informaciją, išskelti, kas pas visus atsikartoja.“ (E1); „<...> jau kai tu turi tuos atsakymus, tau reikia vis tiek pristatyti komandai. <...> Tu negali pasakoti pasakojimų. Analitikas ir yra, kad jis, žinai, sukonsoliduoja kažkokį <i>output'ą</i> .“ (E1); „<...> analitikas turėtų nustatyti terminus, kaip šnekėt, ir nustatyti struktūrą. <...> Nes iš to lengviau orientuotis ir visai komandai.“ (E1); „<...> tiek labai padeda vizualai. <...> dizaino <i>sketchai</i> greičiau negu <i>bullet point'ų</i> sąrašas duoda supratimą.“ (E3); „Visuomet yra tas <i>consistency</i> dalykas <...> jeigu dirba keli BA, svarbu susišnekėti, kad kokybišką dokumentą gautum.“ (E4); „Noras yra susidėlioti struktūrą, kad galėtume proaktyviai spręsti, kaip judės projektas.“ (E5); „Tikslas yra nustatyti <i>definition of ready</i> <...> kad įvest reikalavimų surinkimo ir aprašymo standartą.“ (E5); „Kai atsiranda aiškumas, esu struktūrizuota <...> bet norėčiau gebėti tai daryti nuo pat pradžios, kai yra neaiškumo laipsnis.“ (E6).
Aiškios vizijos turėjimas	„Tikslas yra pagerinti apibrėžimą, dėl ko mes darom projektą, kokią problemą sprendžiam, kokie mūsų sėkmės kriterijai ir kaip juos matuosim.“ (E1); „Turiu viziją, vieną aiškią žinutę, kuri suteikia žmonėms teisingą interpretaciją, išpildant reikalavimus.“ (E1); „Dažnai nesusitariama ne dėl reikalavimų, o dėl pačios idėjos – kokią vertę sistema turėtų kurti.“ (E1); „Reikia labai aiškios žinutės apie dabartinę situaciją, tikslus ir kryptį, kad inžinieriai ir analitikai galėtų užpildyti trūkstamus reikalavimus.“ (E1); „Didžiausia problema – turėti žmogų, kuriam rūpi produktas ir kuris supranta, kur einama. Jei jis tai aiškiai pasako komandai, problemos išsprendžia.“ (E1); „Svarbiausia suprasti problemos esmę – kitaip sprendžiam tik pasekmes, o ne pačią problemą.“ (E2); „Jei klientas nesupranta mūsų pateiktos informacijos, gali būti, kad mes nesupratom problemos.“ (E2); „Pradžioje atliekam tik <i>high-level</i> analizę, kad aiškiai matytume projekto dalis ir planuotume į priekį bent 2-3 <i>sprint'ams</i> .“ (E2); „Produkto strategija diktuoja reikalavimus – tai nuolatinis ciklas.“ (E3); „Sėkmės metrikos apima verslo rodiklius, vartotojų patirtį ir kitus kiekybinius bei kokybinius aspektus.“ (E3); „Svarbiausia, kad vertės suvokimas būtų pasidalintas su visa komanda ir perduotas per reikalavimus.“ (E3); „Reikalavimų rinkimas turi grįžti prie poreikių analizės – kodėl vartotojui to reikia?“ (E5); „Jei pradėsime rinkti reikalavimus kaip „ <i>system should do this</i> “, įstrigsime norų sąrašė, o ne vertės kūrimas.“ (E5); „Svarbiausia – aiškiai suprasti, kokią problemą sprendžiame. Jei tai neaišku, visi kiti žingsniai tampa chaotiški.“ (E5); „Jei nesugebame atskirti skirtingų problemų, projektas nejuda į priekį, nes viskas suplakama į vieną.“ (E5); „Klientai dažnai įsivaizduoja, kad žino, kaip turi būti įgyvendinta, bet pirmiausia reikia suprasti, ką jie iš tikrųjų bando pasiekti.“ (E6).
Aiškų išspręstos problemos apibrėžimas	„Tikslas – aiškiai apibrėžti, kokią problemą sprendžiame, kokie mūsų sėkmės kriterijai ir kaip juos matuosim.“ (E1); „Svarbu įsivertinti, kada problema laikoma išspręsta – tai gali būti skaičiai ar kiti kriterijai, kurie padeda <i>susi'alignin'inti</i> su klientu ir komanda.“ (E2);
Apimties išgryninimas ir ribojimas	„Kai jau mažėja <i>change request'ų</i> , <...> produktas pasiekė brandą.“ (E1); „Lūkesčių suvaldymas <...> vyksta per pristatymus, tačiau <...> atsiranda netikėtų technologinių ar papildomų reikalavimų.“ (E4); „Geriausia signalizuoti apie perviršį iš anksto, <...> ne projekto pabaigoje.“ (E4); „BA turi stebėti <i>functional story</i> ir <...> jei yra <i>estimate</i> , reiškia, kad pakeitimas <...> buvo įtrauktas į refinamentą.“ (E4); „Dalis užklausų atmetama <...>, siekiant, kad ateitų geriau apgalvotos idėjos.“ (E5); „Kai pokyčiai plečia apimtį, svarbu <...> įvertinti jų poveikį ir informuoti klientą.“ (E5); „ <i>Feedback'ą</i> geriausia <i>log'inti</i> atskirai ir <...> spręsti, kas patenka į <i>development backlog'ą</i> .“ (E5); „Kartais būtina apriboti kliento <i>scope</i> , <...> nes jisai ten gali labai plėstis ir fantazuoti labai daug. Kur tu jau supranti, kad tu netilpsi į laiką.“ (E6);
Pasitikėjimo pagrįsto santykio kūrimas	Šalti santykiai padeda lengviau spręsti klausimus. <...> Jei santykiai šalti, tampa sudėtinga gauti pagalbą.“ (E2); „Komunikacija svarbi kuriant ryšį su klientu ir prakalbinant jį.“ (E2); „<...> kaip būti patraukliu klientui.“ (E3); „Jei su klientu yra pasitikėjimas, nereikia papildomo proceso. <...> Jei pasitikėjimo trūksta ar yra daug komandų, <...> atsiranda procesinė našta.“ (E5); „<...> kad tu turi kaip ir sukurti tokį labai gerą ryšį su klientu <...>.“ (E6); „Analitiko darbas – rasti bendrą kalbą su klientu, kad pilnai jį išgirsti ir kad jis pasitikėtų.“ (E6); „Geriausia dirbti per diskusijas ir

	aktyvias sesijas, o ne per formalius pristatymus.“ (E6); „Klausantis diskusijų galima pagauti <...> ir ištraukti pasislėpusius reikalavimus.“ (E6);
Aiškus kuriamos vertės suvokimas	"Tavo tikslas nėra vien tik rinkti reikalavimus, bet suprasti kliento problemą ir sukurti geriausią sprendimą, kuris atneštų didžiausią vertę.“ (E5); „ <i>Product owneris</i> fokusuojasi nebe į pačius reikalavimus, o į vertę, kurią nori atnešti. <...> Svarbu įvertinti, kiek sprendimas atliepia poreikius.“ (E5);
Analitiko patirtis ir domeno išmanymas	„<...> kuo daugiau patirties analitikas turi apie panašius projektus <...>, tuo greičiau tai vyksta. Jis netgi <...> gali pasiūlyti vieną ar kitą variantą <...>. Tai tas <i>know-how</i> apie domeną, problemą, verslą yra labai svarbi dalis.“ (E2); „<...> loginis mąstymas padeda surinkti reikalavimus <...>. Taip pat reikia psichologinių žinių <...>. Kartais ir konfliktus suvaldyti <...>.“ (E4); „Atidumas detalėms praverčia, bet loginis mąstymas – būtinas.“ (E5); „Patirtis labai padeda. <...> Kai turi patirties, greičiau pamatai dalykus <...> ir žinai, kur gali kilti papildomų klausimų.“ (E6);
Aiškių sąvokų naudojimas	„<...> reikėtų labai atidžiai vartoti žodžius ir nevartoti perteklinių visokių žodžių tam, kad kuo mažiau turėti dviprasmiškumo.“ (E6);
Kurti bendrą paveikslą	„<...> mūsų tikslas yra surinkti iš potencialių klientų poreikį <...> kaip jie dirba, kokias problemas turi.“ (E1); „Aš negaliu eiti į detales, kol nesuprantu, kaip viskas jungiasi. Kai turiu didįjį paveikslą, tada einu detaliau.“ (E1); „ <i>User story mapping</i> padeda suprasti, ar pasiekėme norimą rezultatą <...>. Kai turi <i>user story map</i> 'o backbone'ą, gali žiūrėti, kaip <i>slice</i> 'inti iki minimalaus <i>deliverable</i> .“ (E5); „Didelio paveiksliuko gebėjimas sudėliot, darbas neapibrėžtume“ (E5);
Šakninio poreikio identifikavimas	„<...> pirmiausia gavę reikalavimus, turime grįžti prie tikrojo jų poreikio <...>. Vartotojai mano, kad jų siūlomas sprendimas veiks, bet dažnai taip nėra. <...> Surinkti reikalavimus viena, tačiau svarbu iš jų sukurti tinkamą sprendimą.“ (E5); „Dažniausiai vartotojai pasako, ką padaryti, bet ne kodėl <...>. Iššūkis yra išsiaiškinti tikrąjį poreikį, nes jis dažnai nebūna aiškiai išreikštas.“ (E5).
Kliento pasirengimas	„Kiek jie yra technologiškai išprusę <...> kad kuo produktyviau ir daugiau ištrauktum.“ (E1).
Lyderio svarba	„Labai svarbu, kas veda projektą. Turiu viziją, turiu tikėjimą. <...> tokia graži žinutė dažniausiai suteikia žmonėms interpretaciją teisingą išpildant reikalavimus.“ (E1);
Reikalavimų validavimas	„<...> visą laiką jausti, kur mes judam, tą kryptį, tą viziją.“ (E1); „vienas pasisako nuomonę, antras, užtvirtini, trečias tvirtinu, visi laimingi.“ (E1); „<...> visi pasako, tu visus paklausai ir pagal tai, kurie daugiausia atsikartoja, nusakai, kad tas tai tikrai, greičiausiai, visiems reikalinga.“ (E1); „<...> tu suvedi visus į vieną vietą ir realiai pasitvirtini taip reikalavimus.“ (E1); „<...> pasivaliduoji su kitų šaltinių ar keliais ir nusprendi, kad jis validus kaip reikalavimas.“ (E1); „<...> nuvaryti vėl pas klientą su tuo sprendimu ir sakyti – žiūrėk, va turim tokį sprendimą, veikų taip ir taip, ar manot, kad tai išspręstų jūsų problemą?“ (E2); „<...> grįžtam pas juos, tai va čia yra būtent tam, kad pasitikrinti, ar tikrai mes šnekam apie tą patį dalyką, ar tas sprendimas iš tikrųjų atitinkamas jų problemai.“ (E2); „<...> mes darom prielaidą, kad tie dalykai atneš vertę. Tada implementuojam, verifikuojam.“ (E3); „<...> vidinė validacija tai nėra galutinė validacija <...>, bet geriau negu neturėti jos.“ (E3); „<...> visuomet reikia prasivaliduot.“ (E4); „ <i>Acceptance criteria</i> sudėdami, jie yra siunčiami verifikuoti ir pavaliduot.“ (E4); „<...> surenki sprendimą, tu turi pasitikrint, ar visgi <i>good</i> ar <i>bad</i> <...> negali vėl ten bėgti lėkti akis išdeges.“ (E5); „Paprastai tų pačių dalykų kitą kartą klausai ir ne vieną kartą, ir skirtingai apvertęs, duodi kažkokias praktines situacijas iš jų veiklos.“ (E6); „Visuomet geriau pasisaugot, jeigu reikia patvirtinimo judėti toliau, visuomet turi turėti kažkokį tai patvirtinimą.“ (E4); „kuo mažiau laiko turi, tai tuo mažiau validuosi, gal mažiau detalių aprašysi, gal sprendimo daug neišgryninsi ir turi, aišku, tiesiog galimybę prasišauti.“ (E5); „Feedbacko davimas yra ta vieta, kur užtvirtini, ar čia OK, ar ne.“ (E5); „Dauguma kompanijų, kurios nėra iš <i>software product</i> , vis dar masto projektiškai <...> jiems sunku operuoti šiame kontekste ir validuoti reikalavimus“ (E5); „Kai kurie dirba produktiškai <...> tikslas – priimti geriausią sprendimą ir išsivaliduoti, kuris variantas bus geriausias“ (E5); „Reikalavimai dažnai nekvestionuojami, nors juos būtina kvestionuoti <...> ar tas, kas sugalvojo, tikrai viską gerai įsivertino?“ (E5); „Kartais tų pačių dalykų klausai kelis kartus, skirtingai suformulavęs <...> geriausias įrodymas, kad supratai – kai klientas naudoja sistemą ir džiaugiasi“ (E6).
Reikalavimų vizualizavimas	„Kartais dar net neturint <i>backlog'o</i> jau braižomės <i>journey map'ą</i> <...> labai padeda suprasti skirtingų klinikų įpročių <i>flow</i> .“ (E1); „Vieni nori viską dėlioti vizualiai – lapukais, piešinukais“

	(E1); „Sėkmei daug prisideda įrankiai, leidžiantys vizualiai pasibraižyti, padiskutuoti, susirašyti mintis.“ (E1); „Vizualai – greitesnis būdas suprasti nei <i>bullet point'ai</i> , ypač kai kalbame apie dizaino <i>sketch'us</i> .“ (E3); „Geriausia – viską vizualizuoti <...> kai parodai <i>flow</i> diagramą, ateina tas „aha“ momentas.“ (E5); „Viskas prasideda nuo interviu, o tada jau einama prie <i>mock-up'ų</i> ir pavyzdžių.“ (E5); „Prototipas gali būti piešinukas, <i>clickable</i> versija ar net skaičiavimo logikos pavyzdys.“ (E5); „Vienas paveikslukas vertas tūkstančio žodžių – braižau diagramas, kad būtų mažiau „ne taip padarė“ situacijų“ (E6); „Vizualika padeda sumažinti riziką, turėti netiksliai įgyvendintus reikalavimus.“ (E6); „Aktyviai naudojame BPMN – ypač kai reikia suvaldyti alternatyvius scenarijus.“ (E6); „Vizualizuoti galima ir ant servetėlės – svarbiausia, kad visi vienodai suprastų.“ (E6); „Kuo daugiau vizualikos – tuo lengviau suprasti vieniems kitus.“ (E6); „Mėgstu greitai susimodeliuoti <i>wireframe'us</i> , naudoju UML, <i>activity</i> diagramas – jos daug atskleidžia apie visą sistemos lifecycle.“ (E6); „Kai diskusija vyksta gyvai, su pieštuku prie schemos, užsimezga aktyvi, įtrauki diskusija <...> esu diskusijų, o ne pristatymų šalininkas.“ (E6).
Komandos įtraukimas į reikalavimų validavimą	„Jeigu komanda nesupranta, ką reikia padaryti, reiškia, kad informacija buvo surinkta nekokybiškai.“ (E2); „ <i>Refinement'e</i> komanda užduoda klausimus <...> jeigu jų daug – reikalavimai dar neaiškūs ir reikės papildomų iteracijų.“ (E2); „Prieš refinementą geroji praktika iš anksto pasidalinti juodraštinio variantu, kad komanda pasiruoštų klausimus.“ (E3); „Po diskusijos su komanda gali būti dar vienas žingsnis – patikrinimas su klientu, ar visi vienodai suprantam, kas parašyta.“ (E3).
Komandos įtraukimas į apimtį vertinimą	„I komandą atnešama interviu santrauka, pagal kurią prioritetizuojami ir tvarkomi reikalavimai.“ (E1); „Detales susirinkus, aprašyti kiekvieną iš tų storių, kaip ir reikalavimus pačius, ir tada jau perduoti komandai įvertinti.“ (E2); „Su komanda vertinam, kiek tai užtruks, įsivertinam <i>story point'ais</i> ir planuojam.“ (E2).
Techninis kliento atstovas	„Turime techninį žmogų iš kliento pusės, kuris padeda suprasti infrastruktūrą, esamus apribojimus ir galimybes.“ (E1); „Dažniausiai būna projekto atstovas ir <i>Subject Matter Expert</i> – žmogus, kuris pats susiduria su problema ir žino specifiką.“ (E2); „Viduje kliento įmonės reikia žmogaus, kuris tave nukreiptų <...>.“ (E2); „Techniniame lygyje gali būti, kad jau ištraukia visai kiti žmonės <...>: IT atstovai, techniniai žmonės iš jų pusės.“ (E2); „Būdavo tokių, kad neprileisdavo tiesiogiai prie programuotojų kitos komandos <...> per dieną su programuotoju išsprendžiau viską, ko savaitę ieškojau per <i>menedžerį</i> .“ (E4).

12 lentelėje pateikti rezultatai atskleidžia ekspertų išskirtus svarbiausius reikalavimų surinkimo aspektus. Nuolatinis prioritetų vertinimas išryškėja kaip esminis procesas, kai „reikalavimai ir projekto apimtis nuolat kinta“, o „jei klientas prigeneravo daug *change request'ų*, mes viską metame į vieną krūvą ir darome prioritetus“. Kitas svarbus aspektas – gebėjimas nusistatyti tinkamą smulkmeniškumo lygį: „ant tiek nusismulkinti neverta, nes tada pametam, ko iš tikrųjų siekiame, ir sukuriam galimybę suklysti.“ Aiškus komunikavimas dominuoja kaip kritinė kompetencija: „70 procentų tavo darbe, žinok, vien tik bendravimas yra“, o „komunikavimo įgūdžiai – tai ne tik kaip tu kalbi, bet ir kaip tu moki klausyti ir išgirsti“. Taip pat svarbu kurti aiškią reikalavimų struktūrą – „analitikas ir yra, kad jis, žinai, sukonsoliduoja kažkokį *output'ą*“ bei turėti aiškią viziją: „turiu viziją, vieną aiškią žinutę, kuri suteikia žmonėms teisingą interpretaciją, išpildant reikalavimus.“ Pažymima būtinybė aiškiai sukurti išspręstos problemos apibrėžimą – „svarbu įsivertinti, kada problema laikoma išspręsta“ ir išgryninti apimtį – „kartais būtina apriboti kliento *scope*, <...> nes jisai ten gali labai plėstis.“ Stiprus, pasitikėjimu grįstas, santykis su klientu palengvina darbą: „jei su klientu yra pasitikėjimas, nereikia papildomo proceso.“ Vertės suvokimas išskiriamas kaip prioritetas – „tavo tikslas nėra vien tik rinkti reikalavimus, bet suprasti kliento problemą.“ Be to, akcentuojamas analitiko patirties ir domeno išmanymo svarba – „kuo daugiau patirties analitikas turi apie panašius projektus <...>, tuo greičiau tai vyksta.“ Kiti aspektai – aiškių sąvokų vartojimas, gebėjimas matyti „didįjį paveikslą“, identifikuoti šakninį poreikį, įvertinti kliento pasirengimą, lyderio svarba, reikalavimų verifikavimas („visuomet reikia prasivaliduot“), vizualus informacijos pateikimas („vienas paveikslukas vertas tūkstančio žodžių“), komandos įtraukimas tiek į validavimą („jeigu komanda nesupranta, ką reikia padaryti, reiškia, kad informacija

buvo surinkta nekokybiškai“), tiek į apimtį vertinimą („su komanda vertinam, kiek tai užtruks“), bei techninio atstovo buvimas kliento pusėje: „turime techninį žmogų iš kliento pusės, kuris padeda suprasti infrastruktūrą.“

4.2. Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo diskusija

Siekiant palyginti mokslinėje literatūroje identifikuotus reikalavimų inžinerijos iššūkius su nustatytais empirinio tyrimo metu, buvo sudaryta 13 lentelė. Kadangi iššūkių apibūdinimai, rasti mokslinėje literatūroje ir suformuluoti empirinio tyrimo metu šiek tiek skiriasi, todėl vertinant jų atitikimą buvo remiamasi iššūkio apibūdinimu, atitikimu pagal prasmę. Taip pat, kai kurių iššūkių paminėjimas gali būti suprantamas iš konteksto, randamo literatūroje arba empiriniame tyrime, tad lentelėje pateikiami iššūkių apibrėžimai gali skirtis nuo pateiktų literatūroje arba empiriniame tyrime savo formuluote, tačiau atitinka pagal prasmę.

13 lentelė. Literatūroje ir empiriniame tyrime nustatytų reikalavimų inžinerijos iššūkių palyginimas

Iššūkis	Pateikta mokslinėje literatūroje	Nustatyta empiriniame tyrime
Ignoruojami kokybiniai reikalavimai	X	
Dokumentavimo trūkumas	X	X
Netinkami prioritizavimo metodai	X	
Prastas pokyčio valdymas	X	X
Prastas reikalavimų aprašymas	X	X
Netikslus darbo sąnaudų įvertinimas	X	X
Kliento nepasiekiamumas	X	X
Kliento žinių trūkumas	X	X
Silpni kliento atstovai	X	X
Skirtingi suinteresuotieji kliento pusėje	X	X
Netinkama architektūra	X	
Netinkama komunikacija	X	X
Netikslūs reikalavimai	X	X
Neišsamūs reikalavimai	X	X
Programinės įrangos kūrimas – tiriamasis darbas		X
Analizės klaidos brangiai kainuoja	X	X
Įsisenėjusios kliento praktikos		X
Neteisingos prielaidos		X
Ribota prieiga prie įrankių	X	X
Nėra bendro paveikslo, vizijos		X

Perteklinė informacija		X
Senos sistemos palaikymas		X

Apibendrinus mokslinės literatūros analizės ir empirinio tyrimo rezultatus, iš viso identifikuoti 22 skirtingi reikalavimų inžinerijos iššūkiai. Dauguma literatūroje identifikuotų reikalavimų inžinerijos iššūkių (Rasheed et al., 2021, Hoy & Xu, 2023; Inayat et al., 2015; Kasauli et al., 2021) buvo patvirtinti empirinio tyrimo metu. Vis dėlto tyrimo dalyviai neidentifikavo ignoruojamų kokybinių reikalavimų, netinkamos architektūros kaip iššūkio, taip pat netinkamas prioritetizavimo metodų pasirinkimas nebuvo tyrimo dalyvių išskirtas kaip iššūkis, keliantis problemų reikalavimų inžinerijos procesuose. Empirinio tyrimo metu buvo atrasti nauji iššūkiai, kurie nebuvo nustatyti analizuojant mokslinę literatūrą. Programinės įrangos kūrimą tyrimo dalyviai įvardijo kaip tiriamąjį darbą, o tai reiškia, jog kuriama sistema ar jos dalis dar niekur neegzistuoja, todėl susiduriama su tokiais pačiais iššūkiais kaip tiriamajame darbe – iš pradžių susiduriama su daug neaiškumų, daromos prielaidos, eksperimentuojama, tačiau galutinis rezultatas gali būti ne toks, kokio tikėtasi projekto pradžioje. Iš to išplaukia kitas mokslinėje literatūroje neaptiktas iššūkis – neteisingos prielaidos. Tyrimo dalyviai pabrėžė, jog pasitaiko, kad tam tikras funkcionalumas yra kuriamas remiantis interpretacijomis ar prielaidomis, kurios, kaip dažnai pasitaiko, yra klaidingos. Dėl to ypatingai svarbu gera komunikacija su klientu, reikalavimų patikrinimas, interpretacijų suvienodinimas. Su neteisingomis prielaidomis susijęs tyrimo metu atrastas iššūkis – kliento taikomos įsisenėjusios darbo praktikos neleidžia tinkamai patikrinti reikalavimų ir įtraukti grįžtamąjį ryšį į sistemos kūrimą, dėl ko vėliau gali nukentėti projekto rezultatai. Literatūroje aptiktas prieigos prie tinkamų įrankių iššūkis, kurį ekspertai taip pat įvardijo empirinio tyrimo metu. Šis iššūkis pasireiškia, jeigu nesutampa kliento primygtinai reikalaujami naudoti įrankiai bendravimui, vizualizavimui, reikalavimų valdymui. Ekspertams tokiu atveju tenka adaptuotis prie senesnių ir situacijai neviseškai tinkamų darbo įrankių, kurie lėtina darbo procesą, palyginus su tinkamų įrankių naudojimu. Tyrimo dalyviai akcentavo didelio poveikslo arba vizijos nebuvimą kaip vieną didžiausių iššūkių, kadangi kuriant sistemas yra ypatingai svarbu žinoti kryptį ir ilgalaikį tikslą, ne tik apsiriboti vienos sistemos dalies reikalavimais, kurie nesukuria bendro vaizdo.

Remiantis literatūros analizėje identifikuotais pagrindiniais veiklos rodikliais, rezultatus galima palyginti su empirinio tyrimo rezultatais. Kaip ir įvardinta literatūroje (Gadatsch, 2023; Phokwane & Arnesh, 2021; Hollender et al., 2016) , tyrimo dalyviai išskyrė kokybinius ir kiekybinius pagrindinius veiklos rodiklius, kuriuos taiko reikalavimų inžinerijos procesų stebėjimui. Skaitine reikšme išreiškiamus rodiklius ekspertai išskyrė: proceso trukmę, atvirų užklausų skaičių, klaidų skaičių, dokumentacijos veiklų trukmę. Vis dėlto daugiau identifikuota kokybinių rodiklių, kuriuos pamatuoti skaitine verte neįmanoma, reikia taikyti tam tikrus metodus, padedančius juos paversti į skaitinę vertę. Tai tokie rodikliai kaip klientų rekomendavimo indeksas (NPS), reikalavimų išbaigtumas, pasiektas tikslas, pokyčio atsekamumas, kliento pasitenkinimas. Klientų rekomendavimo indeksas (NPS) parodo, kiek klientai linkę rekomenduoti suteiktas paslaugas (Content Square, 2024). Kadangi reikalavimų inžinerija daugiau susijusi su kūrybiniu darbu nei su tikslumu, todėl ir rodikliai yra daugiau kokybiniai ir sunkiau pamatuojami. Kaip ir minima literatūroje (Tømmerås et al., 2020; Wetzstein, 2016; Pidun & Felden, 2011; Calabro et al., 2015), taip ir ekspertai pabrėžė, jog labai svarbu derinti kiekybinius ir kokybinius pagrindinius veiklos rodiklius, o nustatyti rodikliai būtų pamatuojami ir netaptų siekiamybe. Taip pat ekspertai išskyrė taikomą SPACE metodą, kuris padeda pamatuoti komandos patirtį ir efektyvumą. SPACE metodas (angl. *Satisfaction, Performance, Activity, Communication, Efficiency*) yra rodiklių rinkinys, leidžiantis įvertinti komandų produktyvumą ir veiklos efektyvumą per penkias

pagrindines sritis: komandos pasitenkinimą darbu, rezultatyvumą, aktyvumą, komunikacijos efektyvumą bei efektyvumą. Šis metodas suteikia galimybę identifikuoti komandų stipriąsias ir tobulintinas sritis bei efektyviau valdyti reikalavimų inžinerijos procesus (Codacity Platform, 2024).

Lyginant mokslinėje literatūroje aptartus procesų tobulinimo metodus su empirinio tyrimo metu identifikuotais metodais, abiem atvejais išryškėja vienas plačiai naudojamas metodas – šakninės priežasties analizė (angl. *Root Cause Analysis*). Daugumoje atvejų šakninės priežasties analizei naudojama „5 kodėl“ technika. Šios technikos nauda buvo pabrėžta tiek mokslinėje literatūroje (Wolniak & Grebski, 2023; Shen et al., 2017; Benjamin et al., 2015; Kataoka et al., 2011; Keyser & Pooyan, 2023), tiek empirinio tyrimo dalyvių – galimybė identifikuoti esmines problemų priežastis. Vis dėlto ekspertai nurodė ir tam tikrus šios technikos iššūkius: dažnai žmonėms ji kelia psichologinį diskomfortą, pasipriešinimą, ypač tais atvejais, kai analizuojamos darbuotojų klaidos ar priimti sprendimai. Empirinio tyrimo metu taip pat nustatyta, kad ekspertai „5 kodėl“ techniką taiko ne tik procesų gerinimui, bet ir reikalavimų rinkimo procese, kai siekiama išsiaiškinti tikruosius klientų poreikius. Analizuojant mokslinę literatūrą (Alrabadi et al., 2023; Botezatu et al., 2019; Korecki & Adámková, 2020; Bilsel & Lin, 2012; Wolniak, 2019) aptikta Ishikawa diagrama empiriniame tyrime ekspertų nebuvo minima, tačiau savo esme taip pat priskiriama prie šakninės priežasties analizės metodų, nes leidžia struktūriškai nustatyti pagrindines problemų priežastis. Literatūroje (Boonpracha et al., 2023) ir informaciniuose šaltiniuose (Elmansy, 2015) minimas SCAMPER metodas empirinio tyrimo metu ekspertų nebuvo įvardintas – šis metodas gali būti vertinamas kaip galimybė, kurios tyrime dalyvavę ekspertai kol kas nėra išbandę praktikoje. Vizualizavimas, kaip procesų tobulinimo metodas, buvo identifikuotas tiek literatūros analizėje (Phokwane & Arnesch, 2021; Gadatsch, 2023; Pergl et al., 2019), tiek ir empirinio tyrimo metu – ekspertai pažymėjo, kad dažnai pakanka paprastų eskizų ar scheminių piešinių, aiškiai perteikiančių pagrindinę mintį. Vis dėlto sudėtingesniems procesams vizualizuoti dažnai pasirenkamos BPMN (angl. *Business Process Model and Notation*) diagramos, kurios taip pat buvo identifikuotos literatūroje kaip viena iš standartizuotų procesų vizualizavimo priemonių.

Empirinio tyrimo metu nustatyta, jog reikalavimų inžinerijos ekspertai naudoja tam tikras praktikas ir darbo metodus, kurie prisideda prie sklandesnio proceso. Kaip esmines praktikas ekspertai pabrėžė literatūroje (Anifa et al., 2024; Inayat et al., 2014; Rasheed et al., 2021) taip pat išskiriamus aspektus – nuolatinį grįžtamojo ryšio rinkimą ir darbą iteracijomis. Grįžtamojo ryšio rinkimas ir iteratyvus darbas leidžia pastebėti klaidas arba lūkesčių nesutapimą ankstyvose stadijose – ekspertų paminėtas perstūmimo kairėn (angl. *Shift Left*) principas užtikrina, kad taisymai būtų atliekami, kol jų įtaka yra minimali. Iš kitos pusės – rezultato pristatymas etapais leidžia klientams pajauti kuriamo produkto vertę, stebėti progresą ir tuo pačiu aktyviai įsitraukti į kūrimą. Nors pagal „Agile“ metodologiją dokumentacijos kiekis turėtų būti minimalus, vis dėlto empirinio tyrimo dalyviai pažymėjo, jog ypatingai svarbu dokumentuoti esminius sprendimus, pakeitimų priežastis ir sudėtingesnių sistemos dalių veikimo logiką. Pabrėžiama, jog dokumentacijos nauda išryškėja perduodant projektą palaikymui arba keičiantis projekto komandai, kadangi kartu su žmonėmis išeina ir žinios apie sistemą. Anot ekspertų, dokumentacijos veiklos turėtų užimti ne daugiau 20% visų reikalavimų inžinerijos procesų – tai leidžia išgauti optimalų darbo ir naudos santykį ruošiant dokumentaciją. Empirinio tyrimo respondentai nurodė, jog procesų standartizavimas – tai svarbus aspektas, siekiant sumažinti komandų adaptacijos laikotarpį prisijungiant prie naujo projekto. Standartizuoti reikalavimų aprašai ir procesai įneša aiškumo ir struktūrą, o tai gerokai sutaupo laiko sąnaudas. Kaip ne mažiau svarbų aspektą respondentai išskyrė komunikaciją ir kliento edukavimą apie taikomus darbo procesus. Kaip teigia ekspertai, klientai dažnai būna nepratę prie iteratyvaus darbo modelio, todėl reikia siųsti aiškią žinutę

apie tokių darbo principų naudą ir nuolat skaidriai komunikuoti projekto progresą, tiek geruosius aspektus, tiek nesėkmes, kadangi tik esant pasitikėjimų grįstam santykiui galima orientuotis į efektyvų darbą vietoje sutarties eilučių tikrinimo.

Vienas esminių reikalavimų inžinerijos etapų yra reikalavimų surinkimas, kuris empirinio tyrimo metu buvo analizuotas remiantis ekspertų praktine patirtimi. Tyrimo metu nustatyti pagrindiniai reikalavimų surinkimo metodai – interviu, dirbtuvės, smegenų šturmas ir prototipavimas – atitinka literatūroje (Hegade, 2020; Mesquita et al., 2023; Mich, 2021; Ribeiro et al., 2014; Khade et al., 2022; Gardner et al., 2020; Brandolini, 2021) dažniausiai išskiriamus metodus. Tačiau tyrimo metu ekspertai nepaminėjo tokių literatūroje (Mesquita et al., 2023; Ribeiro et al., 2014) minimų reikalavimų surinkimo metodų kaip dokumentų analizė ar šešių skrybėlių metodas. Beje, empirinio tyrimo metu buvo nustatytas kitas metodas – naudotojų stebėjimas. Šis metodas leidžia įvertinti, kiek patogus ir intuityvus yra sukurtas sistemos funkcionalumas. Pagal stebėjimo rezultatus atliekamos korekcijos, tokiu būdu patobulinant sistemą. Respondentai išskyrė kūrybinių ir tradicinių reikalavimų surinkimo metodų naudą bei trūkumus ir nurodė, jog renkant reikalavimus metodai turi būti derinami tarpusavyje – vieno efektyviausio metodo nėra. Remiantis ekspertų patirtimi – projekto pradžioje reikalavimai surenkami aukštu lygiu bendram vaizdui susidaryti, o vėliau žinios gilinamos konsultuojantis su konkrečios srities ekspertais kliento pusėje. Ekspertai nurodė ypatingą vizijos ir krypties svarbą sėkmingam projektui. Bendrą vaizdą suratyti padeda dirbtuvių metu taikomi įvykių modeliavimo (angl. *Event Storming*), naudotojų patirties žemėlapis (angl. *User Story Mapping*) ir konteksto žemėlapis (angl. *Context Mapping*) metodai, kurių produktas – aukšto lygio sistemos žemėlapis. Siekdami surinkti išsamius reikalavimus ekspertai derina interviu ir prototipavimo metodus – atlikus interviu greitai sukuriamas sistemos prototipas, vizualiai atspindintis esminį funkcionalumą. Tokių būdu greitai patikrinami reikalavimai, ženkliai sumažinant sistemos klaidų ir klaidingos interpretacijos tikimybę. Remiantis tiek literatūros analize (Hoy & Xu, 2023; Srivastava et al., 2023; Dong et al., 2024; Rasheed et al., 2021; Yaseen et al., 2022), tiek empirinio tyrimo rezultatais – nuolatinis reikalavimų validavimas ir prioritizavimas yra kertinis aspektas, siekiant sukurti realius verslo poreikius ir kliento lūkesčius atitinkančią sistemą.

4.3. Reikalavimų inžinerijos procesų tobulinimo IT projektuose rekomendacijos

Reikalavimų inžinerijos procesų tobulinimo rekomendacijos pateikiamos 14 lentelėje, išskiriant tyrimo metu identifikuotas pagrindines reikalavimų inžinerijos IT projektuose problemas, identifikuotus sprendimo būdus ir laukiamą rezultatą, pritaikius rekomendaciją.

14 lentelė. Reikalavimų inžinerijos procesų tobulinimo IT projektuose rekomendacijos

Procesas	Identifikuotas iššūkis	Rekomendacija	Laukiamas rezultatas
Reikalavimų surinkimas, analizė ir validavimas	Netikslus verslo poreikių nustatymas: neteisingos prielaidos, neišgryninti reikalavimai, brangios analizės klaidos.	Efektyviam verslo poreikių nustatymui būtina aiškiai apibrėžti sprendžiamą problemą/iššūkį bei nustatyti kriterijus, leidžiančius objektyviai įvertinti, kada problema bus išspręsta. Ankstyvoje projekto stadijoje būtina išgryninti ir apibrėžti projekto apimtį, vėliau griežtai kontroliuojant jos kitimą. Prioritetas turi būti kuriamos vertės suvokimas – reikalavimų analizė turi būti orientuota ne į specifikacijų surinkimą, bet į kliento problemos esmės identifikavimą.	Tiksliai apibrėžta sprendžiama problema sukuria bendrą supratimą tarp suinteresuotųjų šalių, aiškiai identifikuoja kuriamą vertę ir užtikrina sklandų bendradarbiavimą nuo pat projekto pradžios.

		Siekiant nustatyti pagrindines sprendžiamas problemas, rekomenduojama taikyti „5 kodėl“ metodą. Svarbu kaip įmanoma ankstyvesnėse stadijose atlikti reikalavimų validavimą, naudojant prototipus, vizualizavimo priemones. Rekomenduojama taikyti mišrią reikalavimų surinkimo metodiką – dirbtuvės padeda suformuoti bendrą projekto viziją ir išgryninti aukšto lygmens reikalavimus, o interviu su specifiniais naudotojais leidžia detaliai analizuoti siauresnes sritis bei identifikuoti tikslus reikalavimus kiekvienam sistemos segmentui.	
Reikalavimų dokumentacija	Prastas reikalavimų aprašymas	Siekiant tinkamai ir suprantamai aprašyti reikalavimus, rekomenduojama nusistatyti arba adaptuoti standartizuotus reikalavimų dokumentavimo principus. Ypatingai svarbu apibrėžti detalumo lygį – nurodyti, kokia informacija turi būti aprašyme, o kokie sprendimai gali būti paliekami programuotojų sprendimui. Siekiant apsaugoti nuo potencialių problemų, rekomenduojama į reikalavimų gryninimą įtraukti visą projekto komandą. Patariama naudoti kontrolinius sąrašus su pagrindiniais reikalavimų aprašymo aspektais.	Užtikrintas aiškus, nuoseklus ir pakankamai detalus reikalavimų aprašymas, kuris leidžia sumažinti netinkamo interpretavimo ir iš to kylančių klaidų tikimybę. Standartizuotai aprašomi reikalavimai palengvina darbą projekto komandai, o projekto perdavimo atveju – sumažina adaptacinį periodą.
Reikalavimų dokumentacija	Dokumentavimo trūkumas	Rekomenduojama sukurti nuoseklią ir lengvai prieinamą dokumentacijos sistemą. Joje turi atsispindėti esminiai architektūriniai sprendimai, sudėtinga logika. Dokumentacija turi būti palaikoma viso projekto metu, kad informacija išliktų aktuali ir naudinga tiek dabartinei, tiek būsimoms projekto komandoms. Rekomenduojama, kad dokumentavimo veiklos neužimtų daugiau nei 20 % analizės veiklų laiko.	Struktūruota ir naudinga dokumentacija, kuri leidžia greitai suprasti, dėl ko buvo priimti tam tikri sprendimai arba dėl ko sistemos veikimas yra būtent toks. Esant tinkamai dokumentacijai, projekto perdavimas ir palaikymas tampa sklandesnis.
Reikalavimų surinkimas	Vizijos trūkumas	Rekomenduojama ankstyvoje projekto stadijoje suformuoti bendrą visų suinteresuotų šalių supratimą apie kuriamos sistemos paskirtį, tikslus ir laukiamą pridėtinę vertę. Pasiekti bendrą suvokimą bei sukurti sistemos žemėlapi leidžia įvykių modeliavimo (angl. <i>Event Storming</i>), naudotojų patirties žemėlapi (angl. <i>User Story Mapping</i>) ir konteksto žemėlapi (angl. <i>Context Mapping</i>) metodai, pritaikomi dirbtuvių metu.	Sukurtas bendras supratimas tarp visų suinteresuotųjų šalių, kokią problemą siekiama išspręsti. Tokiu būdu išvengiama nukrypimų nuo esminių tikslų, sumažinama rizika spręsti netinkamas problemas, o kuriamas produktas atitinka kliento lūkesčius.

Reikalavimų surinkimas, validavimas	Netinkama komunikacija	Siekiant užtikrinti sklandų ir greitą projekto vykdymą, rekomenduojama nuo projekto pradžios megzti pasitikėjimu grįstą santykį su klientais. Rekomenduojama dalykiniams santykiams su klientu skirti žmones, kurie yra empatiški ir turi gerai išugdytus komunikavimo, įsiklausymo įgūdžius.	Sukurtas draugiškas, pasitikėjimu grįstas ryšys su klientais leidžia visą dėmesį sutelkti į produkto vystymą ir projekto įgyvendinimą, atsiribojant nuo griežto sutarties sąlygų sekimo.
Reikalavimų valdymas	Prastas pokyčio valdymas	Rekomenduojama įsivesti aiškų procesą, kuris užtikrintų grįžtamojo ryšio ir pakeitimų užklausų įtraukimą į projekto eigą, tuo pačiu užtikrinant, jog tokie pakeitimai neišplėstų projekto apimtį. Priimančiam tikrus pakeitimų prašymus ypatingai svarbu užtikrinti atvirą ir dažną komunikaciją apie pakeitimų įtaką projekto terminams ir kuriamam rezultatui. Atsiradus naujai informacijai rekomenduojama iš naujo sudėlioti prioritetus. Tokia veikla leidžia užtikrinti maksimaliai vertingą projekto rezultatą ir išlaikyti gerus santykius.	Sukurtas pokyčių valdymo procesas, kuris užtikrina projekto apimtį pastovumą, atitinkamai surikiuojant prioritetus, kurie neša didžiausią vertę.
Reikalavimų surinkimas, analizė	Programinės įrangos kūrimas – tiriamasis darbas	Rekomenduojama nuo pat projekto pradžios edukuoti klientą apie taikomus iteratyvius darbo metodus, pabrėžti, jog kuriamas visiškai naujas produktas, kuriam reikalavimai iki šiol nebuvo žinomi ir kūrimo metu gali keistis. Siekiant sukurti maksimalią naudą užsakovui, rekomenduojama į produkto vystymą kiek įmanoma daugiau įtraukti klientą.	Pasiektas bendras supratimas tarp kliento ir sistemos vystytojo, jog produkto kūrimas yra iteratyvus procesas, kuriame gali kilti nenumatytų iššūkių ir rizikų. Tikimasi, jog į vystymą glaudžiai įtraukiant klientą bus sumažinta rizika sukurti poreikių neatitinkantį produktą.
Reikalavimų surinkimas, analizė, validavimas, dokumentacija	Ribota prieiga prie įrankių	Rekomenduojama projekto pradžioje atidžiai pasirinkti įrankius, reikalingus reikalavimų analizei, vizualizavimui ir bendram darbui. Verta atsižvelgti į tai, jog esant poreikiui būtų galimybė įsigyti naudojamų įrankių licencijas, užtikrinti, kad klientas taip pat galėtų naudotis tais pačiais įrankiais, siekiant bendradarbiauti ir dalintis rezultatu.	Sumažėjusios kliūtys dėl poreikio naudoti skirtingus, tam tikrais atvejais nepatogius darbo įrankius, leidžia fokusuotis į vertės kūrimą, sumažinant laiko švaistymą, atsirandantį dėl nepatogių įrankių.

Reikalavimų inžinerijos procesų tobulinimo IT projektuose tyrimo rezultatai gali būti naudingi projektiniu principu darbą vykdančioms IT kompanijoms, siekiančioms patobulinti reikalavimų inžinerijos procesus, pagerinti santykį su klientu ir padidinti sėkmingų projektų skaičių. Ateities tyrimuose rekomenduojama ištirti, kaip pasikeičia reikalavimų inžinerijos procesų efektyvumas, projekto sėkmės rodikliai, pritaikius šiame tyrime pateiktas rekomendacijas.

Išvados

1. Atlikus statistikos ir mokslinių tyrimų analizę nustatyta, jog investicijos į IT sritį tendencingai didėja, tačiau sėkmingai užbaigiami vos 29% IT projektų. Remiantis mokslinių tyrimų rezultatais, galima teigti, jog reikalavimų inžinerija daro lemiamą įtaką IT projektų (ne)sėkmingam užbaigimui. Netikslus verslo poreikių nustatymas įvardijamas kaip svarbiausia priežastis, sukelianti IT projektų nesėkmes. Vis dėlto mokslinėje literatūroje daugumoje nagrinėtų atvejų analizuojami reikalavimų inžinerijos iššūkiai ir neidentifikuojamos gerosios praktikos procesų tobulinimui. Reikalavimų inžinerijos procesų tobulinimo tyrimo rezultatai užpildytų šią mokslinėje literatūroje randamą spragą ir rekomendacijas pritaikiusioms įmonėms suteiktų konkurencinį pranašumą, pagerintų prisitaikymą prie rinkos poreikių.
2. Išanalizavus reikalavimų inžinerijos procesų tobulinimo teorinius sprendimus, taikomus IT projektuose, galima teigti, jog reikalavimų inžinerijos procesai yra pripažįstami kaip svarbiausia ir sudėtingiausia programinės įrangos kūrimo ciklo dalis. Atlikus mokslinės literatūros analizę, buvo nustatyta, jog:
 - IT projektų valdyme dažniausiai taikomos dvi pagrindinės projektų valdymo metodologijos – krioklio (angl. *Waterfall*) ir „Agile“. „Agile“ projektų valdymo metodologija IT srityje yra gerokai labiau paplitusi dėl jos privalumų: lengvesnis prisitaikymas prie pokyčių, klientų įtraukimas, nuolatinis tarpinio rezultato validavimas, lankstumas vertinant pasikeitusią situaciją;
 - reikalavimų inžinerijos procesus sudaro: reikalavimų surinkimas, reikalavimų analizė, reikalavimų validavimas, reikalavimų dokumentacija ir reikalavimų valdymas. Šie procesai nėra linijiniai, galimas sugrįžimas iš vieno proceso į kitą;
 - siekiant tobulinti procesus, naudojamos metrikos pamatuoti jų efektyvumui ir našumui. Reikalavimų inžinerijoje vyrauja kokybiniai pagrindiniai veiklos rodikliai, kadangi ši sritis yra susijusi su neapibrėžtumu ir tiksliai skaitine verte išreiškiamų rodiklių yra nedaug: klaidų skaičius, proceso trukmė; taip pat identifikuoti kokybiniai rodikliai: reikalavimų aiškumas, suinteresuotųjų šalių įsitraukimas, pakankamas dokumentacijos kiekis;
 - kaip galima reikalavimų inžinerijos procesų tobulinimo priemonė literatūroje išskiriamas vizualizavimas, padedantis vaizdžiai pavaizduoti procesus ir pamatyti silpnąsias vietas. Vizualizavimui dažnai naudojami įvairūs diagramų metodai dėl jų paprastumo ir intuityvumo. Šie metodai yra suprantami net mažai techninių žinių turintiems atstovams, todėl labiau tinkami skirtingų sričių ekspertų bendradarbiavimui. Procesų vizualizacijai dažnai naudojama standartizuota notacija – BPMN;
 - procesų tobulinimo įrankiai Ishikawa diagrama ir „5 kodėl“ metodas leidžia atrasti šaknines problemos priežastis ir jas pašalinus panaikinti iš to kylančius nepageidaujamus simptomus. Literatūroje aprašomas SCAMPER metodas yra kūrybiško mąstymo metodas, skirtas idėjų generavimui, leidžiantis kūrybiškai pažiūrėti į problemą, remiantis tam tikra klausimų sistema. Jis leidžia atsitraukti nuo įprastos logikos ir sistemingai kurti kūrybiškas ir įvairias idėjas pagal iš anksto parengtą kontrolinį sąrašą. Jis gali pasitarnauti siekiant atrasti patobulinimus silpnosioms procesų vietoms, taip pat gryninant reikalavimus, kuomet ieškoma geresnės esamo sprendimo alternatyvos;
 - Išanalizavus reikalavimų inžinerijos procesus, reikalavimų surinkimas buvo identifikuotas kaip pats sudėtingiausias procesas, kuriame padaroma daugiausiai klaidų. Pagrindiniai reikalavimų surinkimo iššūkiai siejami su neaiškiai suformuluotais, dviprasmiškais reikalavimais, skirtingais suinteresuotųjų šalių lūkesčiais bei komunikacijos trūkumu tarp klientų ir projekto komandos. Reikalavimų surinkimo metodai išskiriami į tradicinius (interviu, dokumentų analizė) ir

kūrybinius (dirbtuvės, kurių metu naudojami įvykių modeliavimo (angl. *Event Storming*), naudotojų patirties žemėlapis (angl. *User Story Mapping*), šešių skrybėlių (angl. *Six Thinking Hats*) metodai). Kūrybiniai reikalavimų surinkimo metodai praplečiančia mąstymo ribas ir skatinama į problemą pažvelgti iš skirtingų pusių. Jie leidžia išnaudoti žmogaus kūrybiškumą, sprendžiant sudėtingas, kelių lygių problemas. Pastebėta, jog mokslinėje literatūroje siūloma derinti kūrybinius ir tradicinius reikalavimų rinkimo metodus, tokius kaip interviu ir dirbtuvės.

3. Atliekant empirinį tyrimą pasirinktas pusiau struktūruoto interviu metodas. Šis metodas pasirinktas dėl galimybės giliau pažvelgti į reikalavimų inžinerijos ekspertų patirtis ir sužinoti, kokie metodai skirtingose situacijose pasiteisina praktinėje veikloje. Tyrimo dalyvių atranka – patogioji tikslinė, kurioje tinkami tyrimui ekspertai pasirenkami iš žinomos aplinkos, o sekantys tyrimo dalyviai randami sniego gniūžtės metodu, kai tyrime dalyvavę ekspertai rekomenduoja kitus asmenis, turinčius tyrimui tinkamos patirties. Interviu klausimynas buvo parengtas remiantis mokslinės literatūros analizės rezultatais ir sudarytu conceptualiuoju modeliu. Tyrimo metu klausimynas buvo papildytas pagal atrastą naują informaciją. Tyrime dalyvavo trijų skirtingų įmonių atstovai, dirbantys IT projektuose. Interviu buvo tęsiami tol, kol buvo randama naujos informacijos. Kai didžioji dalis informacijos pradėjo kartotis, buvo pasiekta tyrimo saturacija ir interviu vykdymas baigtas.
4. Atlikus mokslinės literatūros analizės ir empirinio tyrimo rezultatų palyginamąją analizę, identifiкуotos šios pagrindinės reikalavimų inžinerijos iššūkių grupės: kliento pasirengimo ir įsitraukimo trūkumas, pernelyg abstraktūs projekto tikslai, nesuteikiantys aišrios krypties, netikslus verslo poreikių nustatymas – analizės trūkumas ir neteisingos prielaidos, netinkamas pokyčių valdymas. Remiantis tyrimo rezultatais, pateikiamos sekančios rekomendacijos reikalavimų inžinerijos procesų tobulinimui:
 - projekto pradžioje turi būti aiškiai ir vienareikšmiškai apibrėžiamas projekto tikslas ir bendra vizija;
 - identifiкуojant pagrindinę sprendžiamą problemą rekomenduojama taikyti „5 kodėl“ analizės metodą;
 - rekomenduojama taikyti suderintus reikalavimų surinkimo metodus, kurie leidžia pirmuoju etapu sukurti bendrą projekto vaizdą ir viziją, vėliau, įsigilinus į detales, sukurti detalius reikalavimų aprašus;
 - aukšto lygio reikalavimams surinkti rekomenduojamas bendras suinteresuotųjų šalių susitikimas, kuriame dirbtuvių principu būtų taikomi įvykių modeliavimo (angl. *Event Storming*), naudotojų patirties žemėlapis (angl. *User Story Mapping*) ar konteksto žemėlapis (angl. *Context Mapping*) metodai. Sekančiuose etapuose rekomenduojama naudoti interviu, prototipavimo metodus tikslų reikalavimų išgryninimui;
 - rekomenduojama reguliariai validuoti tarpinį rezultatą, naudojant prototipus ar kitas vizualizavimo priemones;
 - rekomenduojama dažnai ir reguliariai įtraukti klientų grįžtamąjį ryšį į projekto vystymą, siekiant sukurti realius poreikius atitinkantį ir problemas sprendžiantį produktą. Vis dėlto grįžtamojo ryšio ir kitų pakeitimų įtraukimas į projekto apimtį turėtų būti atidžiai kontroliuojamas, iš naujo dėliojami prioritetai siekiant išvengti didelių apimties pasikeitimų ir nenukrypti nuo nustatytų biudžeto ir laiko apribojimų;
 - rekomenduojama nuolat edukuoti klientą ir aiškiai komunikuoti taikomus darbo metodus, pabrėžiant, jog sistemos kūrimas yra tiriamasis darbas ir dėl to tam tikros prielaidos gali nepasitvirtinti;
 - rekomenduojama sukurti arba adaptuoti ir palaikyti svarbiausių sprendimų ir sistemos logikos dokumentaciją, jai skiriant iki 20% viso reikalavimų inžinerijos procesų laiko.

Literatūros sąrašas

1. Alazzawi, A., Yas, Q. M., & Rahmatullah, B. (2023). A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions. *Iraqi Journal for Computer Science and Mathematics*, 4(4), 173–190. doi:10.52866/ijcsm.2023.04.04.014
2. Anifa, M., Ramakrishnan, S., Kabiraj, S., & Joghee, S. (2024). Systematic Review of Literature on Agile Approach. *NMIMS Management Review*, 32(2), 84-105. <https://doi.org/10.1177/09711023241272294>
3. Brandolini, A. (2021). *Introducing EventStorming*. Leanpub.
4. Alrabadi, T. D. S., Talib, Z. M., & Abdullah, N. ' . B. (2023). The role of quality 4.0 in supporting digital transformation: Evidence from telecommunication industry. *International Journal of Data and Network Science (Print)*, 7(2), 717–728. doi:10.5267/j.ijdns.2023.2.006
5. Barsalou, M., Starzyńska, B., & Konrad, M. (2024). Five whys: A possible path to failure. *International Journal of Six Sigma and Competitive Advantage*, 15(1), 19–32. doi:10.1504/IJSSCA.2024.139599
6. Behutiye, W., Karhapää, P., López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Oivo, M. (2020). Management of quality requirements in agile and rapid software development: A systematic mapping study. *Information and Software Technology*, 123, 106225. doi:10.1016/j.infsof.2019.106225
7. Benjamin, S. J., Marathamuthu, M. S., & Murugaiah, U. (2015). The use of 5-WHYs technique to eliminate OEE's speed loss in a manufacturing firm. *Journal of Quality in Maintenance Engineering*, 21(4), 419–435. doi:10.1108/JQME-09-2013-0062
8. Bilsel, R. U., & Lin, D. K. J. (2012). Ishikawa cause and effect diagrams using capture recapture techniques. *Quality Technology & Quantitative Management*, 9(2), 137–152. doi:10.1080/16843703.2012.11673282
9. Boonpracha, J., Roong-In, J., Lookraks, S., Wongtanasuporn, P., Kooptiwoot, S., & Seangkong, S. (2023). Creativity of students' cultural product design using the SCAMPER technique. *Journal of Mekong Societies*, 19(2), 179–196.
10. Borhan, N. H., Zulzalil, H., Hassan, S., Ali, N. M., Sultan, A. B. M., & Bahsoon, R. (2025). Stakeholder analysis using fuzzy logic operations for integrated user story prioritisation approach in agile-scrum method. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 47(2), 76–93. doi:10.37934/ARASET.47.2.7693
11. Botezatu, C., Condrea, I., Oroian, B., Hrițuc, A., Ețcu, M., & Slătineanu, L. (2019). Use of the ishikawa diagram in the investigation of some industrial processes. *IOP Conference Series. Materials Science and Engineering*, 682(1), 12012. doi:10.1088/1757-899X/682/1/012012
12. Calabro, A., Lonetti, F., & Marchetti, E. (2015). Monitoring of business process execution based on performance indicators. *2015 41st Euromicro Conference on Software Engineering and Advanced Applications, Madeira, Portugal, 2015* (pp. 255-258). doi: 10.1109/SEAA.2015.73.
13. Casebolt, J. M., Jbara, A., & Dori, D. (2020). Business process improvement using Object-Process Methodology. *Systems Engineering*, 23(1), 36–48. <https://doi.org/10.1002/sys.21499>
14. Chari, K., & Agrawal, M. (2018). Impact of incorrect and new requirements on waterfall software project outcomes. *Empirical Software Engineering*, 23(1), 165–185. doi:10.1007/s10664-017-9506-4
15. Chirumamilla, A., Nguyen Duc, A., Gao, S., & Sindre, G. (2018). A systematic mapping study on requirements engineering in software ecosystems. *Journal of Information Technology Research*, 11, 49–69. doi:10.4018/JITR.2018010104
16. Copei, S., Emme, C., Von Künßberg, M. F., Malik, A., Nolte, N., Norbistrath, U., & Zündorf, A.

- (2022). Supporting requirements engineering and development with event modeling - an overview. *Modellierung 2022*, (pp. 145–154). doi:10.18420/modellierung2022-007
17. Crowder, J. A., & Hoff, C. W. (2022). *Requirements engineering: Laying a firm foundation*. Switzerland: Springer Nature Switzerland AG.
 18. do Nascimento, F. R., dos Santos, A. G., da Fonseca Junior, L. A., & Nunes, D. M. (2024). Application of lean six sigma to reduce delays in engineering changes. *IEEE Transactions on Engineering Management*, 71, 2786–2799. doi:10.1109/TEM.2022.3199388
 19. Dong, H., Dacre, N., Baxter, D., & Ceylan, S. (2024). What is agile project management? developing a new definition following a systematic literature review. *Project Management Journal*, 55(6), 668–688.
 20. Edwards, R., & Holland, J. (2013). *What is Qualitative Interviewing?* (1st ed.). Bloomsbury Publishing Plc. <https://doi.org/10.5040/9781472545244>
 21. Ferrari, A., Spoletini, P. & Debnath, S. (2022). How do requirements evolve during elicitation? An empirical study combining interviews and app store analysis. *Requirements Engineering* 27(2), 489–519. <https://doi.org/10.1007/s00766-022-00383-7>
 22. Gadatsch, A. (2023). *Business process management: Analysis, modelling, optimisation and controlling of processes* (1st ed.). Wiesbaden: Springer Fachmedien Wiesbaden. doi:10.1007/978-3-658-41584-6
 23. Gangidi, P. (2019). A systematic approach to root cause analysis using 3 × 5 why's technique. *International Journal of Lean Six Sigma*, 10(1), 295–310. doi:10.1108/IJLSS-10-2017-0114
 24. Gardner, H., Blackwell, A. F., & Church, L. (2020). The patterns of user experience for sticky-note diagrams in software requirements workshops. *Journal of Computer Languages*, 61, 100997. doi:10.1016/j.cola.2020.100997
 25. Grimm, V., & Rubart, J. (2021). Modelling gamified systems with event storming augmented by spatial hypertext. *HUMAN '21: Proceedings of the 4th Workshop on Human Factors in Hypertext* (pp. 3–10). doi:10.1145/3468143.3483927
 26. Hegade, P. (2020). Design for requirements engineering. *Journal of Engineering Education Transformations* 33, 219–224. doi: 10.16920/jeet/2020/v33i0/150149
 27. Hilmer, C. (2016). *Prozessmanagement und indirekte Bereiche*. In: *Prozessmanagement in indirekten Bereichen. Unternehmensführung & Controlling*. Springer Gabler, Wiesbaden. doi: 10.1007/978-3-658-14917-8_2
 28. Hollender, M., Chioua, M., Schlake, J., Merkert, L., & Petersen, H. (2016). KPI-based process operation: Management of highly automated processes. *GWF, Wasser - Abwasser*, 157(9), 52–61
 29. Hoy, Z., & Xu, M. (2023). Agile software requirements engineering challenges-Solutions—A conceptual framework from systematic literature review. *Information (Switzerland)*, 14(6) doi:10.3390/info14060322
 30. Imanbekova, A. N. (2023). Development of an information system of a medical institution. *Mechanics and Technologies* 4(82), 140–148. doi: 10.55956/ZCGT6039
 31. Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51, 915–929. doi:10.1016/j.chb.2014.10.046
 32. Jarzębowicz, A., & Poniatowska, K. (2020). Exploring the impact of requirements engineering on other IT project areas - A case study. *Computer Science* 21(3), 261–286. doi: 10.7494/csci.2020.21.3.3618
 33. Jon Tømmerås, S., Ian, S., & Eirik Bjorheim, A. (2020). SMART criteria for quality assessment of key performance indicators used in the oil and gas industry. *International Journal of Performability*

- Engineering*, 16(7), 999. doi:10.23940/ijpe.20.07.p2.9991007
34. Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *The Journal of Systems and Software*, 172, 110851. doi:10.1016/j.jss.2020.110851
 35. Kataoka, T., Furuto, K., & Matsumoto, T. (2011). The analyzing method of root causes for software problems. *SEI Technical Review*, (73), 81–85.
 36. Keyser, R. S., & Pooyan, P. (2023). Lean at home: Applying 5 whys and lean PFMEA to home projects. *Journal of Applied Research on Industrial Engineering*, 10(1), 45–55. doi:10.22105/jarie.2022.320848.1410
 37. Khade, V., Masoudi, N., Acena, D., Freeman, G., Rai, R., Gorsich, D., Castanier, M. (2023) Requirements elicitation: Impacts of gamification on variety, novelty, and completeness. ASME 2022 International Mechanical Engineering Congress and Exposition. doi:10.1115/IMECE2022-96016
 38. Korecki, Z., & Adámková, B. (2020). Multi-criteria evaluation of critical infrastructure resilience and economic impacts in selected airports in the czech republic. *Zeszyty Naukowe Politechniki Śląskiej*, 108(108), 95–107. doi:10.20858/sjsutst.2020.108.9
 39. Levkivska, L., & Levkivskyi, S. (2024). Use of information technologies in higher school. *Scientific Notes of Junior Academy of Sciences of Ukraine*, 1(29), 57–66. doi: 10.51707/2618-0529-2024-29-07
 40. Mendez, D., & Moreira, A. (2024). *Requirements engineering: Foundation for software quality. 30th international working conference, REFSQ 2024, Winterthur, Switzerland, April 8-11, 2024, proceedings*. Springer.
 41. Mesquita, R., Silva, G., & Canedo, E. (2023) On the experiences of practitioners with requirements elicitation techniques. *SBES '23: Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (pp. 442-451). doi:10.1145/3613372.3613410
 42. Mich, L. (2021). Choosing a creativity technique for requirements elicitation. *CREARE: 10th International Workshop on Creativity in Requirements Engineering*.
 43. Mohylina, L., & Makarova, V. (2023). Implementation of information technologies in domestic enterprises. *Priazovs'kij Ekonomičnij Visnik*, 1(33). doi:10.32840/2522-4263/2023-1-11
 44. Money, W., Kaisler, S., & Cohen, S. (2021). Understanding failed software projects through forensic analysis. *Journal of Computer Information Systems*, 62, 1–14. doi:10.1080/08874417.2021.1950076
 45. Nizam, A. (2022). Software project failure process definition. *IEEE Access*, 10, 34428–34441. doi:10.1109/ACCESS.2022.3162878
 46. Ochodek, M., Koronowski, K., Matysiak, A., Miklosik, P., & Kopczyńska, S. (2016). Sketching use-case scenarios based on use-case goals and patterns. *Software engineering: Challenges and solutions* (pp. 17–30). Switzerland: Springer International Publishing AG. doi:10.1007/978-3-319-43606-7_2
 47. Ogirri, K. O., & Idugie, I. J. (2024). A comparative analysis of traditional versus agile project management methodologies on IT project outcomes. *Asian Journal of Research in Computer Science* 17(9), 1-12. doi: 10.9734/ajrcos/2024/v17i9495
 48. Panduro, A. F. (2023). Technologies applied to information control in organizations: A review. *DecisionTech Review*, 3, 1–6. doi:10.47909/dtr.02
 49. Parkhatova, R., & Erzhanova, Z. (2022). Importance of IT technologies nowadays. *Humanitarian and socio-economic sciences journal*, 6(17), 61-64.
 50. Pergl, R., Babkin, E., Lock, R., Malyzhenkov, P., Merunka, V., Malyzhenkov, P., Babkin, E. (2019). *Enterprise and organizational modeling and simulation: 15th international workshop, EOMAS*

- 2019, held at CAiSE 2019, Rome, Italy, June 3-4, 2019, selected papers (1st ed.). Netherlands: Springer Nature.
51. Phokwane, I., & Arnesh, T. (2021). Optimizing and modelling business processes for a successful implementation of process automation. *30th International Conference of the International Association for Management of Technology: MOT for the World of the Future, IAMOT 2021* (pp. 389–398). doi:10.52202/060557-0027
 52. Pidun, T., & Felden, C. (2011). Limitations of performance measurement systems based on key performance indicators. *A Renaissance of Information Technology for Sustainability and Global Competitiveness. 17th Americas Conference on Information Systems, AMCIS 2011, Detroit, Michigan, USA, August 4-8 2011*.
 53. Pratama, A., Sensuse, D., Noprisson, H. (2017). A systematic literature review of business process improvement. *2017 International Conference on Information Technology Systems and Innovation (ICITSI)* (pp. 26-31). doi: 10.1109/ICITSI.2017.8267913.
 54. Rasheed, A., Zafar, B., Shehryar, T., Aslam, N., Sajid, M., Ali, N., Khalid, S. (2021). Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering, 2021*, 1–18. doi: <https://doi.org/10.1155/2021/6696695>
 55. Řepa, V. (2019). Deriving key performance indicators from business process model. *Perspectives in business informatics research* (pp. 148–162). Switzerland: Springer International Publishing AG. doi:10.1007/978-3-030-31143-8_11
 56. Ribeiro, C., Farinha, C., Pereira, J., & Mira da Silva, M. (2014). Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholders collaboration. *Entertainment Computing, 5*(4), 335–345. doi:10.1016/j.entcom.2014.04.002
 57. Rojas, C., Fernando, L., Jaramillo, Z., & Mario, C. (2013). Executable pre-conceptual schemas for representing key performance indicators. *2013 8th Computing Colombian Conference (8CCC)* (pp. 1–6). doi:10.1109/ColombianCC.2013.6637526
 58. Sardjono, W., Selviyanti, E., Perdana, W. G., & Sudrajat, J. (2020). Modeling of development of employee performance factors and dissemination. *Journal of Physics. Conference Series, 1465*(1), 12024. doi:10.1088/1742-6596/1465/1/012024
 59. Shen, G. Q. P., Al-Hussein, M., Zhu, Y., & Wang, Y. (2017). Looking into not-completed activities by applying the 5 whys method. *ICCREM 2016 - BIM application and offsite construction - proceedings of the 2016 international conference on construction and real estate management, September 29-October 1, 2016, Edmonton, Alberta, Canada* (pp. 1). doi: 10.1061/9780784480274.040
 60. Siwiec, D., Gawlik, R., & Pacana, A. (2023). Fuzzy multi-criteria decision model to support product quality improvement. *Management and Production Engineering Review, 14*(2), 134. doi:10.24425/mper.2023.146030
 61. Srivastava, P., Srivastava, N., Agarwal, R., Singh, P. (2023). Incorporating Requirement Engineering into Agile Methodologies: Challenges and Proposed Solutions. *2023 9th International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2023* (pp. 351-355). doi: 10.1109/ICSC60394.2023.10441607.
 62. Vululleh, P. (2021). *Cyber Careers: The Basics of Information Technology and Deciding on a Career Path*. CRC Press.

63. Wetzstein, B. (2016). KPI-related monitoring, analysis, and adaptation of business processes. doi:10.18419/opus-8956
64. Wolniak, R. (2019). Downtime in the automotive industry production process – cause analysis. *Kvalita Inovacia Prosperita*, 23(2), 101–118. doi:10.12776/qip.v23i2.1259
65. Wolniak, R., & Grebski, W. (2023). The usage of 5 why in industry 4.0 conditions. *Scientific Papers of Silesian University of Technology. Organization and Management Series*, 2023(189), 735–745. doi:10.29119/1641-3466.2023.189.46
66. Yaseen, M., Ali, S., Mustapha, A., & Mazhar, N. (2022). Success factors analysis for requirement elicitation in global software development paradigm: An empirical study. *Journal of Software: Evolution and Process*, 34(7). doi:10.1002/smr.2460
67. Yugay, A., Pimenta, G., Zhukin, A., Daghmouni, H. B., Silchenok, M., Bongiovanni, C., Alam, M. S. (2021). Root cause analysis of wells with integrity failures. *ADIPEC 2021* doi:10.2118/207382-MS

Informacijos šaltinių sąrašas

1. Alberto Brandolini. Alberto brandolini [žiūrėta 2024-12-07]. Prieiga per internetą: <https://www.avanscoperta.it/en/trainer/a-brandolini/>
2. Codacity Platform. (2024). SPACE Framework: How to Measure Developer Productivity [žiūrėta 2025-04-15]. Prieiga per internetą: <https://blog.codacy.com/space-framework>
3. Colleen, M. (2024). 17th state of agile report: 71% use agile in their SDLC; small organizations report strong business benefits; medium and larger-sized companies continue to experience barriers in successfully scaling agile [žiūrėta 2024-12-15]. Prieiga per internetą: <https://www.businesswire.com/news/home/20240116199385/en/>
4. Content Square. (2024). What is Net Promoter Score (NPS)? How to Calculate & Interpret [žiūrėta 2025-04-15]. Prieiga per internetą: <https://contentsquare.com/guides/net-promoter-score/>
5. Elmansy, R. (2015). A guide to the SCAMPER technique for design thinking [žiūrėta 2024-12-30]. Prieiga per internetą: <https://www.designorate.com/a-guide-to-the-scamper-technique-for-creative-thinking/>
6. Hannah, P. (2022). Project management statistics: 45 stats you can't ignore [žiūrėta 2024-12-15]. Prieiga per internetą: <https://www.workamajig.com/blog/project-management-statistics>
7. Kusbeci, P. (2023). Use of information technologies in businesses [žiūrėta 2024-12-15]. *Uluslararası Sosyal Bilimler Dergisi*, 7(28), 47–58. Prieiga per internetą: <https://www.ceeol.com/search/article-detail?id=1118639>
8. Lawless, J. (2024). Fishbone diagram (ishikawa) [žiūrėta 2025-01-15]. Prieiga per internetą: <https://purplegriffon.com/blog/fishbone-diagram-ishikawa>
9. Nicholas, J.9 elicitation techniques used by business analysts – tips and guidance [žiūrėta 2025-01-13]. Prieiga per internetą: <https://businessanalystmentor.com/elicitation-technique/>
10. Palumbo, S., Rehberg, B. & Li, H. (2024). Software projects don't have to be late, costly, and irrelevant [žiūrėta 2024-12-01]. Prieiga per internetą: <https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant>
11. PMWorld 360. (2024, May 24). 5 whys of root cause analysis + examples [žiūrėta 2025-01-15]. Prieiga per internetą: <https://www.peworld360.com/the-power-of-the-5-whys-approach-example-proven-in-root-cause-analysis/>
12. Statista. (2024). Information technology (IT) worldwide spending from 2005 to 2024 [žiūrėta 2024-12-20]. Prieiga per internetą: <https://www.statista.com/statistics/203935/overall-it-spending-worldwide/>
13. Test Bytes. (2024). Requirements elicitation in software engineering: A complete guide [žiūrėta 2025-02-02]. Prieiga per internetą: <https://www.testbytes.net/blog/requirements-elicitation/>
14. Visual Paradigm. (2023a). Demystifying use cases, scenarios, flow of events, and templates [žiūrėta 2025-02-01]. Prieiga per internetą: <https://guides.visual-paradigm.com/demystifying-use-cases-scenarios-flow-of-events-and-templates/>
15. Visual Paradigm. (2023b). Driving successful software development with the use case driven approach: Real-life templates and examples [žiūrėta 2025-02-01]. Prieiga per internetą: <https://guides.visual-paradigm.com/driving-successful-software-development-with-the-use-case-driven-approach-real-life-templates-and-examples/>

Priedai

1 priedas. Pusiau struktūruoto interviu klausimynas

1. Įvadas

- Trumpai apibūdinkite savo vaidmenį reikalavimų inžinerijos procesuose.
- Koks šiuo metu reikalavimų inžinerijos procesas jūsų projekte? Kokie pagrindiniai etapai ir veiklos išskiriami?
- Ar turite tikslą tobulinti reikalavimų inžinerijos procesus? Jei taip, kokias pagrindines kryptis esate numatę?

2. Reikalavimų inžinerijos procesų vertinimas

- Kokius pagrindinius veiklos rodiklius (KPI) taikote reikalavimų inžinerijos procesams?
- Kaip šie veiklos rodikliai padeda įvertinti, ar reikalavimų inžinerijos procesai atitinka keliamus standartus ir reikiamą kokybę?

3. Dažniausiai pasitaikantys iššūkiai

- Su kokiais sunkumais susiduriate renkant reikalavimus iš suinteresuotųjų šalių?
- Kaip dažnai susiduriate su besikeičiančia projekto apimtimi arba reikalavimais? Kokius to valdymo būdus taikote?
- Kaip sprendžiate komunikacijos, misinterpretacijos problemas, pavyzdžiui jeigu reikalavimai nėra aiškiai suformuluoti arba netinkamai interpretuojami?
- Kaip reaguojate į besikeičiančius reikalavimus projekto eigoje? Kokį poveikį turi reikalavimų kaita?

4. Taikomi reikalavimų inžinerijos procesai

- Kokius būdus taikote reikalavimų surinkimui? Kurie iš jų pasiteisina labiausiai, o kurie mažiausiai?
- Kaip užtikriname, kad reikalavimai yra surinkti kokybiškai ir tinkamai dokumentuoti?
- Kaip vyksta reikalavimų validavimas ir derinimas su suinteresuotomis šalimis? Kokius įrankius ar metodus naudojate, kad įsitikintumėte, jog reikalavimai atitinka realius kliento poreikius?
- Kokie sprendimai taikomi reikalavimų valdymui viso projekto metu? Kaip sekate reikalavimų pokyčius?

5. Reikalavimų inžinerijos procesų tobulinimo metodai

- Ar naudojate kokius nors procesų tobulinimo metodus? Kaip tai pasireiškia praktikoje ir kokį poveikį turi reikalavimų inžinerijos procesams?
- Ar naudojate kūrybinius reikalavimų surinkimo metodus? Kaip jie padeda generuoti naujas idėjas arba atrasti tiesiogiai nematomus reikalavimus?
- Ar derinate kūrybinius reikalavimų surinkimo metodus su tradiciniais? Kokias tokio derinimo naudas pastebite?
- Pagal ką pasirenkate, kurį metodą taikyti?
- Kaip manote, kokios kompetencijos yra pačios svarbiausios sėkmingam reikalavimų inžinerijos procesui? Kurios iš jų daugiausiai lemia sėkmę – techninės, komunikacijos, kūrybiškumo?