

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

**„DevOps“ proceso modeliavimas**  
**DevOps Process Modeling**

Magistro baigiamasis darbas

Atliko: Šarūnas Kūjalis (parašas)

Darbo vadovas: dr. Saulius Ragaišis (parašas)

Recenzentas: dr. Stasys Peldžius (parašas)

Vilnius – 2020

## SANTRAUKA

DevOps sąvoka pirmą kartą buvo panaudota 2009 metais ir nors DevOps nuo tada sulaukė daug dėmesio, tačiau vis dar nėra praktiškai taikomo DevOps brandos modelio, dėl ko organizacijoms yra sudėtinga pradėti vykdyti DevOps, o organizacijoms, kurios jau vykdo DevOps, sudėtinga įsivertinti bei efektyviai kelti savo DevOps brandą. Siekiant spręsti šią problemą, šio darbo pagrindinis tikslas yra pasiūlyti naują praktiškai taikomą DevOps brandos modelį.

Darbe buvo priimti sprendimai susiję su naujo DevOps brandos modelio architektūra, brandos lygiais, proceso sritimis, praktikomis bei vertinimu. Remiantis atlikta analize ir priimtais sprendimais buvo sudarytas naujas DevOps brandos modelis.

Taip pat, šiame darbe buvo atliktas naujai pasiūlyto DevOps brandos modelio vertinimas bei nustatyta, koks gebėjimo profilis užtikrinamas pagal CMMI-DEV (kūrimo) ir CMMI-SVC (eksploatavimo) brandos modelius.

Raktiniai žodžiai: DevOps, brandos modelis, proceso modeliavimas, proceso vertinimas.

## SUMMARY

The concept of DevOps was first mentioned in 2009 and although DevOps has received a lot of attention since then, there is still no DevOps maturity model, which could be applied in practice. It makes difficult for organizations to adopt DevOps and for organizations which have already adopted DevOps culture – to self-assess and effectively raise their DevOps maturity. To address this issue, the main goal of this paper is to propose a new DevOps maturity model, which could be applied in practice.

Decisions related to the architecture, maturity levels, process areas, practices and evaluation procedure of the new DevOps maturity model were made in this paper. Based on the analysis performed and the decisions made, a new DevOps maturity model was developed.

Furthermore, in this paper, the evaluation of the newly proposed DevOps maturity model was performed and capability profile provided by the CMMI-DEV (development) and CMMI-SVC (operations) maturity models was determined.

**Keywords:** DevOps, maturity model, process modeling, process assessment.

# TURINYS

ĮVADAS .....	6
1. DEVOPS BRANDOS MODELIAI .....	9
1.1. S. I. Mohamed DevOps brandos modelis .....	9
1.2. I. Bucena ir M. Kirikova DevOps brandos modelis .....	12
1.3. E. Minick DevOps brandos modelis .....	15
1.4. P. Bahrs DevOps brandos modelis .....	18
1.5. Capgemini DevOps brandos modelis .....	19
1.6. H. Beal DevOps brandos modelis .....	21
1.7. Solinea DevOps brandos modelis .....	22
1.8. Rezultatai .....	24
2. DEVOPS PROCESE TAIKOMOS PRAKTIKOS .....	25
2.1. Kultūros proceso srities praktikos .....	25
2.2. Dalinimosi proceso srities praktikos .....	26
2.3. Kokybės proceso srities praktikos .....	27
2.4. Valdymo proceso srities praktikos .....	29
2.5. Diegimo proceso srities praktikos .....	29
2.6. Rezultatai .....	30
3. BAZINIO DEVOPS BRANDOS MODELIO PASIRINKIMAS .....	31
4. NAUJO DEVOPS BRANDOS MODELIO ELEMENTAI .....	40
4.1. Architektūra .....	40
4.2. Brandos lygiai .....	41
4.3. Proceso sritys .....	43
4.4. Praktikos .....	44
4.5. Vertinimas .....	44
5. NAUJAS DEVOPS BRANDOS MODELIS .....	47
5.1. Brandos lygių samprata .....	47
5.2. Proceso sritys .....	48

5.2.1.	Kokybė .....	48
5.2.2.	Valdymas .....	53
5.2.3.	Diegimas .....	57
5.2.4.	Dalinimasis .....	60
5.2.5.	Kultūra .....	62
5.3.	Apibendrinimas .....	63
6.	CMMI GEBĖJIMO PROFILIAI.....	68
6.1.	Gebėjimo profilis pagal CMMI-DEV .....	68
6.2.	Gebėjimo profilis pagal CMMI-SVC.....	71
7.	REZULTATAI IR IŠVADOS.....	75
	ŠALTINIAI.....	76
	PRIEDAI.....	80
	1 priedas. Vertinimas pagal CMMI-DEV .....	80
	2 priedas. Vertinimas pagal CMMI-SVC .....	103

## ĮVADAS

Sparti šiuolaikinė informacinių technologijų raida, didėjanti konkurencija tarp verslo įmonių bei daugybė kitų veiksnių keičia aplinką, kurioje turi dirbti programų sistemas kuriančios bei jas eksploatuojančios įmonės [Vir15]. Viena iš pagrindinių bendrų problemų, su kuria susiduria rinkai skirtas programų sistemas kuriančios bei jas naudojančios įmonės: greitesnis ir dažnesnis programų sistemų atnaujinimas [WBL14]. Šiuolaikiniam naudotojui nėra priimtina laukti programų sistemos atnaujinimo mėnesius, naudotojai tikisi greito atsako iš programų sistemas kuriančių įmonių ir nuolatos besikeičiančius naudotojų poreikius. Nespėjant reaguoti į besikeičiančius naudotojų reikalavimus, programų sistema gali tapti nebeaktuali ir pamiršta. Todėl, norint turėti rinkoje konkurencingas, paklausias, šių dienų aktualijas atitinkančias programų sistemas, jas kuriančioms įmonėms labai svarbu gauti pastovų atgalinį ryšį iš savo programų sistemos naudotojų tam, kad kuriama sistema visiškai atitiktų naudotojų poreikius ir lūkesčius [WBL14]. Pasaulinės sėkmės ir pripažinimo susilaukusios įmonės kaip, pavyzdžiui, „Facebook“, atnaujinimus leidžia kiekvieną dieną [Par17]. Akivaizdu, jog gebėjimas greitai ir dažnai atnaujinti programų sistemą savo naudotojams, sukuria didžiulę vertę tiek sistemos naudotojams, tiek ją kuriančiam verslui, tačiau, kad tai pasiektų, įmonė turi gebėti būti judri (angl. agile) visuose programų sistemos gyvavimo ciklo etapuose, kadangi atsiradus bent vienai daliai nesugebančiai išlaikyti tempo, visas procesas sulėtėja – toji dalis tampa viso proceso silpnąją vieta (angl. bottleneck) [Vir15].

Judriosios metodikos, pavyzdžiui, Scrum, XP, yra labai populiarios tarp įvairių programų sistemas kuriančių įmonių, kadangi šios metodikos padeda sumažinti programų sistemų kūrimo laiką, dėl ko sumažėja ir programų sistemų kūrimo kaštai [OW15]. Naudotojai taip pat iš to gauna didelę vertę, kadangi į jų poreikius atsižvelgiama daug dažniau ir galutinis produktas būna gerokai arčiau to, ko naudotojams iš tikrųjų reikia [OW15].

Deja, judriosios metodikos visą dėmesį skiria programų sistemų kūrimui (angl. development), pamiršdamos eksploatavimo (angl. operations) procesą [Vir15]. Tradiciškai, įmonėse kūrimo ir eksploatavimo komandos dirba atskirai viena nuo kitos, kiekviena turėdama savo individualius tikslus. Susidaro tokia situacija, kai kūrimo komanda siekia greičiau pridėti naują funkcionalumą, spręsti sistemos klaidas ir leisti naujas versijas, o eksploatavimo komanda siekia išlaikyti patikimą infrastruktūrą, kur naudotojas visada gali pasiekti esamą programų sistemos versiją. Kitaip sakant, kūrimo ir eksploatavimo komandos siekia visiškai skirtingų tikslų – kūrimo komanda nori atlikti pakeitimus, o eksploatavimo – išlaikyti stabilumą, nieko nekeisti [Htt12]. Tai, žinoma, yra problema, kadangi abi komandos vis tiek priklauso viena nuo kitos leidžiant naujas programų sistemos versijas naudotojams. Tokia atskirtis tarp kūrimo ir eksploatavimo komandų daro neigiamą poveikį

[WEH15]. Dėl šios atskirties, tarp dviejų glaudžiai susijusių komandų, smarkiai padidėja nesusikalbėjimo ar kitų konfliktų galimybė [WEH15].

Taigi akivaizdu, jog bendradarbiavimas tarp šių dviejų disciplinų yra labai svarbi dalis, siekiant sukurti sklandų procesą, kuris įgalintų dažnesnius atnaujinimus programų sistemos naudotojams, išlaikant aukštą kokybę. Kitais žodžiais tariant, jeigu kūrimo ir eksploatavimo komandos nepradės veikti išvien, visas programų sistemų naujų versijų leidimo procesas išliks lėtas, dėl ko atgalinis ryšys iš naudotojų programų sistemų kūrėjus pasieks lėčiau ir ženkliai rečiau, nei kad kūrimo ir eksploatavimo komandoms dirbant kartu [WZ06]. Didelis atsako laikas į nuolatos besikeičiančius programų sistemos naudotojų reikalavimus gali padaryti didelę žalą verslui. Sprendžiant šią problemą, siekiant panaikinti atskirtį tarp programų sistemų kūrimo ir eksploatavimo komandų, buvo pasiūlyta „DevOps“ idėja. Šią sąvoką pirmą kartą oficialiai paminėjo Patrick Debois 2009 metais. Tai yra dviejų anglišku žodžių darinys – kūrimas (angl. development) ir eksploatavimas (angl. operations) [Htt12]. DevOps pabrėžia bendradarbiavimą ir tarpusavio ryšį tarp šių dviejų disciplinų, o tai skatina greitesnius programų sistemų kūrimo bei diegimo ciklus [OW15].

Nuo 2009 metų DevOps susilaukė daug dėmesio ir ši idėja smarkiai paplito – tokios įmonės kaip Netflix, Flickr, Yahoo ar Facebook savo procesuose taiko DevOps [Par17, BCC+13]. Tačiau nors vis daugiau įmonių pradeda taikyti DevOps ir supranta šios idėjos teikiamą naudą, vis dar nėra praktiškai taikomo DevOps brandos modelio. Dėl šio trūkumo, įmonėms yra sunku pradėti taikyti DevOps praktikas, kadangi nėra aišku, kokias praktikas reikia pradėti taikyti pirmiausia, kaip žinoti, jog tos praktikos teisingai taikomos, kaip po to palaipsniui tobulinti savo DevOps bei kelti proceso brandos lygį. Taip pat, įmonėms, kurios jau taiko DevOps, yra sunku įsivertinti dabartinį savo įmonės DevOps brandos lygį. Literatūroje galima rasti DevOps modelių, pavyzdžiui, Samer I. Mohamed [Moh15], Helen Beal [Bea14], Capgemini [MM15] ar IBM [Bah13] DevOps brandos modeliai, tačiau jų detalumas nėra pakankamas tam, kad tuos modelius būtų galima taikyti praktiškai.

Siekiant išspręsti anksčiau įvardintą problemą, formuluojamas magistro baigiamojo **darbo tikslas** – pasiūlyti DevOps brandos modelį, kurio detalumas leistų modelį taikyti praktiškai, bei nustatyti, kokius CMMI-DEV ir CMMI-SVC gebėjimo profilius užtikrina pasiūlytas DevOps brandos modelis.

Tikslui pasiekti, darbo pradžioje buvo atrinkti ir pristatyti detaliesi mokslinėje literatūroje randami DevOps brandos modeliai. Modeliai buvo išanalizuoti ir nustatyta, jog šiuo metu nėra DevOps brandos modelio, kurio detalumas leistų jį taikyti praktiškai, ne tik tų modelių autoriams.

Taip pat, siekiant pasiūlyti naują praktiškai taikomą DevOps brandos modelį, buvo atlikta DevOps procese taikomų praktikų analizė. Jos metu buvo nustatyta kaip iš tiestų organizacijos vykdo DevOps, kokias praktikas taiko savo procese.

Kuriant naują DevOps brandos modelį buvo priimti principiniai sprendimai susiję su kuriamo modelio architektūra, brandos lygiais, proceso sritimis, praktikomis bei vertinimu. Sprendimai buvo paremti esamais DevOps brandos modeliais, DevOps vykdančių organizacijų procesais bei CMMI.

Kadangi DevOps apjungia kūrimą bei eksploatavimą, remiantis CMMI modeliais – CMMI-DEV [CMD10] (kūrimo modelis) ir CMMI-SVC [CMS10] (eksploatavimo modelis) buvo atliktas naujo DevOps brandos modelio vertinimas ir nustatyta kokius, kiekvieno šių modelių, gebėjimo profilius atitinka, naujai apibrėžtas DevOps brandos modelis.



# 1. DEVOPS BRANDOS MODELIAI

Siekiant pasiūlyti praktiškai taikomą DevOps brandos modelį, reikia išanalizuoti esamus DevOps brandos modelius. Šiame skyriuje analizuojami 7 atrinkti DevOps brandos modeliai. DevOps brandos modelių buvo ieškoma mokslinių darbų duomenų bazėse bei naudojantis paieškų sistema – Google. Modeliai buvo ieškomi naudojant raktažodžius: „DevOps maturity model“, „DevOps maturity“, „DevOps maturity assessment“. Atliekant paiešką DevOps brandos modelių buvo rasta daugiau nei yra pateikiama šiame skyriuje, tačiau dalis rastų brandos modelių buvo nepakankamai detalūs, pavyzdžiui, buvo pateiktas tik modelį apibendrinantis paveikslukas, be jokių paaiškinimų, kad juos būtų galima įtraukti į nagrinėtinų modelių sąrašą. Taip pat, buvo rastas 1 tolydinės architektūros modelis bei 1 dėmesio sričių architektūros modelis, kurie nors ir nėra brandos modeliai, bet kalba apie brandą ir dėl to buvo randami pagal pasirinktus raktažodžius, tačiau, šie modeliai nebuvo įtraukti į nagrinėtinų sąrašą, kadangi darbe siekiama pasiūlyti praktiškai taikomą brandos modelį, o ne kokį nors kitą. Šiame skyriuje pateikti DevOps brandos modeliai buvo atrinkti, kadangi turėjo detaliausius aprašymus, lyginant su visais kitais rastais brandos modeliais.

Atrinkti DevOps brandos modeliai šiame skyriuje nagrinėjami pagal šiuos kriterijus:

- Ar modelio brandos lygiai sutampa su kokio nors kito žinomo modelio, pavyzdžiui, CMMI ar ISO 15504, brandos lygiais?
- Ar yra pateikta brandos lygių samprata?
- Ar pateikti proceso sričių apibrėžimai?
- Ar pateiktos praktikos?
- Ar pradinis lygis suteikiamas automatiškai?
- Ar nurodyta vertinimo procedūra?
- Ar yra kokios nors kitos papildomos informacijos?

Taip pat, tam kad būtų galima geriau suprasti nagrinėjamo DevOps brandos modelio detalumą, pateikiamos pavyzdinės modelio elementų dalys, pavyzdžiui, vienos proceso srities samprata, viename brandos lygyje, jeigu tokia modelyje yra pateikiama. Pateikiamas nagrinėjamų DevOps brandos modelių eiliškumas yra atsitiktinis.

## 1.1. S. I. Mohamed DevOps brandos modelis

Samer I. Mohamed pasiūlytas DevOps brandos modelis [Moh15] yra sudarytas praplečiant Hewlett Packard DevOps brandos modelį [Bto13]. Kaip teigia šio modelio autorius, brandos modelis yra padarytas remiantis CMMI brandos modeliu. Jis buvo publikuotas 2015 metais. Modelyje yra 5 brandos lygiai ir 4 proceso sritys.

Brandos lygiai (nuo žemiausio iki aukščiausio):

- Pradinis
- Valdomas
- Apibrėžtas
- Matuojamas
- Optimizuotas

Brandos lygių pavadinimai šiame DevOps brandos modelyje sutampa su CMMI, tačiau samprata – ne. Pateikiama apibrėžto brandos lygio samprata:

- *3 brandos lygis: Apibrėžtas* – 3 brandos lygyje, organizacijoje vykdomos procedūros bei naudojami įrankiai yra aiškiai apibrėžti ir aprašyti organizacijos standartuose. Šie organizacijos standartai yra naudojami užtikrinti procedūrų vykdymo nuoseklumui organizacijoje tarp skirtingų komandų bei projektų. Kiekvieno projekto komanda turi galimybę nežymiai pakoreguoti procesą, esant specialioms poreikiams, tačiau tą darant, procesas vis tiek turi išlikti suderintas su organizacijoje priimtais standartais.

CMMI terminologijoje esančios proceso sritys, šiame DevOps brandos modelyje vadinamos dimensijomis. Tam, kad būtų galima pereiti iš vieno brandos lygio į kitą, organizacija turi gerinti visas 4 modelyje esančias proceso sritis.

Proceso sritys:

- Komunikacija
- Automatizacija
- Valdymas
- Kokybė

Autorius pateikia proceso sričių apibrėžimus. Kadangi, apibrėžimai kiekvienai proceso sričiai pateikiamas, kiekviename brandos lygyje bei atsižvelgiant į apibrėžimų turinį, galima suprasti, jog tai, iš tiesų, yra patikslinta brandos lygių samprata. Tam, kad būtų galima susidaryti įspūdį apie šio DevOps brandos modelio proceso sričių apibrėžimo detalumą, pateikiami proceso sričių, pirmame ir trečiame brandos lygiuose, aprašymai:

- Pradinis (pirmasis) brandos lygis
  - Komunikacijos proceso sritis – komunikacija šiame lygyje paprastai vykdoma reaktyviai (angl. ad-hoc), be jokių aiškiai apibrėžtų procesų ar naudojamų įrankių. Nėra aiškių rolių ir atsakomybių, o tai lemia chaotišką aplinką ir bendravimą. Sprendimai centralizuoti.
  - Automatizacijos proceso sritis – paprastai šiame lygyje nėra jokios automatizacijos, dauguma užduočių atliekama rankiniu būdu.

- Valdymo proceso sritis – valdymas šiame lygyje vykdomas reaktyviai ir vykdomų procedūrų rezultatai nebūna numatomi iš anksto. Nėra jokių vykdymo kontrolės priemonių.
- Kokybės proceso sritis – kokybė nėra išreikštinai vertinama ir pagrinde užtikrinama reaktyviu būdu, remiantis proceso iniciatoriaus iniciatyva.
- Apibrėžtas (trečiasis) brandos lygis
  - Komunikacijos proceso sritis – komunikacija yra labiau apibrėžta tarp skirtingų komandų, nepriklausomai nuo to, ar jos yra lokaliai ar skirtingose fizinėse vietose, naudojantis atitinkamais įrankiais bei mechanizmais. Komunikacijos valdymas reikšmingas ir suprantamas kaip organizacijos valdymo procedūrų standarto dalis. Komandos įtraukiamos į komunikaciją iš anksto, pavyzdžiui, kokybės užtikrinimo komanda bendradarbiauja su kūrimo komanda ankstyvajame programų sistemos kūrimo etape tam, kad būtų paruošti reikalingi testavimo scenarijai kiekvienam moduliui ar komponentui. Per komunikaciją, įtrankius, idėjų dalinimąsi, komandos įgalinamos pašalinti izoliaciją bei išvengti darbų dubliavimo.
  - Automatizacijos proceso sritis – automatizacija šiame brandos lygyje yra organizacijos standartas. Organizacija komandoms suteikia automatizacijos šabloną (angl. framework), infrastruktūrą, techninį apmokymą bei įrankius, tam, kad būtų galima geriau įsisavinti šio etapo rezultatus, siekiant optimalaus tikslo – maksimaliai padidinti organizacijos našumą.
  - Valdymo proceso sritis – valdymas šiame brandos lygyje yra standartizuotas. Standartas yra priimtas visoje organizacijoje ir valdomas už standarto įgyvendinimą atsakingos komandos. Kiekvieno projekto komanda turi galimybę nežymiai pakoreguoti procesą, esant specialiems poreikiams, tuo pat metu vis tiek laikantis visos organizacijos proceso šablono.
  - Kokybės proceso sritis - kokybė šiame brandos lygyje yra priimta kaip organizacijos standartas. Kokybės užtikrinimo procesai bei įrankiai taikomi nuosekliai skirtinguose projektuose ir komandose. Defektų stebėjimas bei ryšiai tarp kokybės užtikrinimo ir kūrimo komandų skirtinguose kūrimo gyvavimo cikluose yra aiškiai apibrėžti. Nepriklausomai nuo to, kokia yra organizacijos kūrimo metodika, ar tai būtų krioklio, ar tai būtų judriosios metodikos, kokybės užtikrinimo komanda turi laikytis organizacijos standarto, tam kad pasiektų užbrėžtą tikslą.

Šaltinyje išreikštinau nenurodoma, ar pradinis brandos lygis yra duodamas bet kokiai veiklą vykdančiai informacinių technologijų įmonei, t.y. nėra jokių reikalavimų. Tačiau, pagal pateiktą pradinio brandos lygio aprašymą, galima daryti išvadą, kad pirmasis brandos lygis yra duodamas visiems. Taip pat, modelis kurtas remiantis CMMI, o CMMI pirmasis brandos lygis neturi jokių reikalavimų ir būna suteikiamas visiems.

1 pav. pateikiama S. I. Mohamed DevOps brandos modelio santrauka.

	PRADINIS	VALDOMAS	APIBRĖŽTAS	MATUOJAMAS	OPTIMIZUOTAS
KOMUNIKACIJA	Reaktyvi komunikacija	Kontroliuojama komunikacija ir bendradarbiavimas	Standartiniai komunikacijos procesai	Komunikacijos metrikų rinkimas srities proceso gerinimui	Konstruktivi komunikacijos aplinka, įrankiai, procesai
AUTOMATIZACIJA	Jokios automatizacijos	Reaktyvi automatizacija	Standartizuota automatizacija	Automatizacijos metrikų rinkimas siekiant įvertinti pažangą verslo tikslų kontekste	Protinga automatizacija su tikslu maksimizuoti našumą
VALDYMAS	Nekontroliuojamas valdymas	Nestandardizuotas valdymas	Standartizuotas valdymas	Valdymo metrikų rinkimas siekiant įvertinti proceso našumą	Optimizuotas valdymas
KOKYBĖ	Negyvuojantys kokybės standartai	Reaktyvus kokybės valdymas	Kokybės standartai egzistuoja	Kokybės metrikų rinkimas	Nuolatinis kokybės gerinimas

1 pav. S. I. Mohamed DevOps brandos modelis [Moh15].

S. I. Mohamed DevOps brandos modelyje nėra minimos konkrečios praktikos, kurios turėtų būti taikomos proceso srityse, duotajame brandos lygyje. Taip pat, nėra kalbama apie vertinimo procedūrą t.y. kada laikoma, jog proceso sritis konkrečiame brandos lygyje yra įgyvendinta. Dėl šių priežasčių, modelio negalima taikyti praktiškai.

## 1.2. I. Bucena ir M. Kirikova DevOps brandos modelis

I. Bucena ir M. Kirikova DevOps brandos modelis [BK17] buvo pristatytas 2017 metais. Nors darbe tas nėra sakoma, bet pagal naudojamą terminologiją ir modelio struktūrą, galima teigti, jog šis brandos modelis taip pat yra sudarytas, remiantis CMMI brandos modeliu. Jame yra 5 brandos lygiai ir 4 proceso sritys. Proceso sritys šiame darbe vadinamos verslo sritimis (angl. enterprise areas).

Brandos lygiai (nuo žemiausio iki aukščiausio):

- Pradinis
- Atkartojamas
- Apibrėžtas
- Valdomas
- Optimizuotas

Brandos lygių aprašymai darbe nėra pateikiami, apsiribojama jų pavadinimais. Taip pat, nėra pateikiamas paaiškinimas kodėl būtent tokie brandos lygiai ir kaip jie buvo parinkti. Tačiau, šiame

brandos modelyje naudojami brandos lygių pavadinimai sutampa su senu CMM modeliu – SW-CMM.

Proceso sritys nėra aprašomos, ar atskirai įvardijamos. Sritis galima matyti tik originaliaame darbe pateiktoje DevOps brandos modelio lentelėje.

Proceso sritys:

- Technologijos
- Procesas
- Žmonės
- Kultūra

Kiekvienai proceso sričiai ir brandos lygiui pateikiamos taikytinos praktikos, tačiau jos nėra aprašomos. Praktikos sunumeruotos ir joms keliami aukštesni reikalavimai, kylant brandos lygiui. Tai, nestandartinė praktikos struktūra, kadangi standartiškai praktikos būna įvardinamos tik vieną kartą, kažkuriame lygyje. Jeigu vietoj reikalavimų praktikai pateikiamas brūkšnio simbolis („–“), tai reiškia, jog susijusių reikalavimų aukštesniame brandos lygyje nebėra. Tam, kad būtų galima susidaryti įspūdį apie šaltinyje pateikto DevOps brandos modelio detalumą 1 lentelėje pateikiamos „Proceso“ proceso srities praktikos kiekvienam brandos lygiui.

1 lentelė. „Proceso“ proceso srities praktikos.

	<b>Pradinis</b>	<b>Atkartojamas</b>	<b>Apibrėžtas</b>	<b>Valdomas</b>	<b>Optimizuotas</b>
1.	Nenuoseklus pristatymo procesas	Suplanuotas pristatymo procesas	Automatizuotas pristatymo procesas	Dažnai vykdomo pristatymo procesas	Kūrimo procesas integruotas su „Six Sigma“
2.	Reaktyvus kūrimo procesas	Scrum taikymas kūrimo procese	Judriųjų metodikų taikymas kūrimo procese	Lean taikymas kūrimo procese	Nuolatinis testavimas
3.	Reaktyvus testavimas	Reikalavimais pagrįstas testavimas	Integruotas testavimas	Kokybinis testavimas	Organizuotas našumo valdymas
4.	Nenuoseklus projekto valdymas	Projektų ir reikalavimų valdymas	Integruotas projekto valdymas	Kiekybinis projektų valdymas	–
5.	Diegimo ir kūrimo dokumentacijos neegzistuoja arba yra pasenusios	Diegimo ir kūrimo dokumentacija ir susiję konfigūracijos failai nuolat atnaujinami	Dokumentacija reguliariai tikrinama ir pateikiami susijusių konfigūracijos failų aprašymai	Dokumentacijos procesas ir struktūra atnaujinama remiantis surinkta patirtimi ir kokybės reikalavimais	–
6.	Nekontroliuojami arba reaktyvus procesai	Procesai valdomi, bet ne standartizuoti	Procesai standartizuoti visos organizacijos mastu	Procesų matomumas ir nuspėjamumas bei našumas	Labai optimizuoti ir integruoti procesai

Pradinis brandos lygis aprašo blogiausią variantą, tad jis yra suteikiamas visoms veiklą vykdančioms įmonėms.

Žiūrint į pirmą lentelę, galima matyti, jog modelis palieka labai daug neatsakytų klausimų, kadangi tai yra visas praktikų paaiškinimas t.y. detaliau nesiplečiama. Kadangi, kiekviena praktika turi savo numerį ir jai yra keliami aukštesni reikalavimai, kylant brandos lygiui, galima įžiūrėti tam tikrų nelogiškų dalykų. Pavyzdžiui, antrajame brandos lygyje antroji praktika yra Scrum taikymas kūrimo procese, o trečiajame lygyje antroji praktika – judriųjų metodikų taikymas kūrimo procese. Scrum yra viena iš judriųjų metodikų [RJ00]. Be to, ketvirtajame lygyje jau reikalaujama taikyti Lean, kas nebėra judrioji metodika. Taip pat, dokumente nėra pasakoma, ar praktikos aukštesniame lygyje pakeičia žemesniame lygyje nurodytą praktiką, ar ją papildo. Pastarojo pavyzdžio atveju, su judriosiomis metodikomis, galima numanyti, kad pakeičia, tačiau, žiūrint į trečiąją praktiką, susijusią su testavimu, toks variantas nebūtų logiškas.

Papildomai, darbe pateikiami įrankiai, kurie galėtų būti naudojami, siekiant įgyvendinti modelyje nurodytas praktikas. Pateikiamas pavyzdys, kur taikytinai praktikai, priskiriami galimi įrankiai:

- Nuolatinis diegimas
  - Capistrano
  - Jenkins
  - Ansible
  - Codeship
  - Travis CI
  - Circle CI
- Stebėjimas
  - Zabbix
  - New Relic
  - Nagios
  - Splunk
  - AppDynamics

Darbe teigiama, kad modelis buvo patikrintas, pritaikant jį kompanijai, kurios pavadinimas nebuvo atskleistas, o dydis įvardijamas kaip vidutinis (~220-250 darbuotojų). Tačiau, bent jau atsižvelgiant į modelio detalumą bei įvardintas problemas, kurios nėra darbe paaiškintos, šio modelio ne autoriai negalėtų taikyti.

### 1.3. E. Minick DevOps brandos modelis

Eric Minick DevOps brandos modelį [Min14] pristatė 2014 metais, dirbdamas IBM. Pats autorius, šį modelį vadina nuolatinio pristatymo brandos modeliu (angl. Continuous Delivery Maturity Model). Tačiau, galima teigti, kad tai dar vienas DevOps brandos modelis, tik išskirtinai dėmesį skiriantis techninei daliai. Pats autorius teigia, kad nuolatinis pristatymas yra DevOps dalis. Šis modelis sukurtas, remiantis CMMI brandos modeliu ir anksčiau aptartieji turi 5 brandos lygius bei 4 proceso sritis. Proceso sritys šiame darbe vadinamos temomis (angl. topics).

Nagrinėjamo DevOps brandos modelio brandos lygių pavadinimai nesutampa su CMMI brandos lygiais, tad ir jų samprata yra kitokia. Bazinis lygis turi reikalavimus ir nėra duodamas automatiškai visoms veiklą vykdančioms organizacijoms. Taigi, jeigu įmonė neįgyvendina bazinio brandos lygio reikalavimų, jai nepriskiriamas joks lygis.

Brandos lygiai (nuo žemiausio iki aukščiausio):

- Bazinis – šis brandos lygis suteikiamas, kai galutinai atsisakoma pilnai rankinių procesų.

- Pradedantysis – taikomas kai kurias verslo nuolatinio pristatymo (angl. ECD - Enterprise Continuous Delivery) praktikos.
- Tarpinis – nuolatinio pristatymo praktikos taikomos šiek tiek aukštesniu lygiu ir jau duoda didesnę efektyvumą bei mažesnę incidentų kiekį. Didžiąją dalį įmonių, šis lygis būtų pakankamas.
- Pažengęs – komandos dirba aukštesniame lygyje nei dauguma rinkoje esančių organizacijų ir kaip rezultatą mato didesnę efektyvumą ir incidentų prevenciją. (Autorius nenurodo kaip žinoti, kaip dirba dauguma organizacijų).
- Ekstremalus – elementai esantys šiame lygyje yra tie, kuriuos brangu pasiekti, bet kai kurioms komandoms tai vis tiek turėtų būtų siektina.

Visos temos yra verslo nuolatinio pristatymo (ECD) kontekste. Pateiktos proceso sritys:

- Kodo kompiliavimas
- Diegimas
- Testavimas
- Raportavimas

Kiekviena proceso sritis, kiekviename brandos lygyje, šaltinyje aprašyta pastraipa. Taip pat pateikiamas trumpas autoriaus ne visai formalus komentaras pradžioje. Tam, kad būtų galima susidaryti įspūdį apie proceso srities apibrėžimo detalumą šiame DevOps brandos modelyje, pateikiamas „Testavimo“ proceso srities aprašymas:

- *Bazinis* – turi būti kažkokie automatiniai testai. Turėtų būti kažkiek modulių testų, kurie padengtų svarbiausią programų sistemos funkcionalumą.
- *Pradedantysis* – Norint pasiekti pradedančiojo brandos lygį turi būti greitų testų rinkinys, kuris būtų paleidžiamas kiekvieno kodo surinkimo metu. Šie testai, suteikia komandai pasitikėjimo, kad programų sistema, atlikus pakeitimus, veiks. Kai testų rinkinys baigiasi ne taip, kaip turėtų, dėl kodo esančios klaidos, kūrimo komanda apie tai būna automatiškai informuojama. Reagavimas į šiuos pranešimus yra svarbus pradedančiojo brandos lygio indikatorius, komanda kuri nereaguoja į tokius testų pranešimus, yra žemiau pradedančiojo lygio.
- *Tarpinis* – reikalaujama daugiau nei greitų, kodo surinkimo metu leidžiamų testų. Turi būti įvairių testų rinkinys. Papildomai turėtų būti automatiniai funkciniai testai, taip pat statinė kodo analizė turėtų būti vykdoma nuolatinės integracijos sistemoje. Statinė



analizė gali būti vykdoma nebūtinai kodo surinkimo metu, bet periodiškai. Visi saugumo trūkumai turėtų būti ištaisomi prieš programų sistemos išleidimą.

- *Pažengęs* – šis brandos lygis charakterizuojamas kaip testavimo kruopštumas. Kiekvienas testavimo tipas artėja prie galimybių ribos, kiek informacijos jis gali suteikti. Modulių testai padengia beveik visą sudėtingiausią programų sistemos kodą. Funkciniai testai padengia visą svarbų sistemos funkcionalumą. Atsitiktinis testavimas taip pat turi būti taikomas. Statinė kodo analizė leidžiama dažnai, kartu su saugumo patikromis, tam, kad būtų pastebėtos problemos, kurių negali aptikti testai. Testai leidžiami keliose sistemose paraleliai, tam, kad atgalinis ryšys būtų gaunamas greičiau.
- *Ekstremalus* – komandos siekia 100% kodo padengimo testais. Komandos siekiančios tokio padengimo testais, turi priimti faktą, kad kažkiek testavimo darbo bus atliekama veltui.

Daugiau nei pateiktame proceso srities aprašymo pavyzdyje nėra plečiamasi. Pateikiamos praktikos modelį apibendrinančiame paveiksliuke (žiūrėti 2 pav.). Tačiau, papildomai informacijos apie praktikas, daugiau nei pateiktame paveiksliuke – nėra.

	BAZINIS	PRADEDANTYSIS	TARPINIS	PAŽENGŲS	EKSTREMALUS
KODO KOMPILIAVIMAS	<ul style="list-style-type: none"> <li>• Kodo kompiliavimo skriptas</li> <li>• Kodo kompiliavimo mašina</li> </ul>	<ul style="list-style-type: none"> <li>• Naktinis kodo kompiliavimas</li> <li>• Sukompiliuotų artefaktų saugojimas</li> </ul>	<ul style="list-style-type: none"> <li>• Automatinis kodo kompiliavimas po kodo pakeitimo</li> <li>• Kodo priklausomybių repozitorija</li> <li>• Apsaugota konfigūracija</li> </ul>	<ul style="list-style-type: none"> <li>• Iš išorės iššaukiamas kodo kompiliavimas</li> <li>• Kodo kompiliavimo mašinų klasteris</li> </ul>	<ul style="list-style-type: none"> <li>• Apsaugoti kodo pakeitimai</li> </ul>
DIEGIMAS	<ul style="list-style-type: none"> <li>• Diegimo skriptas</li> </ul>	<ul style="list-style-type: none"> <li>• Galimybė pačiam diegti į testinę aplinką</li> <li>• Automatinis diegimas į pirmąją aplinką</li> <li>• Daugmaž standartizuotas diegimas</li> </ul>	<ul style="list-style-type: none"> <li>• Galimybė pačiam diegti į testinę ir produkcinę aplinkas</li> <li>• Standartizuotas diegimo procesas per visas aplinkas</li> </ul>	<ul style="list-style-type: none"> <li>• Duomenų bazės diegimas</li> <li>• Koordinuotas SOA/kelių lygių programinės įrangos diegimas</li> </ul>	<ul style="list-style-type: none"> <li>• Nuolatinis diegimas į produkcinę aplinką</li> </ul>
TESTAVIMAS	<ul style="list-style-type: none"> <li>• Kaikurių testų automatizavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Svarbiausių testų paleidimas kodo kompiliavimo metu</li> </ul>	<ul style="list-style-type: none"> <li>• Kažkiek statinės kodo analizės</li> <li>• Automatizuoti funkciniai testai leidžiami kasnakt</li> </ul>	<ul style="list-style-type: none"> <li>• Aukštas kodo testais padengimo lygis</li> <li>• Saugumo patikrinimai</li> <li>• Rizikom paremtas rankinis testavimas</li> </ul>	<ul style="list-style-type: none"> <li>• 100% kodo padengimas automatiniais testais</li> </ul>
RAPORTAVIMAS	<ul style="list-style-type: none"> <li>• Matomumas: Raportų generavimo įrankis</li> <li>• Įrankiu sugeneruoti raportai</li> </ul>	<ul style="list-style-type: none"> <li>• Matomumas: Komanda</li> <li>• Naujausi raportai visada prieinami</li> </ul>	<ul style="list-style-type: none"> <li>• Matomumas: Visos komandos</li> <li>• Istoriniai raportai prieinami</li> </ul>	<ul style="list-style-type: none"> <li>• Raportų tendencijų stebėjimas</li> </ul>	<ul style="list-style-type: none"> <li>• Tarpkomandinė analizė</li> </ul>

2 pav. E. Minick IBM DevOps brandos modelis 2014 [Min14].

Modelio autorius argumentuoja, jog pasirinko būtent šias proceso sritis, kadangi jos yra bendros visoms organizacijoms, ar tai būtų organizacija kurianti medicininius įrenginius, ar tai būtų organizacija, kuri kuria žaidimus. Iš to galima suprasti, kad proceso sričių galėtų būti ir daugiau, tačiau jis nusprendė jų neįtraukti. Nėra konkrečių įrodymų, kad šis modelis būtų taikomas praktiškai,

tačiau, kadangi tai yra IBM darbuotojo pasiūlytas modelis, galima numanyti, jog modelis galbūt buvo arba tebėra taikomas IBM viduje.

## 1.4. P. Bahrs DevOps brandos modelis

Literatūroje randamas ir kitas, IBM darbuotojo Paul Bahrs paskelbtas DevOps brandos modelis [Bah13]. Šaltinio autorius šį modelį vadina DevOps praktikomis paremtu brandos modeliu. Kiekvienos proceso srities brandos lygis susietas su DevOps praktikomis, kurios turi būti taikomos siekiant užtikrinti pasirinktą DevOps brandos lygį. Brandos modelis turi 4 brandos lygius bei 4 proceso sritis. Proceso sritys šiame darbe vadinamos įsisavinimo keliais (angl. adoption paths).

Brandos lygiai šiame DevOps brandos modelyje nesutampa su CMMI brandos lygiais. Šis modelis netgi turi vienu lygiu mažiau. Modelio autorius atsisako pradinio lygmens, kuris būna automatiškai suteikiamas įmonėms vykdančioms veiklą. Dokumente nepateikiama pasirinktų brandos lygių samprata, tik nurodoma jų eilės tvarka, nuo žemiausio iki aukščiausio:

- Praktikuojamas
- Atkartojamas
- Patikimas
- Išplėstas

Pateiktos proceso sritys:

- Planavimas / Matavimas
- Kūrimas / Testavimas
- Išleidimas / Diegimas
- Stebėjimas / Optimizavimas

Kiekviena proceso sritis šaltinyje aprašoma pastraipa, kuri apibrėžia proceso sritį kiekviename brandos lygyje. Tam, kad būtų galima susitaryti išpūdi apie šio brandos modelio proceso sričių apibrėžimo detalumą modelyje, pateikiamas Planavimo ir Matavimo proceso srities aprašymas:

- Praktikuojamame brandos lygyje, organizacijos fiksuoja kiekvieno projekto verslo atvejus (angl. business cases) ar tikslus, kad apibrėžtų strategijos taikymo sritį, tačiau projekto išteklių valdomi departamento lygiu. Kai projektas jau būna vykdomas, pakeitimai ir jų apimtis yra valdomi projekto kontekste taip, kad būtų pasiekti tikslai susiję su kaštais ir laiku. Organizacijai tampant brandesne, verslo poreikiai yra dokumentuojami verslo kontekste ir matuojami taip, kad atitiktų klientų vertės metrikas. Šie verslo poreikiai vėliau būna prioretizuojami ir suderinami su programų sistemos išleidimu bei susiejami su projekto reikalavimais. Projekto pakeitimai ir jų apimtis būna valdomi portfelio (angl. portfolio) lygiu.

Aprašymą sunku suprasti, kadangi jis ganėtinai abstraktus. Be to, kad pradžioje paminėtas pirmasis brandos lygis, vėliau pateiktame aprašyme nėra išskiriami kiti brandos lygiai, neišku, kur pasibaigia vienas lygis ir prasideda kitas.

Praktikos yra pateikiamos 3 paveiksluke. Be antraščių paveiksluke, daugiau apie praktikas darbe nėra kalbama, praktikos nedetalizuojamos.

	PRAKTIKUOJAMAS	ATKARTOJAMAS	PATIKIMAS	IŠPLĖSTAS
<b>PLANAVIMAS MATAVIMAS</b>	<ul style="list-style-type: none"> <li>Tikslai dokumentuoti lokaliai</li> <li>Valdomi departamento resursai</li> </ul>	<ul style="list-style-type: none"> <li>Tikslai sujungti su išleidžiamomis versijomis</li> <li>Centralizuotas reikalavimų valdymas</li> <li>Matuojama projekto metrikomis</li> </ul>	<ul style="list-style-type: none"> <li>Strateginis planavimas</li> <li>Bendras metrikų stebėjimas vienoje vietoje</li> </ul>	<ul style="list-style-type: none"> <li>Išleidžiamos versijos sujungtos to verslo tikslais</li> <li>Matuojama klienti/vartotojui teikiama vertė</li> </ul>
<b>KŪRIMAS TESTAVIMAS</b>	<ul style="list-style-type: none"> <li>Valdomas artefaktų gyvavimo ciklas</li> <li>Suplanuoti automatiniai sistemos surinkimai</li> </ul>	<ul style="list-style-type: none"> <li>Sujungta artefaktų gyvavimo ciklo informacija</li> <li>Pristatymas ir sistemos surinkimas susietas su testais</li> <li>Automatinis testavimas</li> </ul>	<ul style="list-style-type: none"> <li>Valdomi duomenys ir virtualios paslaugos testavimui</li> <li>Nuolatinė naujo kodo integracija</li> </ul>	<ul style="list-style-type: none"> <li>Nuolatinis kodo kokybės gerinimas naudojantis automatinio kodo analizavimu</li> <li>Nuolatinis testavimas</li> </ul>
<b>IŠLEIDIMAS DIEGIMAS</b>	<ul style="list-style-type: none"> <li>Planuojami ir valdomi versijų išleidimai</li> <li>Standartizuotas diegimas</li> </ul>	<ul style="list-style-type: none"> <li>Planuojami skirtingų departamentų išleidimai ir automatizuotas esamas statusas</li> </ul>	<ul style="list-style-type: none"> <li>Standartizuota ir automatizuota tarp dukterinių įmonių</li> <li>Automatizuotas patarnais paremtas aplinkos kūrimas ir diegimas</li> </ul>	<ul style="list-style-type: none"> <li>Automatinis aplinkų valdymas</li> <li>Galimybė pačiam sukurti aplinką, surinkti kodą ir įdiegti automatizuotu būdu</li> </ul>
<b>STEBĖJIMAS OPTIMIZAVIMAS</b>	<ul style="list-style-type: none"> <li>Resursai nuosekliai stebimi</li> <li>Neformalus bendradarbiavimas tarp sistemos kūrimo ir veikimo komandų</li> </ul>	<ul style="list-style-type: none"> <li>Stebėjimas atsižvelgiant į galutinio vartotojo ir verslo kontekstus</li> <li>Centralizuoti, susieti įvykių pranešimai ir jų sprendimai</li> </ul>	<ul style="list-style-type: none"> <li>Optimizuotos programų sistemos</li> <li>Naudojamos bendros įmonės problemų sprendimo procedūros</li> </ul>	<ul style="list-style-type: none"> <li>Automatizuotas problemų izoliavimas ir jų sprendimas</li> <li>Nuolatinis vartotojo KPI optimizavimas</li> </ul>

3 pav. Paul Bahrs IBM DevOps brandos modelis 2013 [Bah13].

Šis DevOps brandos modelis pagrindinį dėmesį teikia techniniams aspektams. Deja, dėl neaiškių proceso sričių aprašymų ir nedetalizuotų praktikų šį DevOps brandos modelį taikyti praktiškai būtų negalima. Visgi, kadangi šis modelis buvo pateiktas IBM darbuotojo, gali būti, kad yra detalesnis šio brandos modelio variantas, kuris nėra prieinamas viešai ir jis, galbūt, taikomas praktiškai.

## 1.5. Capgemini DevOps brandos modelis

Capgemini – tarptautinė informacinių technologijų paslaugas teikianti įmonė. Įmonė išleido straipsnį (angl. white paper), kuriame pristatė savo DevOps brandos modelį [MM15]. Modelis sudarytas, remiantis CMMI brandos modeliu. Yra 5 brandos lygiai, taip pat modelis turi 3 procesų sritis.

Brandos lygiai šiame DevOps brandos modelyje nesutampa su CMMI brandos lygiais. Šaltinyje taip pat nepateikiama duotųjų brandos lygių samprata, tik jų pavadinimai išvardinti nuo žemiausio lygio iki aukščiausio:

- Bazinis
- Atsirandantis
- Suderintas
- Pagerintas
- Aukščiausias

Proceso sritys šiame DevOps brandos modelyje vadinamos fokuso sritimis (angl. focus areas).

Pateiktos 3 proceso sritys:

- Žmonės
- Procesas
- Įrankiai

Dokumente pateikti visų proceso sričių kiekviename brandos lygyje aprašymai. Tam, kad būtų galima susidaryti bendrą įspūdį apie šio brandos modelio proceso sričių aprašymo detalumą, pateikiami brazilio bei atsirandančio brandos lygių, proceso sričių aprašymai:

- Bazinis
  - *Žmonės* – atskiros strateginės, projektavimo, kūrimo, testavimo, eksploatavimo komandos. Jokių bendrų susitikimų, komandos fokusuojasi ties savo tiesioginėmis užduotimis. Atskiros komandos neturi bendrų tikslų. Nėra organizacijos mastu vykdomų skatinimo programų. Žmonės jaučiasi atsakingi tik už savo tiesiogiai vykdomas pareigas.
  - *Procesas* – komandose vykdomos atskiri ir nesusiję procesai, kurie yra reaktyvūs ir chaotiški. Niekur nėra aprašytas bendras proceso šablonas, kuris nusakytų, kaip reikia tinkamai vykdyti procesą.
  - *Įrankiai* – jokių automatizacijos įrankių, dauguma veiklų atliekamos rankiniu būdu, reaktyviai ir yra nesuplanuotos. Nėra integracijos tarp kompiuterinės įrangos tiekimo, operacinių sistemų diegimo ir konfigūravimo, programinės įrangos diegimo. Informacija saugoma skirtingose vietose, sunkiai randama.
- Atsirandantis
  - *Žmonės* – nedaug pasikeitimų nuo bazinio brandos lygio. Vis dar izoliuotos ir atskiros komandos, be vienos komandos ar žmogaus, kuris galėtų prisiimti pilną atsakomybę. Programuotojai pagrindė fokusuojasi ties funkciniais reikalavimais

ir labai nežymiai ties ne funkciniais. Yra šiek tiek bendrų sąlyčio taškų tarp kūrimo ir eksploatavimo komandų.

- *Procesas* – nedaug pasikeitimų nuo bazinio brandos lygio. Pradėti diegti valdymo procesai.
- *Įrankiai* – nežymūs pasikeitimai nuo bazinio lygmens, naudojami automatiniai scenarijai, skirti paruošti operacines sistemas darbui. Tai daugiausia apima kūrimo aplinkos paruošimą, tačiau kitos aplinkos, pavyzdžiui, testavimo, vis dar ruošiamos rankiniu būdu.

4 paveiksluke galima matyti abstraktų visą Capgemini DevOps brandos modelio vaizdą.



4 pav. Capgemini DevOps brandos modelis [MM15].

Praktikos, reikalingos proceso sritims kiekviename brandos lygyje, nėra nurodomos. Taip pat, brandos modelį aprašančiame dokumente nekalbama apie vertinimą, kada laikoma, kad proceso sritis įgyvendinta. Dėl šių priežasčių ir nepakankamo brandos modelio detalumo, šį modelį praktiškai taikyti nebūtų galima.

## 1.6. H. Beal DevOps brandos modelis

H. Beal DevOps brandos modelis [Bea14] – tai dar vienas modelis, randamas literatūroje. Modelį sudaro 5 brandos lygiai, kurių samprata nėra pateikta. Brandos lygiai matomi tik šaltinyje pateiktoje lentelėje.

Brandos lygiai (nuo žemiausio iki aukščiausio):

- Nepradėtas – nutraukimai, kaltinimai, neplanuotas darbas, vėlavimai, klaidos.
- Pradedamas – galvojimas apie kultūros pokyčius, pradedami rašyti kodo scenarijai, pradedama galvoti apie testų automatizavimą.
- Fundamentalus – automatizuotas kodo surinkimas, daugiafunkcinės komandos, fokusuojamasi ties produktu, kultūriniai pokyčiai pradeda vykti.
- Valdomas – laimingi žmonės su integruota įrankių grandine, mažinančia klaidų kiekį, automatizuotas testavimas ir diegimas.
- Optimizuojamas – DevOps įgyvendintas. Tobulinama ir optimizuojama derinant su verslo tikslais.

Proceso sritys šaltinyje pateiktame brandos modelyje nėra išskirtos, tačiau autorius mini 3 automatizacijos sritis, kurios galėtų būtų suvokiamos kaip proceso sritys:

- Išleidimas ir diegimas
- Testavimas ir paslaugų virtualizacija
- Našumo valdymas

Deja, pačioje modelio vizualizacijoje, kuri matoma modelio šaltinyje, šios proceso sritys nėra pavaizduotos ir esančios praktikos nėra tiesiogiai susietos su šiomis sritimis.

Praktikos pateiktos tik kartu su brandos lygių aprašais. Detalesnio jų aprašymo darbe nėra.

Taip pat, nėra kalbama apie vertinimą. Tokį modelį taikyti praktiškai negalima, tikėtina, kad organizacija, kuri šį modelį naudoja, turi daug detalesnę versiją, tačiau literatūroje randama tik šį versija.

## 1.7. Solinea DevOps brandos modelis

Solinea yra didelė įmonė, kurios viena iš paslaugų yra pagalba įsidiesti DevOps praktikas, taikant Solinea DevOps brandos modelį [Sol16], arba įsivertinti esamą DevOps brandos lygį, tam, kad būtų galima kelti lygį aukščiau. Turbūt, dėlto, kadangi tai yra šios organizacijos verslo dalis, Solinea DevOps brandos modelis nėra detalai aprašytas. Randama tik grafinė vizualizacija, kurią galima matyti penktame paveiksliuke, su antraštėmis kiekvieno brandos lygio proceso srityse. Solinea siūlomas DevOps brandos modelis yra sudarytas, remiantis CMMI brandos modeliu, turi 5 brandos lygius bei 4 proceso sritis.

Solinea DevOps brandos lygių pavadinimai sutampa su CMMI brandos lygių pavadinimais. Darbe nėra pasakoma, ar modelio brandos lygių samprata taip pat sutampa su CMMI, tačiau sava brandos lygių samprata taip pat nėra pateikiama.

Brandos lygiai (nuo žemiausio iki aukščiausio):

- Pradinis
- Valdomas

- Apibrėžtas
- Kiekybiškai valdomas
- Optimizuojamas

Modelyje galima matyti 4 proceso sritis, tačiau autoriai neįvardina proceso sričių kaip proceso sričių ar kitu pavadinimu. Į proceso sritis nurodoma kaip į 4 stulpus (angl. Four pillars). Nėra pateikiami proceso sričių aprašymai, tik jų pavadinimai:

- Technologijos
- Žmonės
- Procesas
- Kultūra

Praktikas galima matyti tik 5 paveiksliuke. Išsamesnių praktikų aprašymų nėra pateikta. Praktikos darbe vadinamos brandos elementais (angl. maturity items).

	PRADINIS	VALDOMAS	APIBRĖŽTAS	KIEKYBIŠKAI VALDOMAS	OPTIMIZUOJAMAS
<b>KULTŪRA</b>	<ul style="list-style-type: none"> <li>• Neiškomunikuota vizija</li> <li>• Apribotas bendravimas</li> <li>• Mažai/Jokios inovacijos</li> </ul>	<ul style="list-style-type: none"> <li>• Aiškūs užduoties reikalavimai</li> <li>• Greita komandinė komunikacija</li> <li>• Inovacija pagal poreikį</li> </ul>	<ul style="list-style-type: none"> <li>• Iškomunikuoti verslo tikslai</li> <li>• Aiškūs projekto reikalavimai</li> <li>• Greita tarp-komandinė komunikacija</li> <li>• Inovacija projektuojant</li> </ul>	<ul style="list-style-type: none"> <li>• Iškomunikuota verslo vizija</li> <li>• Aiškūs produkto reikalavimai</li> <li>• Dažna bendradarbiavimo komunikacija</li> <li>• Strateginė inovacija</li> </ul>	<ul style="list-style-type: none"> <li>• Iškomunikuota verslo strategija</li> <li>• Aiškūs verslo reikalavimai</li> <li>• Greitai gaunamas atgalinis ryšys</li> <li>• Atsakomybės / rizikos prisiėmimas</li> </ul>
<b>PROCESAS</b>	<ul style="list-style-type: none"> <li>• Rankinis procesas</li> <li>• Nenuoseklus projekto valdymas</li> <li>• Nenuoseklus rezultatų pateikimas</li> <li>• Ad Hoc kūrimas / testavimas</li> <li>• Nenuoseklus raportavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Paprasti scenarijai</li> <li>• Projekto ir reikalavimų valdymas</li> <li>• Suplanuotas pristatymo procesas</li> <li>• Scrum taikymas kūrimo procese</li> <li>• Reikalavimais pagrįstas testavimas</li> <li>• Taktinis raportavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Paprastas orchestravimas</li> <li>• Integuotas projekto valdymas</li> <li>• Automatizuotas diegimas</li> <li>• Agile taikymas kūrimo procese</li> <li>• Integuotas testavimas</li> <li>• Bendras raportavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Kompleksinis orchestravimas</li> <li>• Kiekybinis projekto valdymas</li> <li>• Dažnas diegimas</li> <li>• Kokybinis testavimas</li> <li>• Lean taikymas kūrimo procese</li> <li>• Strateginis raportavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Paskirstytas orchestravimas</li> <li>• Organizuotas našumo valdymas</li> <li>• Nuolatinis diegimas</li> <li>• Six Sigma</li> <li>• Nuolatinis testavimas</li> <li>• Nuspėjamas raportavimas</li> </ul>
<b>ŽMONĖS</b>	<ul style="list-style-type: none"> <li>• Reaktyvūs</li> <li>• Darbo bilietai</li> <li>• Komandos organizuotos pagal įgūdžius</li> <li>• Nerašytinės žinios</li> </ul>	<ul style="list-style-type: none"> <li>• Darbo bilietai ir scenarijų kūrimas</li> <li>• Komandos organizuotos pagal atliekamą darbą</li> <li>• Geri technologiniai įgūdžiai</li> <li>• Rašytinės žinios</li> </ul>	<ul style="list-style-type: none"> <li>• Scenarijų palaikymas ir orchestracijų kūrimas</li> <li>• Komandos organizuotos pagal projektus</li> <li>• Geri technologiniai įgūdžiai</li> <li>• Automatizacija kaip dokumentacija</li> </ul>	<ul style="list-style-type: none"> <li>• Orchestracijų palaikymas</li> <li>• Komandos organizuotos pagal produktus</li> <li>• Geri technologiniai įgūdžiai</li> <li>• Bendros žinios</li> </ul>	<ul style="list-style-type: none"> <li>• Proaktyvūs</li> <li>• Tarpdisciplininės komandos organizuotos pagal KPI</li> <li>• Nuolatinis mokymasis</li> <li>• Bendrų žinių perdavimas</li> </ul>
<b>TECHNOLOGIJOS</b>	<ul style="list-style-type: none"> <li>• Nėra bendradarbiavimo įrankių</li> <li>• Nėra artefaktų valdymo</li> <li>• Minimalus stebėjimas</li> <li>• Nėra automatizacijos</li> <li>• Minimalus testavimo įrankių rinkinys</li> <li>• Rankinis aplinkų kūrimas</li> </ul>	<ul style="list-style-type: none"> <li>• Projekto planavimo įrankiai</li> <li>• Svarbiausių dalių stebėjimas</li> <li>• Kodo surinkimo automatizacija</li> <li>• Funkcinio testavimo įrankiai</li> <li>• Standartizuotas aplinkų surinkimas</li> </ul>	<ul style="list-style-type: none"> <li>• Žinių valdymas</li> <li>• Komandų / Įrankių integracija</li> <li>• Kodo surinkimo / testavimo automatizacija</li> <li>• Integuotas stebėjimas</li> <li>• Ne produkcinės aplinos diegimo automatizacija</li> <li>• Automatizuotas aplinkų kūrimas</li> </ul>	<ul style="list-style-type: none"> <li>• Analitika</li> <li>• Savaiminis problemų sprendimas</li> <li>• Produkcinės aplinkos diegimo automatizacija</li> <li>• Kokybinio testavimo automatizacija</li> </ul>	<ul style="list-style-type: none"> <li>• Įrankiai kaip produktas</li> <li>• Įrankių rinkinio optimizavimas</li> </ul>

5 pav. Solinea DevOps brandos modelis [Sol16].

Apie vertinimą taip pat nešnekama, tačiau minima, jog Solinea gali atlikti organizacijos brandos lygio vertinimą.

Vis tik, modelis palieka abejonių. Matyti ta pati loginė klaida kaip ir jau identifikuota I. Bucena ir M. Kirikova DevOps brandos modelyje su judriosiomis metodikomis: „Proceso“ proceso srityje antrajame lygyje reikalaujama taikyti Scrum metodiką, o trečiajame – Agile. Scrum yra vienas iš Agile procesų, tad toks reikalavimas trečiajame lygyje nėra logiškas. Kadangi šis modelis yra senesnis, tikėtina, kad I. Bucena ir M. Kirikova tai perkėlė pas save iš šio modelio.

Nepakankamas detalumas neleidžia šio modelio taikyti praktiškai, tai galėtų daryti tik modelio autoriai.

## 1.8. Rezultatai

2 lentelė. DevOps brandos modelių detalumo apibendrinimas.

Modelis	Brandos lygių pavadinimai sutampa su	Brandos lygių samprata	Proceso sričių apibrėžimai	Praktikos	Vertinimas
[Moh15]	CMMI	Patikslinta kiekvienai proceso sričiai	Išsamūs	-	-
[BK17]	SW-CMM	-	-	Tik pavadinimai	-
[Min14]	-	Neišsami	Neišsamūs	Tik pavadinimai	-
[Bah13]	-	-	Neišsamūs	Tik pavadinimai	-
[MM15]	-	-	Vidutiniškai išsamūs	-	-
[Bea14]	-	-	-	Tik pavadinimai, nesusietos su proceso sritimis	-
[Sol16]	CMMI	-	-	Tik pavadinimai	-

Išnagrinėjus atrinktus DevOps brandos modelius, paaiškėjo, jog šiuo metu nėra brandos modelio, kurį būtų galima praktiškai taikyti, ne tik tų modelių autoriams. Modeliai nėra išsamūs, dauguma nepateikia brandos lygių sampratos, ne visi modeliai turi nurodytas taikytinas praktikas ir nei vienas iš nagrinėtų modelių nepateikia vertinimo procedūros, pagal kurią būtų galima įvertinti, ar pasiektas norimas brandos lygis. 2 lentelėje yra pateikiamas nagrinėtų DevOps brandos modelių detalumo apibendrinimas.



## 2. DEVOPS PROCESSE TAIKOMOS PRAKTIKOS

Siekiant sukurti praktiškai taikomą DevOps brandos modelį, būtina suprasti, kokios yra esminės DevOps procese taikomos praktikos. Tą galima išsiaiškinti, kadangi daug informacinių technologijų įmonių taiko DevOps [Sta18], net ir neturėdamos DevOps brandos modelio, kuris joms padėtų įsivertinti bei efektyviau kelti savo DevOps brandos lygį.

Ieškant literatūroje minimų su DevOps procesu susijusių bei taikomų praktikų, galima pastebėti, kad minimos praktikos autorių yra grupuojamos į praktikų rinkinius, pavyzdžiui, [Lwa17] šaltinyje autorius išskiria matavimo, stebėjimo, dalinimosi bei kultūros grupes, kurios apjungia rinkinį praktikų, kurios prisideda prie tų pačių tikslų siekimo. Šie praktikų rinkiniai brandos modelio terminologijos kontekste atitinka proceso sritis. Dalis praktikų grupių, kartojasi skirtinguose šaltiniuose ir yra įvardijamos kaip itin svarbios, be kurių DevOps proceso taikymas neįsivaizduojamas. Pasikartojančios grupės bei konkrečios DevOps praktikos yra detaliau analizuojami šiame skyriuje. Šio skyriaus poskyriuose pateiktos praktikos nebuvo atrenkamos bei filtruojamos t.y. buvo imamos visos randamos praktikos, identifikuotose proceso srityse.

Šio skyriaus poskyrius galima suvokti kaip proceso sritis, kurios apjungia rinkinį konkrečių DevOps taikytinų praktikų, taip sugrupuojamos panašioms tikslams skirtos praktikos. Proceso sričių pavadinimai atitinka literatūroje randamų su DevOps procesu susijusių praktikų rinkinių pavadinimus. Pateikiamas proceso sričių eiliškumas yra atsitiktinis.

### 2.1. Kultūros proceso srities praktikos

Literatūroje DevOps minimas kaip judėjimas [KHD+16] ir šis judėjimas didelę reikšmę teikia ne vien technologijoms, bet ir socialiniam aspektui [Lwa17]. DevOps judėjimas socialiniame kontekste reikalauja pokyčių įmonių, norinčių taikyti DevOps, kultūroje.

Kultūra prasideda nuo saugios, didelio pasitikėjimo darbo aplinkos [DD16]. Tai pasireiškia pasitikėjimu darbuotojais, negąsdinant jų nuobaudomis už padarytas klaidas. Įvykus incidentui, užuot ieškant žmogiškosios klaidos, turėtų būti mąstoma bei ieškoma būdų kaip pakeisti, patobulinti esamą sistemą, kad ateityje būtų išvengiama tokių pat ar panašių klaidų. Tą galima būtų įgyvendinti taikant peržiūras pasibaigus incidentui (angl. post-mortem), kurių esmė suprasti priežastis, bet ne ieškoti kaltų [All12]. Esant saugiai aplinkai, turi būti skatinama atvira komunikacija [DD16], kur darbuotojai galėtų išreikšti savo nerimą ar idėjas nesivaržydami. Be to, DevOps kultūroje labai svarbus nuolatinis mokymasis, tobulėjimas bei savo žinių perdavimas, skleidimas organizacijos viduje [DD16]. Užtikrinus saugią aplinką bei suteikus reikiamas žinias, darbuotojai turėtų būti skatinami patys priimti sprendimus bei su jais ateinančias rizikas [Wie18], kas mažina biurokratiją bei pagerina įmonės našumą, kadangi sprendimus gali priimti labiausiai su problema susipažinęs žmogus. Taip pat, sprendžiant problemas, reikia stengtis atlikti mažus, izoliuotus pakeitimus, gaunant dažną atgalinį

ryšį [FF14]. Kitas, su DevOps kultūra susijęs aspektas – nuopelnų skyrimas komandai [FF14]. Komandos turi būti pastebėtos ar apdovanotos už savo pasiekimus, bet ne individualūs asmenys, kas reiškia, jog komandos turi suprasti, jog jos yra komandos ir jeigu pasiekiami geri rezultatai, tai juos pasiekia visa komanda, o ne kažkas. Lygiai taip pat esant nesėkmei – nesėkmė priimama kaip visos komandos nesėkmė, o ne bandoma rodyti pirštais ir ieškoti kaltų. DevOps yra apie bendradarbiavimą ir komandinį darbą.

Kultūros proceso srityje taikomos praktikos:

- Saugi darbo aplinka
- Pasitikėjimas darbuotojais
- Peržiūros po incidentų (angl. post-mortem) be kaltinimų
- Nuolatinis procedūrų tobulinimas
- Atvira komunikacija
- Nuolatinis mokymasis
- Žinių dalinimasis
- Darbuotojų įgalinimas priimti sprendimus
- Izoliuoti pakeitimai
- Maži pakeitimai
- Dažnas atgalinis ryšys
- Nuopelnai skiriami komandoms

## 2.2. Dalinimosi proceso srities praktikos

Informacija, susijusi su projektu, bei techninės, profesinės žinios turi būti nuolatos skleidžiamos tarp organizacijos darbuotojų, siekiant darbuotojų asmeninio tobulinimosi bei skatinant projekto informacijos mainus tarp komandų ir darbuotojų [NJT16].

Aktualios projekto informacijos prieinamumas, kai nereikia informacijos papildomai prašyti, ženkliai padidina darbo našumą, sumažina klaidų kiekį [KHD+16]. Kad tai būtų įgyvendinta, reikia, jog apie kiekvieną pasikeitimą projekte būtų pranešama ir ta informacija būtų matoma. Tą galima padaryti pasitelkiant technologijas, tokias kaip JIRA, Slack, Flowdock ir kitos [KHD+16]. Komunikacija turėtų būti daugiausiai rašytinė arba, esant gyvai komunikacijai, vėliau dokumentuota tam, kad kitą kartą iškilus tam pačiam klausimui, informacija galėtų būti randama. Be to, kaip teigia [Lwa17] šaltinis, automatizacija veikia, kaip procesas įgalinantis veiklų dokumentavimą. Žinių dalinimasis tarp kūrimo komandų gali būti įgyvendintas pasitelkiant kodo peržiūras [KS18]. Taikant šią praktiką, kuri ne tik padeda dalintis projekto informacija, bet ir gerina projekto kokybę, reikia į kodo peržiūras įtraukti kitų komandų narius. Techninės ar profesinės žinios, kurias organizacijos

darbuotojai turi, ar naujai įgauna konferencijų bei kitų mokymų metu, turi būti pristatomos kitiems darbuotojams [KHD+16]. Tai galėtų būti prezentacija ar straipsnis į organizacijos vidinę sistemą, kuri prieinama visiems darbuotojams.

Be to, organizacijoje turėtų būti matuojama ir stebima kuo daugiau aspektų, nes tik turint konkrečius duomenis galima efektyviai gerinti savo procesą. Gera praktika surinktas metrikas atvaizduoti gerai visiems organizacijoje matomoje arba lengvai randamoje vietoje, taip, kad visi iš to galėtų gauti daugiausiai naudos [SBO18].

Dalinimosi proceso srityje taikomos praktikos:

- Projekto informacijos nuolatinis skleidimas
- Techninių ar kitų profesinių žinių nuolatinis skleidimas
- Skleidžiama informacija turi būti lengvai randama visų
- Rašytinė komunikacija
- Dokumentacija
- Automatinė dokumentacija
- Kodo peržiūros
- Renkamų metrikų nuolatinis skleidimas

### 2.3. Kokybės proceso srities praktikos

Kokybės užtikrinimas – proceso sritis, kurios niekaip negalima praleisti kalbant apie DevOps. Užtikrinti kokybę reiškia, ne tik pasirūpinti, kad programų sistema veiktų, bet, kad ir taip pat būtų saugi nuo potencialių kenkėjiškų atakų [Sto15]. Literatūroje galima rasti nemažai DevOps praktikų skirtų kokybės užtikrinimui.

Kuriant programų sistemą ir esant dažniems pakeitimams negalima neturėti testų [Sto15], testai turi būti automatiniai ir paleidžiami kiekvieną kartą surenkant sistemą [Sto15]. Gera praktika kurti programų sistemą taikant testavimu grįstą kūrimą (angl. Test-Driven Development (TDD)) [JS05]. Visapusiškai kokybei užtikrinti reikalingi modulių testavimas (angl. unit testing) bei integravimo testavimas (angl. integration testing) [WEH15].

Be to, kokybei gerinti naudinga ir statinė kodo analizė, kuri gali būti įgyvendinta atliekant kodo peržiūras, prieš kodui patenkant į gamybinę aplinką [Sto15, RW16]. Kitas būdas atlikti statinę kodo analizę bei kelti kodo kokybę – porinis programavimas (angl. pair programming) [Sto15]. Taikant šias statinės kodo analizės praktikas, dalinamasi žiniomis apie projekto funkcionalumą bei pasikeitimus ir tuo pačiu atkreipiamas kitų programuotojų dėmesys į naują kodą, kas leidžia atrasti vietas, kurias būtų galima optimizuoti bei užtikrinti nuoseklų kodo stilių. Išėties kodą vėliau bus lengviau suprasti ir modifikuoti.

Kuriant programų sistemą, svarbu turėti vienodos konfigūracijos aplinkas tiek kūrimui, tiek testavimui, tiek gamybinei aplinkai. Praktikos siūlo aplinkas turėti aprašytas kodu (angl. infrastructure as a code). Tokie įrankiai kaip Ansible, Chef, Puppet leidžia sukurti identišką aplinką programų sistemai, įvykdydamos kodu aprašytas funkcijas, kurios nusako kaip operacinė sistema turi būti paruošta ir kokios programų sistemos, turi būti sudiegtos [Gee15]. Taikant šią praktiką išvengiama situacijų, kai funkcionalumas veikia kūrimo ir testavimo aplinkoje, tačiau netikėtai neveikia patekęs į gamybinę aplinką. Be to, kad būtų galima palaikyti aukštą programų sistemos kodo kokybę, reikia reguliariai atlikti kodo pertvarkymus (angl. refactoring) [FBB+99], taip neleidžiant kodui pasenti ir jį optimizuojant.

Siekiant užtikrinti programų sistemos kokybišką eksploatavimą, svarbu stebėti su tuo susijusias metrikas. Pasitelkiant tokius įrankius kaip Datadog, Datawatch ar New Relic, galima matyti vienu metu vykstančių (angl. concurrent) naudotojų seansų kiekius, duomenų bazės apkrovas, centrinio procesoriaus apkrovas, vidutinį programų sistemos atsako laiką ir kitas programų sistemos eksploatavimui svarbias metrikas [SBO18]. Turi būti stebimas iškilusių incidentų sistemoje kiekis bei kokį kiekį naudotojų jis paveikė, tam, kad būtų lengviau prioretizuoti incidentų sprendimą ir kuo skubiau išspręsti problemą [KHD+16].

Taip pat, stebėjimas padeda suprasti produkto naudotojų poreikius bei lūkesčius programų sistemai [Lwa17]. Tiek verslo sprendimų priėmimui, tiek kliento geros produkto patirties užtikrinimui, reikia stebėti naudotojo elgseną sistemoje. Tą padaryti galima renkant metrikas apie tai, kur naudotojas lankosi, kokių funkcionalumų naudojami, kiek laiko užtrunka atlikdamas norimą funkciją arba kuriuo paros metu atliekama daugiausiai pirkimų [Cuk13], siekiant aktualiausias funkcijas padaryti lengviau prieinamas ir greičiau įvykdomas.

Kokybės užtikrinimo proceso srityje taikomos praktikos:

- Automatiniai testai
- Testų leidimas kiekvieno programų sistemos surinkimo metu
- Testavimu grįstas kūrimas
- Kodo peržiūros
- Porinis programavimas
- Vienodų konfigūracijų kūrimo, testavimo ir gamybos aplinkų užtikrinimas
- Apsaugos priemonių taikymas nuo potencialių kenkėjiškų atakų
- Kodo pertvarkymai
- Programų sistemos eksploatavimo metrikų stebėjimas
- Incidentų kiekio stebėjimas
- Incidentais paveiktų naudotojų kiekio stebėjimas

- Naudotojo elgsenos sistemoje stebėjimas

## 2.4. Valdymo proceso srities praktikos

Daugelis DevOps procese taikomų praktikų, pavyzdžiui, nuolatinis diegimas, yra pagrįsti Lean principais [NJT16]. Kadangi vienas iš neabejotinų DevOps taikymo tikslų yra greičiau leisti pakeitimus į gamybos aplinką ir būti lankstesniems, natūralu, kad valdymas turėtų būti paremtas judriosiomis metodikomis (angl. Agile). Galimas ne vienas judriųjų metodikų procesas: Scrum, XP ar kiti. Judriųjų metodikų taikymas leidžia reguliariai, nedideliais kiekiais išleisti programinę įrangą ir jos atnaujinimus į gamybinę aplinką [NJT16].

Visgi, kaip teigiama [KHD+16], nėra būtina taikyti kurį nors iš anksčiau išvardintų ar kitų judriosios metodikos procesų pilnai. Galima pasiimti tik tas judriųjų metodikų praktikas, kurios teiktų daugiausiai naudos organizacijos DevOps procese. Viena pagrindinių valdymo proceso srities veiklų – planavimas. [JAP+16] darbe pateikiamas sąryšis tarp DevOps procese taikytinų praktikų ir atitikmenų judriosiose metodikose esančių praktikų susijusių su planavimu. Išskiriamos tokios praktikos:

- Darbas iteracijomis (angl. Sprint)
- Iteracijų planavimas
- Iteracijų rezultatų peržiūra
- Progreso stebėjimas
- Restrospektyvos

Sėkmingam valdymui, organizacijai svarbu turėti reikalavimų sąrašą (angl. backlog) ir esamos iteracijos reikalavimų sąrašą [KHD+16, Vir15]. Trumpos iteracijos (1-3 savaitės) leidžia organizacijai greitai reaguoti į gautą atgalinį ryšį iš savo klientų. Taip pat, projektui svarbu turėti produkto šeimininką (angl. product owner), kuris galėtų priimti atsakomybę ir priimti sprendimus, susijusius su projektu, greitai.

## 2.5. Diegimo proceso srities praktikos

Kita proceso sritis, be kurios DevOps negalėtų egzistuoti – diegimas [NJT16]. Kadangi vienas iš pagrindinių DevOps tikslų yra kuo greitesnis programų sistemų diegimo ciklas, tam, kad spėti reaguoti į nuolatos besikeičiančius naudotojų reikalavimus, akivaizdu, jog yra poreikis gebėti atlikti šį veiksma automatinio būdu, pasitikint procedūra ir nebijant tą daryti dažnai. Organizacijos, taikančios DevOps praktikas, siekia automatizuoti visą programų sistemos diegimo ciklą [Lwa17], į kurį eina programų sistemos pakeitimų stebėjimas, kodo surinkimas, aplinkų paruošimas bei diegimas į atitinkamas aplinkas. Šios praktikos dažnai apjungiamos ir vadinamos nuolatinio diegimu (angl. continuous deployment) bei nuolatinio pristatymu (angl. continuous delivery) [SBO18].

Pasitelkiant automatizaciją, ženkliai mažinamas žmogiškųjų klaidų, kurios yra susijusios su programų sistemos išleidimu į gamybos aplinką, kiekis [Lwa17].

Automatizuojant diegimą, svarbu gebėti tai atlikti išvengiant prastovų, nepamirštant ir ilgai trunkančių duomenų bazių, nuo kurių priklauso programų sistemos, migracijų [Ver16]. Įvykus incidentui susijusiam su diegimu ar naujai įdiegta programų sistemų versija, organizacija turi gebėti atšaukti įdiegtą versiją ir įdiegti ankstesnę korektiškai veikiančią programų sistemų versiją [Ver16].

Be to, diegimo gyvavimo cikle gera praktika yra matuoti bei stebėti laiką nuo kodo pakeitimo iki jo įdiegimo gamybinėje aplinkoje bei kiekvienos operacijos (automatiniai testai, diegimas ir kt.) įvykdymo laiką atskirai [KHD+16]. Taip pat, turėtų būti stebimi nesėkmingi diegimai bei diegimų dažnis.

Diegimo proceso srityje taikomos praktikos:

- Automatizuotas kodo surinkimas
- Automatizuotas programinės įrangos aplinkų kūrimas
- Automatizuotas programų sistemos diegimas
- Nuolatinis diegimas
- Galimybė atstatyti anksčiau įdiegtą versiją
- Programų sistemos diegimas be prastovos
- Duomenų bazių migracijos be prastovos
- Diegimo dažnumo stebėjimas
- Nepavykusių diegimų kiekio stebėjimas
- Diegimo laiko stebėjimas

## 2.6. Rezultatai

Atlikta DevOps procese taikomų praktikų analizė parodė, kad organizacijos iš tiesų vykdo DevOps ir taiko platų rinkinį praktikų, kurios yra svarbios, siekiant pasiūlyti būtent praktiškai taikomą DevOps brandos modelį. Nors šaltiniuose organizacijos ir nekalba brandos modelių kontekste, tačiau praktikos vis tiek yra grupuojamos ir tos grupės atitinka proceso sritis. Bene didžiausia svarba šaltiniuose teikiama kultūrai, kadangi DevOps yra suvokiamas kaip kultūrinis judėjimas, reikalaujantis, būtent, kultūrinio pokyčio organizacijose, kitokio požiūrio į darbą. Taip pat, išryškėjo DevOps ir judriųjų metodikų glaudus ryšys. DevOps skatina bendradarbiavimą, dalinimąsi, siekia, kad naujos programų sistemų versijos galėtų būti išleidžiamos kuo greičiau ir dažniau, išvengiant incidentų gamybos aplinkoje.

### 3. BAZINIO DEVOPS BRANDOS MODELIO PASIRINKIMAS

Darbe buvo pristatyti 7 detaliausi literatūroje randami DevOps brandos modeliai, taip pat apžvelgtos DevOps procese taikomos praktikos. Analizė atlikta, siekiant išsiaiškinti dabartinę situaciją, kokie yra DevOps brandos modeliai, ar jie gali būti taikomi praktiškai, ir kokias praktikas taiko DevOps vykdančios organizacijos. Deja, analizės metu paaiškėjo, kad visi nagrinėti DevOps brandos modeliai nėra pakankamai detalūs, kad juos būtų galima taikyti praktiškai ne tik jų autoriams. Dauguma nagrinėtų DevOps brandos modelių nepateikia naudojamos brandos lygių sampratos. Iš tiesų, samprata buvo pateikta tik 2 modeliuose iš 7 – [Moh15, Min14]. Nors [Min14] pateiktos brandos lygių sampratos detalumas – žemas, [Moh15] DevOps brandos modelyje pateikta samprata yra išsami ir yra pateikta patikslinta visose proceso srityse. [Moh15, Sol16] modeliai naudoja CMMI atitinkančius brandos lygių pavadinimus, o [BK17] modelis – SW-CMM, tad būtų galima manyti, jog ir šių modelių brandos lygių samprata atitinkamai sutampa su CMMI ar SW-CMM, tačiau nei vienas iš autorių to išreikština nepatvirtina, o [Moh15] pateikta brandos lygių samprata, leidžia suprasti, kad ji nesutampa – autorius pateikia savo sampratą. Dalis modelių ([Moh15, MM15]) neturi įvardintų praktikų, kurias reikėtų taikyti kiekvienos proceso srities kiekviename brandos lygyje. Modeliai, kurie turi įvardintas praktikas, apsiriboja praktikų pavadinimais, bet nėra pateikiama jokia papildoma informacija, kaip praktikos turėtų būti įgyvendintos, o tai labai svarbu praktiškai taikomam DevOps brandos modeliui, kadangi remiantis praktikomis vertinama DevOps proceso branda. DevOps brandos modelių proceso sričių apibrėžimai nėra itin svarbi DevOps brandos modelio charakteristika, kadangi tai tiesiogiai nepadeda vertinti organizacijos DevOps brandos, tačiau, tokiais atvejais, kai modelio autorius nepateikia jokių praktikų ([Moh15, MM15]), proceso sričių aprašymai leidžia suprasti, ko modelio autorius tikisi iš kiekvienos proceso srities, kiekviename brandos lygyje. Galimai dėl šios priežasties, [Moh15, MM15] modelių autoriai pateikia detalesnius proceso sričių aprašymus. [Moh15] proceso sričių aprašymai yra išsamūs, o [MM15] – vidutiniškai išsamūs. Kitų modelių autoriai pateikia praktikas, tad ir jų proceso sričių aprašymai arba neišsamūs, arba jų visai nėra. Be to, nei vienas iš analizuotų DevOps brandos modelių nekalba apie vertinimą, kada laikoma, kad proceso sritis tam tikrame brandos lygyje yra įgyvendinta, bei kada brandos lygis yra įgyvendintas.

Kadangi didžioji dalis brandos modelių buvo sudaryti remiantis CMMI brandos modeliu, tai ir jų struktūra yra panaši. Dauguma turi 5 brandos lygius, išsiskiria tik P. Bahrs DevOps brandos modelis [Bah13], kuris turi 4 brandos lygius. Be to, [Moh15, Sol16] modelių brandos lygiai sutampa su CMMI brandos lygiais, o [BK17] su SW-CMM brandos lygiais. Taip pat, dauguma nagrinėtų DevOps brandos modelių, kaip ir CMMI, pirmą brandos lygį suteikia visoms veiklą vykdančioms organizacijoms, lygiui nekeliama reikalavimai. Vis tik, [Min14, Bah13] DevOps brandos modeliai

pirmam brandos lygiui kelia reikalavimus ir jis nėra suteikiamas visiems automatiškai. Tam, kad būtų galima aiškiau matyti, kokie brandos lygių pavadinimai naudojami nagrinėtuose DevOps brandos modeliuose, pateikiama 3 lentelė. 3 lentelėje, [Min14, Bah13] modelių brandos lygiai surašyti praleidžiant 1 brandos lygį, šalia pavadinimo nurodant DevOps modelio brandos lygį, kaip jį nurodė modelio autorius, kadangi modelio 1 brandos lygiui yra keliami reikalavimai, organizacijoms netenkinančioms 1 brandos lygio reikalavimų negalėtų būti priskiriamas joks įvertis, brandos lygis. Norint šį modelį taikyti praktiškai būtina būtų pridėti 0 brandos lygį.

3 lentelė. DevOps modelių brandos lygių pasiskirstymas.

	1 brandos lygis	2 brandos lygis	3 brandos lygis	4 brandos lygis	5 brandos lygis	6 brandos lygis
[Moh15]	Pradinis	Valdomas	Apibrėžtas	Matuojamas	Optimizuotas	
[BK17]	Pradinis	Atkartojamas	Apibrėžtas	Valdomas	Optimizuotas	
[Min14]	-	1. Bazinis	2. Pradedantysis	3. Tarpinis	4. Pažengęs	5. Ekstremalus
[Bah13]	-	1. Praktikuojamas	2. Atkartojamas	3. Patikimas	4. Išplėstas	
[MM15]	Bazinis	Atsirandantis	Suderintas	Pagerintas	Aukščiausias	
[Bea14]	Nepradėtas	Pradedamas	Fundamentalus	Valdomas	Optimizuojamas	
[Sol16]	Pradinis	Valdomas	Apibrėžtas	Kiekybiškai valdomas	Optimizuojamas	

Nagrinėtuose DevOps brandos modeliuose esančios proceso sritys skiriasi ir tik mažuma proceso sričių sutampa skirtinguose modeliuose. Be to, praktikų analizės metu identifikuotos proceso sritys yra randamos tik mažumoje nagrinėtų brandos modelių. Tam, kad būtų galima lengviau matyti, kokios proceso sritys yra analizuotuose DevOps brandos modeliuose, pateikiama proceso sritys apibendrinanti 4 lentelė. Taip pat, lentelėje paryškintai pažymėtos proceso sritys, kurios praktikų analizės metu buvo identifikuotos, kaip proceso sritys taikomos DevOps procese.

4 lentelė. Proceso sritys DevOps brandos modeliuose.

	[Moh15]	[BK17]	[Min14]	[Bah13]	[MM15]	[Bea14]	[Sol16]
Komunikacija	X						
Automatizacija	X						
<b>Valdymas</b>	X						
<b>Kokybė</b>	X						
Technologijos		X					X
Procesas		X			X		X
Žmonės		X			X		X



<b>Kultūra</b>		X					X
Kodo kompiliavimas			X				
<b>Diegimas</b>			X	X		X	
Testavimas			X	X		X	
Raportavimas			X				
Planavimas				X			
Matavimas				X			
Kūrimas				X			
<b>Stebėjimas</b>				X			
Įrankiai					X		
Našumo valdymas						X	
<b>Dalinimasis</b>							

Remiantis 4 lentelės duomenimis, galima išskirti dažniausiai modeliuose pasitaikančias proceso sritis:

- Procesas (3 brandos modeliuose)
- Žmonės (3 brandos modeliuose)
- Diegimas (3 brandos modeliuose)
- Testavimas (3 brandos modeliuose)

Taip pat, 4 lentelės duomenys leidžia pamatyti, kokią didelę aibę skirtingų proceso sričių naudoja nagrinėti DevOps brandos modeliai. Nėra nei vienos proceso srities, kuri pasikartotų daugiau nei pusėje modelių. Iš dalies, tokia situacija susidaro dėl to, kad modeliuose naudojami skirtingi terminai tiems patiems dalykams įvardinti, pavyzdžiui, kokybės proceso sritis ir testavimo proceso sritis. Yra ir daugiau proceso sričių, kurias galima būtų sujungti. Siekiant išsiaiškinti, kurios proceso sritys yra aktualiausios, bus apjungtos panašios tik skirtingais terminais įvardinamos proceso sritys.

Prieš apjungiant panašias proceso sritis, reikia atkreipti dėmesį į tai, jog nagrinėtuose modeliuose yra proceso sričių, kurios apima platesnę aibę praktikų, kurios galėtų būti priskiriamos siauresnės apimties proceso sritims. Tokia proceso sritis yra „proceso“ proceso sritis, kuri yra randama [BK17, Sol16] DevOps brandos modeliuose. Tam, kad galėtume suprasti, kokias proceso sritis apima plati „proceso“ proceso sritis, esanti [BK17, Sol16] modeliuose, reikia peržiūrėti šiuose modeliuose taikytinas praktikas. Argumentacijai naudojamas tik vienas brandos lygis, jeigu kitų lygių praktikos taip pat susijusios, visas praktikų sąrašas yra matomas 2 skyriuje, kur pateikti DevOps

brandos modelių pristatymai. 5 lentelėje nurodytos 2 brandos lygio [BK17, Sol16] DevOps modelių taikytinos praktikos.

5 lentelė. [BK17, Sol16] praktikų palyginimas „proceso“ proceso srityje 2 brandos lygyje.

[Sol16]	[BK17]
<ul style="list-style-type: none"> <li>• Projekto ir reikalavimų valdymas</li> <li>• Suplanuotas pristatymo procesas</li> <li>• Scrum taikymas kūrimo procese</li> <li>• Reikalavimais pagrįstas testavimas</li> <li>• Paprasti scenarijai</li> <li>• Taktinis raportavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Projektų ir reikalavimų valdymas</li> <li>• Suplanuotas pristatymo procesas</li> <li>• Scrum taikymas kūrimo procese</li> <li>• Reikalavimais pagrįstas testavimas</li> <li>• Diegimo ir kūrimo dokumentacija ir susiję konfigūracijos failai nuolatos atnaujinami</li> <li>• Procesai valdomi, bet ne standartizuoti</li> </ul>

Kaip matyti 5 lentelėje, abu modeliai turi praktiką „Scrum taikymas kūrimo procese“, kuri tiktų [Bah13] DevOps brandos modelyje esančiai „Kūrimas / Testavimas“ proceso sričiai, kurią taip pat reiktų skaidyti į gan akivaizdžias dvi proceso sritis. Kita pasikartojanti praktika – „Reikalavimais pagrįstas testavimas“, kuri galėtų būti priskirta konkretesnei proceso sričiai nei „procesas“ – kokybės proceso sričiai. 5 lentelėje, galima matyti ir trečią praktiką kuri pasikartoja 2 brandos lygyje, tarp šių dviejų DevOps brandos modelių – „Suplanuotas pristatymo procesas“. Deja, negalima tiksliai pasakyti, ką tiksliai autoriai turėjo omeny pridėdami tokią praktiką, bet pažiūrėjus į 3 brandos lygį, kuriame jau yra praktika „automatizuotas pristatymas“, galima daryti išvada, jog kalbama apie techninius dalykus ir ši praktika tiktų diegimo proceso sričiai. Paskutinė sutampanti praktika – „Projektų ir reikalavimų valdymas“. Praktika, žinoma, be papildomos informacijos labai abstrakti, tačiau, galima numanyti, jog autoriai kalba apie valdymo proceso sričiai tinkančias praktikas. Papildomai, [Sol16] brandos modelio autorius į „proceso“ proceso sritį įtraukia praktiką – „taktinis raportavimas“. Šiai praktikai, 4 lentelėje, galima rasti sukonkretintą raportavimo proceso sritį.

Taigi, [BK17, Sol16] modelių „proceso“ proceso sritis, gali būti skaidoma į: kūrimo, kokybės, diegimo ir valdymo proceso sritis. O [Sol16] modelio – papildomai dar į raportavimo proceso sritį. Šis išaiškinimas buvo reikalingas, tam, kad vėliau jungiant panašias proceso sritis, nebūtų painu, kodėl „proceso“ proceso sritis jungiama su keliomis skirtingomis proceso sritimis.

Apjungiant panašias proceso sritis, naudojami 2 skyriuje, atliekant praktikų analizę, panaudoti proceso sričių pavadinimai, nebent, apjungiant, būtų identifikuotos naujos proceso sritys, tuomet sprendimas dėl bendro, panašių proceso sričių, pavadinimo priimamas individualiai kiekvienu atveju.

Pateikiamas sąrašas proceso sričių į kurias apjungiamos panašias praktikas turinčios, bet skirtingai autorių pavadintos, proceso sritys:

- *Dalinimosi proceso sritis* – ši proceso sritis buvo identifikuota DevOps procese taikomų praktikų analizės metu. Tarp nagrinėtų DevOps brandos modelių yra tokių, kurių proceso sritis, įvardintas kitokiais terminais, galima apjungti į šią proceso sritį. [Moh15] modelyje esanti komunikacijos proceso sritis neturi jai priskirtų praktikų, tačiau pats proceso srities pavadinimas pasako, jog tai susiję su komunikacija, taip pat šios proceso srities aprašyme kalbama apie skirtingų komandų komunikacija, o dalinimosi proceso sričiai priskirtos praktikos taip pat apima komunikaciją, pavyzdžiui, rašytinė komunikacija arba projekto informacijos nuolatinis skleidimas. Kita proceso sritis, randama [BK17, Sol16] modeliuose, kurią būtų galima prijungti prie dalinimosi proceso srities – žmonių proceso sritis, modelių praktikos šios proceso srities 2 brandos lygyje pateiktos 6 lentelėje, kartu su praktikų analizėje identifikuotomis dalinimosi praktikomis. Kadangi 2 brandos lygyje esančių praktikų pakanka įrodyti šių proceso sričių sąryšį, kitų brandos lygių praktikos į lentelę neįtraukiamos. [Sol16] brandos modelyje šios proceso srities praktikos gali būti siejamos su dalinimusi bei komunikacija, pavyzdžiui, rašytinė komunikacija, komandų organizavimas pagal produkto pristatymus ir pan. [BK17] brandos modely, taip pat yra tokia pati praktika – komandų organizavimas pagal produkto pristatymus. Papildomai [BK17] modelyje pateikiamos komandinio mokymosi bei kompetencijų ugdymo dalyvaujant mokymuose praktikos, kas taip pat yra priskirtina žinių dalinimuisi. Be to, žmonių proceso sritį turi ir [MM15] brandos modelis, tačiau dėl praktikų nebuvimo jį įvertinti sudėtingiau, bet modelio autoriaus pateiktame proceso srities aprašyme, kalbama apie skirtingų komandų bendradarbiavimą ir komunikaciją, todėl galima šio modelio žmonių proceso sritį taip pat jungti su šia, dalinimosi proceso sritimi. Taigi, dalinimosi proceso sričiai priklauso [Moh15] komunikacijos proceso sritis bei [BK17, MM15, Sol16] modeliuose esanti žmonių proceso sritis.

6 lentelė. [BK17, Sol16] praktikų palyginimas žmonių proceso srityje 2 brandos lygyje.

Dalinimosi praktikos	[Sol16]	[BK17]
<ul style="list-style-type: none"> <li>• Projekto informacijos nuolatinis skleidimas</li> </ul>	<ul style="list-style-type: none"> <li>• Komandų organizavimas pagal produkto pristatymus</li> <li>• Rašytinė komunikacija</li> </ul>	<ul style="list-style-type: none"> <li>• Komandų organizavimas pagal produkto pristatymus</li> <li>• Komandinis mokymasis</li> </ul>

<ul style="list-style-type: none"> <li>• Techninių ar kitų profesinių žinių nuolatinis skleidimas</li> <li>• Skleidžiama informacija turi būti lengvai randama visų</li> <li>• Rašytinė komunikacija</li> <li>• Dokumentacija</li> <li>• Kodo peržiūros</li> </ul>	<ul style="list-style-type: none"> <li>• Tinkamo lygio techniniai įgūdžiai</li> <li>• Darbo bilietai</li> </ul>	<ul style="list-style-type: none"> <li>• Kompetencijų ugdymas dalyvaujant mokymuose</li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------

- *Valdymo proceso sritis* – ši proceso sritis buvo identifikuota DevOps procese taikomų praktikų analizės metu. Tarp nagrinėtų DevOps brandos modelių yra tokių, kurių proceso sritis, įvardintas kitokiais terminais, galima apjungti į šią proceso sritį. Kaip buvo išaiškinta anksčiau [BK17, MM15, Sol16] modelių „proceso“ proceso sritis padengia valdymo proceso sritį, todėl šie modeliai priskiriami modeliams turintiems valdymo proceso sritį. Modelyje [Bah13] esanti proceso sritis „Matavimas / Planavimas“, kuri jau iš pavadinimo matosi, kad apima platesnę nei vienos proceso srities praktikų rinkinį, turi valdymo proceso sričiai priskirtinų praktikų, pavyzdžiui, „atkartojamas“ brandos lygyje esanti praktika – „centralizuotas reikalavimų valdymas“. Taip pat, šioje [Bah13] modelio proceso srityje yra praktikų, susijusių su matavimu, kurios galėtų būti grupuojamos su kitomis panašiomis proceso sritimis, arba atskiriamos į matavimo proceso sritį. Dar vienas modelis, turintis proceso sritį, kurią potencialiai galima būtų jungti į bendrą valdymo proceso sritį – [Bea14] ir modelio „našumo valdymas“ proceso sritis. Deja, [Bea14] modelio detalumas labai žemas ir sunku suprasti, apie kokio našumo valdymą modelio autorius kalba, vis tik, kadangi kalbama apie valdymą, ši proceso sritis taip pat prijungiama į šią bendrą proceso sričių grupę. Taigi, valdymo proceso sričiai priskirtos [BK17, MM15, Sol16] modeliuose esančios „proceso“ proceso srities dalis, taip pat [Bah13] modelio planavimo proceso sritis bei [Bea14] modelio našumo valdymo proceso sritis.
- *Kokybės proceso sritis* – ši proceso sritis buvo identifikuota DevOps procese taikomų praktikų analizės metu. Tarp nagrinėtų DevOps brandos modelių yra tokių, kurių proceso sritis, įvardintas kitokiais terminais, galima apjungti į šią proceso sritį. [Min14, Bah13] modeliai turi testavimo proceso sritį, tiksliau, [Bah13] turi „Kūrimas / Testavimas“ proceso sritį, kurioje yra tiek kūrimo proceso srities praktikų, tiek

testavimo proceso srities praktikų, šiuo atveju žiūrime į [Bah13] modelio proceso srities testavimo dalį. 7 lentelėje nurodytos [Min14, Bah13] modelių testavimo praktikos 2 brandos lygyje bei DevOps praktikų analizės metu identifikuotos kokybės užtikrinimo praktikos. Praktikos parodo, kad nors šių proceso sričių pavadinimai kitokie, autoriai turi omeny tuos pačius dalykus. Be to, testavimo proceso sritį turi ir [Bea14] modelis, bet modelyje praktikos nėra išreikštinai susietos su proceso sritimis, tačiau brandos lygių aprašymuose autorius mini automatinio testavimo praktiką, taigi šis modelio proceso sritis taip pat jungiamas į kokybės proceso sritį. Papildomai, kaip jau buvo išaiškinta anksčiau, jungiamos [BK17, Sol16] brandos modelių „proceso“ proceso sričių dalys, kurios padengia kokybės užtikrinimo praktikas. Taigi, kokybės proceso sričiai priskirtos [Min14, Bah13, Bea14] modelių testavimo proceso sritys bei [BK17, Sol16] brandos modelių „proceso“ proceso sritys.

7 lentelė. [Bah13, Min14] praktikos testavimo proceso srityje 2 brandos lygyje.

Kokybės praktikos	[Bah13]	[Min14]
<ul style="list-style-type: none"> <li>• Automatiniai testai</li> <li>• Testų leidimas kiekvieno programų sistemos surinkimo metu</li> <li>• Kodo peržiūros</li> <li>• Porinis programavimas</li> <li>• Vienodų konfigūracijų kūrimo, testavimo ir gamybinės aplinkų užtikrinimas</li> </ul>	<ul style="list-style-type: none"> <li>• Pristatymas ir sistemos surinkimas susietas su testais</li> <li>• Automatinis testavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Svarbiausių testų paleidimas kodo kompiliavimo metu</li> </ul>

- *Diegimo proceso sritis* – nagrinėtuose DevOps brandos modeliuose yra daugiau modelių, kurių proceso sritys reikalauja įgyvendinti su diegimu susijusias praktikas, tačiau yra įvardintos kitokiais pavadinimais. Tokie modeliai yra [BK17, Sol16] ir jų „proceso“ proceso sritis, kuri, kaip tai buvo išaiškinta anksčiau, turi su diegimo proceso sričiai priskirtinų praktikų.
- *Stebėjimo proceso sritis* – ši proceso sritis buvo identifikuota DevOps procese taikomų praktikų analizės metu. [Min14] DevOps brandos modelis, turintis proceso sritį „Raportavimas“, gali būti jungiamas prie šios proceso srities. [Min14] modelis raportavimo proceso srityje reikalauja įgyvendinti tokias praktikas kaip: „naujausi

raportai visada prieinami“, „raportų tendencijų stebėjimas“. Akivaizdu, kad modelio autorius siekia vertinti su stebėjimu susijusias veiklas. Kita proceso sritis „Planavimas / Matavimas“ esanti [Bah13], kaip jau buvo minėta anksčiau, priskiriant šios proceso srities planavimo dalį prie valdymo proceso srities, taip pat matavimo dalį galima priskirti prie stebėjimo proceso srities. Ši proceso sritis turi stebėjimui priskirtiną taikytiną praktiką, pavyzdžiui, „bendras metrikų stebėjimas vienoje vietoje“. Taigi, į stebėjimo proceso sritį priskiriamos [Min14] raportavimo ir [Bah13] „Planavimas / Matavimas“ proceso sritys.

- *Kūrimo proceso sritis* – kaip jau buvo išaiškinta anksčiau, [BK17, Sol16] brandos modeliuose esanti „proceso“ proceso sritis apima ir kūrimo proceso sritį, tad šie DevOps brandos modeliai prijungiami prie kūrimo proceso srities.

Taigi, identifikavus nagrinėtuose DevOps brandos modeliuose esančias proceso sritis, kurios yra įvardintos skirtingais terminais, galima lengviau atskirti svarbiausias, dažniausiai pasitaikančias proceso sritis. Pateikiama 8 lentelė, tam, kad būtų galima lengviau matyti proceso sričių pasiskirstymą po sugrupavimo. Taip pat, lentelėje paryškintai pažymėtos proceso sritys, kurios literatūros analizės metu buvo identifikuotos, kaip proceso sritys, taikomos DevOps procese.

8 lentelė. Sugrupuotos proceso sritys DevOps brandos modeliuose.

	[Moh15]	[BK17]	[Min14]	[Bah13]	[MM15]	[Bea14]	[Sol16]
<b>Dalinimasis</b>	X	X			X		X
<b>Automatizacija</b>	X						
<b>Valdymas</b>	X	X		X	X	X	X
<b>Kokybė</b>	X	X	X	X	X	X	X
Technologijos		X					X
<b>Kultūra</b>		X					X
Kodo kompiliavimas			X				
Diegimas		X	X	X		X	X
<b>Stebėjimas</b>			X	X			
Kūrimas		X		X			X

Sugrupavus skirtingais terminais įvardintas proceso sritis, 8 lentelėje matosi dažniausiai pasikartojančios (4 ar daugiau kartų) proceso sritys:

- Kokybė (7 brandos modeliai)
- Valdymas (6 brandos modeliai)
- Diegimas (5 brandos modeliai)

- Dalinimasis (4 brandos modeliai)

Taigi, atlikus analizės metu nagrinėtų DevOps brandos modelių palyginimą, su tikslu, nustatyti, ar yra toks nagrinėtas DevOps brandos modelis, kurį būtų galima tikslinti iki praktiškai naudojamo DevOps brandos modelio, paaiškėjo, kad tokio vieno brandos modelio, kurį būtų galima imti kaip bazinį ir jį pildyti nėra. Kaip matyti iš 2 lentelės duomenų, visų nagrinėtų modelių detalumas yra ganėtinai panašus ir, deja, jis yra žemas – DevOps brandos modeliams trūksta esminių elementų (sampratos, praktikų, vertinimo procedūros). Dalies nagrinėtų DevOps brandos modelių struktūra neatitinka klasikinio modelio struktūros (CMMI), pavyzdžiui, kaip matyti 3 lentelėje, [Bah13] turi 4 brandos lygius vietoj standartinių 5. Be to, modeliai [Bah13, Min14] kelia reikalavimus jau pirmajam brandos lygiui, o tai iš esmės neleidžia šių modelių taikyti praktiškai, kadangi naudojant DevOps brandos modelį praktiškai, bet kurios organizacijos vertinimo rezultatas turi būti brandos lygis, o šiuo atveju gaunasi, kad kai kurioms organizacijoms pagal modelį būtų negalima suteikti jokio brandos lygio, nes nėra tenkinami žemiausio lygio reikalavimai. Kita priežastis, kodėl negalima pasirinkti bazinio DevOps brandos modelio – proceso sričių pasiskirstymas. Proceso sričių pasiskirstymas tarp brandos modelių, kaip pateikta 4 lentelėje, parodo, kad nėra modelio, kuris apimtų visas reikiamas proceso sritis: vieni modeliai atkreipia dėmesį į organizacinius dalykus, kaip valdymas, komunikacija, pavyzdžiui, [Moh15], kiti – labiau technines DevOps proceso sritis, kaip diegimas, testavimas, pavyzdžiui, [Bah13, Min14] ir pan. Pasirinkus tik vieną modelį kaip bazinį, tektų iš esmės keisti to modelio proceso sričių sąrašą, pagal 8 lentelę. Tad, remiantis šiais faktais, daroma išvada, kad būtų netikslinga pasirinkti vieną, iš esamų, brandos modelį ir jį, naudojant kaip bazinį, plėsti, kadangi galutiniame rezultate gautume naują brandos modelį, kuris neturėtų daug bendra su pradžioje pasirinktu brandos modeliu. Tokiu atveju, nuspręsta, jog tikslingiau kurti naują praktiškai taikomą DevOps brandos modelį, kitų modelių bei praktikų analizės pagrindu.

## 4. NAUJO DEVOPS BRANDOS MODELIO ELEMENTAI

Siekiant pasiūlyti naują praktiškai taikomą DevOps brandos modelį, reikia apsispręsti dėl būsimo modelio architektūros bei kitų esminių modelio elementų, tokių kaip: brandos lygiai, proceso sritys, praktikos. Šiame skyriuje priimami principiniai sprendimai dėl naujai kuriamo DevOps brandos modelio.

### 4.1. Architektūra

Literatūros analizėje nagrinėti modeliai išreikštinai nenurodo, kokios jie yra architektūros, tačiau, kadangi jie yra vadinami brandos modeliais ir turi brandos lygius bei dauguma jų yra sudaryti remiantis CMMI, galima daryti išvadą, kad jie yra pakopinės (angl. staged) architektūros. Ši architektūra duoda iš karto apibrėžtą gerinimo kelią, o tai organizacijoms duoda aiškumą ir leidžia efektyviai, sistemingai kelti savo brandą [Cep05].

Kuriant pakopinės architektūros brandos modelį, reikia nuspręsti kaip proceso sritys turėtų būti siejamos su brandos lygiais. Tą padaryti galima keliais skirtingais būdais. Pirmas variantas būtų individualiai priskirti kiekvieną proceso sritį konkrečiam brandos lygiui, kaip tai yra daroma, pavyzdžiui, CMMI-DEV brandos modelyje [CMD10]. Kitas variantas, kaip tai daro nagrinėtų DevOps brandos modelių autoriai – visos proceso sritys yra priskiriamos visiems modelyje esantiems brandos lygiams, kas reiškia, kad norint pasiekti, pavyzdžiui, 5 brandos lygį, reikia įgyvendinti visų brandos modelyje esančių proceso sričių praktikas, susietas su visais brandos lygiais. Autoriai nepateikia paaiškinimo, kodėl pasirinktas toks variantas, tačiau reikia atkreipti dėmesį, jog DevOps brandos modelių taikymo sritis yra siauresnė nei CMMI-DEV, todėl ir pasirinkimas, kaip sieti proceso sritis su brandos lygiais, gali būti kitoks. Galima pastebėti, kad nagrinėtų DevOps brandos modelių proceso sričių kiekis yra gerokai mažesnis negu, pavyzdžiui, CMMI-DEV [CMD10], kur proceso sritys individualiai yra priskiriamos brandos lygiams. CMMI-DEV turi 22 proceso sritis, kai dauguma nagrinėtų modelių turi tik 4 proceso sritis (5 iš 7 modelių). Likę du modeliai: [Bah13] turi 6 proceso sritis, o [Bea14] – 3.

Nagrinėtų DevOps brandos modelių autoriai priskiria visas proceso sritis visiems brandos lygiams, tačiau nėra aišku ar toks sprendimas buvo priimtas individualiai kiekvienai proceso sričiai, ar nuspręsta, kad visos proceso sritys turi būti priskirtos visiems brandos lygiams. Siekiant, kad naujai kuriamas DevOps brandos modelis būtų lankstesnis, naujai kuriamame DevOps brandos modelyje sprendimas dėl proceso srities priskyrimo brandos lygiui bus atliekamas individualiai, kiekvienos proceso srities kontekste. Tokiu atveju, jeigu yra proceso sričių priskirtų, pavyzdžiui, tik pirmiems 3 brandos lygiams, tam, kad būtų galima pasiekti 4 brandos lygį, tų proceso sričių, kurios būtų priskirtos pirmiems 3 brandos lygiams, gerinti nebereikėtų, keliant brandą iš 3 į 4 brandos lygį. Taikant tokį



variantą, galima sukurti praktiškesnį DevOps brandos modelį, išvengiant papildomų reikalavimų proceso sritims, kai to nereikia.

Proceso sritis gali būti priskiriama keliems brandos lygiams, bet lygiai turi būti iš eilės vienas po kito ir prasidėti nuo 1 brandos lygio. Pavyzdžiui, proceso sritis gali būti priskiriama pirmiems 2, 3, 4 ar 5 brandos lygiams, bet negali būti priskirta 1, 2 ir 4, bet ne 3 brandos lygiams. Proceso srityse, kiekvienam priskirtam brandos lygiui, išskyrus 1 brandos lygį, pateikiamos papildomos praktikos ir proceso sritis yra vertinama kiekviename brandos lygyje, kuriam ji yra priskirta.

Taigi, priimtas sprendimas, kurti pakopinės architektūros brandos modelį, kuriame kiekviena proceso sritis bus iš anksto priskiriama brandos lygiams, pavyzdžiui, pirmiems 3 arba 5, su skirtingų praktikų rinkiniu kiekvienam brandos lygiui, išskyrus 1 brandos lygį, kuriame nebus priskirtų praktikų.

## 4.2. Brandos lygiai

Kuriant naują DevOps brandos modelį reikia apsispręsti dėl modelio brandos lygių ir jų sampratos. Dauguma analizės metu nagrinėtų DevOps brandos modelių turi 5 brandos lygius (žiūrėti 3 lentelę). Išimtis galioja tik [Bah13] DevOps brandos modeliui, kuriame yra 4 brandos lygiai. CMMI-DEV brandos modelyje [CMD10] taip pat yra 5 brandos lygiai. Kadangi, dauguma darbe nagrinėtų modelių buvo sudaryti CMMI pagrindu, tai ir jų struktūra yra panaši.

[Moh15, Sol16] DevOps brandos modeliuose esantys brandos lygių pavadinimai atitinka CMMI brandos lygių pavadinimus, o [BK17] modelio brandos lygių pavadinimai atitinka SW-CMM brandos lygių pavadinimus (žiūrėti 2 lentelę). Tačiau, nei vieno iš šių modelių autorius išreikštinais nenurodė, jog ir tų brandos lygių samprata sutampa atitinkamai su CMMI ar SW-CMM. [Moh15] modelio autorius pateikia savo brandos lygių sampratą, tad akivaizdu, kad ji nėra tokia pati kaip CMMI. Kitų autorių brandos lygiai neatitinka jokių literatūroje plačiai žinomų modelių brandos lygių bei jų samprata nėra pateikiama arba pateikiama neišsamiai. Tikėtina, kad likę autoriai patys sugalvojo savo kuriamo DevOps modelio brandos lygius, nesiremami jokiais kitais modeliais.

Kaip matyti 2 lentelėje, brandos lygių sampratas pateikė tik [Moh15, Min14] DevOps brandos modelių autoriai. [Min14] modelio brandos lygių samprata yra neišsami, tačiau [Moh15] modelio autorius pateikia patikslintą brandos lygių sampratą kiekvienai, modelyje esančiai, proceso sričiai, kurios yra išsamios ir iš jų galima nesunkiai atstatyti bendra modelio brandos lygių sampratą.

Taip pat, 2 nagrinėti DevOps brandos modeliai ([Bah13, Min14]) kelia reikalavimus jau pirmajame savo brandos lygyje, t.y. 1 brandos lygis nėra suteikiamas automatiškai visoms veiklą vykdančioms organizacijoms. Likę 5 nagrinėti DevOps brandos modeliai pirmą lygį suteikia visiems ir jam nėra keliami reikalavimai. Pirmo lygio reikalavimų nebuvimas yra priimtinesnis variantas, tam,

kad būtų galima duoti įvertį minimalių reikalavimų neįvykdančioms organizacijoms – taip elgiamasi ir CMMI.

Naujai kuriamas DevOps brandos modelis turi turėti brandos lygius su apibrėžta jų samprata. Kadangi kuriamas naujas DevOps brandos modelis, brandos lygiai ir jų samprata gali būti pasiūlyti nauji, arba galima remtis nagrinėtais DevOps brandos modeliais. Taip pat, galima būti imti brandos lygius ir jų sampratą iš kitų autoritetingų brandos modelių, pavyzdžiui, CMMI.

Pasiūlyti naują brandos lygių sampratą būtų sudėtingiausia ir tai ženkliai padidintų šio darbo apimtį, todėl pirmiausia svarstomi kiti variantai, siekiant pasinaudoti kitomis, jau esančiomis, brandos lygių sampratomis.

CMMI esanti brandos lygių samprata yra detaliausia, tačiau, naudojant šio brandos modelio sampratą, kuriamas modelis būtų pernelyg sudėtinamas ir būtų neįgyvendintas darbo tikslas pasiūlyti praktiškai taikomą DevOps brandos modelį. CMMI brandos modelio taikymo sritis yra kita, todėl tektų DevOps dirbtinai sprauti į modelio sampratos diktuojamus rėmus, dėl ko modelyje atsirastų pertekliniai elementai, pavyzdžiui, bendriniai tikslai, iš kurių pirmasis tikslas yra įgyvendinti specifinius tikslus. Toks modelio elementas – specifinis tikslas – nėra būtinas naujai kuriamam DevOps brandos modeliui ir tik įveda papildomą sudėtingumą.

Iš nagrinėtų DevOps brandos modelių tik 2 ([Moh15, Min14]) brandos modeliai turi brandos lygių sampratą, tačiau [Min14] samprata nėra priimtina dėl to, kad yra keliami reikalavimai pirmam brandos lygiui bei sampratos detalumas yra žemas. Vis tik, [Moh15] DevOps brandos modelyje esanti samprata yra pakankamai išsami ir lengvai suprantama, kas yra svarbu praktiškai taikomam DevOps brandos modeliui, taip pat šio modelio brandos lygių samprata yra tos pačios taikymo srities (DevOps), todėl tai yra tinkamas kandidatas, iš kurio galima paimti brandos lygius ir jų sampratą, naujai kuriamam DevOps brandos modeliui.

Taigi, naujai kuriamo DevOps brandos modelio brandos lygiai ir jų samprata bus imama iš [Moh15] DevOps brandos modelio.

Naujo DevOps modelio brandos lygiai:

- Pradinis
- Valdomas
- Apibrėžtas
- Matuojamas
- Optimizuotas

Brandos lygių pavadinimai sutampa su CMMI, tačiau brandos lygių pavadinimai imami iš [Moh15] DevOps brandos modelio. [Moh15] brandos modelio autorius kurdamas DevOps brandos

modelį rėmėsi CMMI, tikėtina dėl to ir pavadinimai sutampa, tačiau, pateikta brandos lygių samprata rodo, kad brandos lygiai neturi daug bendra su CMMI.

### 4.3. Proceso sritys

Kitame etape, priimant principinius sprendimus dėl naujai kuriamo DevOps brandos modelio, reikia nuspręsti, kokios proceso sritys turėtų būti modelyje.

Tikrinant, ar galima pasirinkti bazinį brandos modelį, buvo identifikuotos proceso sritys, kurios analizės metu nagrinėtuose DevOps brandos modeliuose pasikartoja dažniausiai, tačiau dėl bendrų terminų nebuvo rezultatai nebuvo prasmingi. Vėliau sujungus panašias proceso sritis, įvardintas skirtingais pavadinimais, buvo išskirtos 4 dažniausiai pasikartojančios proceso sritys, kurios taip pat sutampo ir su proceso sritimis, kurios buvo išskirtos, kaip galimos, DevOps procese taikomų praktikų analizės metu. Tačiau viena proceso sritis (kultūros), iš identifikuotų DevOps praktikų analizės metu, nepateko į dažniausiai DevOps brandos modeliuose naudojamų proceso sričių sąrašą. Dėl to, kyla klausimas ar yra prasminga įtraukti kultūros proceso sritį į naujai kuriamą DevOps brandos modelį?

Iš 7 nagrinėtų tik 2 DevOps brandos modeliai turi kultūros proceso sritį. Visgi, galima pastebėti, kad modeliai, turintys kultūros proceso sritį ([BK17, Sol16]), yra pristatyti vėliau nei visi likę. Taip pat, analizuojant praktikas ir nagrinėjant, kaip taikomas DevOps procesas organizacijose, aiškiai buvo matoma kultūros svarba taikant DevOps. Kadangi, DevOps samprata dar yra palyginti nauja ir atsirado tik 2009 metais, gali būti, kad kultūros svarba iš pradžių nebuvo pakankamai vertinama ir tik vėliau įgavo vertę. Iš nagrinėtų DevOps brandos modelių, vėliausiai pristatyti modeliai tą patvirtina.

Taigi, kuriant naują praktiškai taikomą DevOps brandos modelį, nuspręsta naudoti 8 lentelėje identifikuotas dažniausiai pasikartojančias proceso sritis bei papildomai pridėti kultūros proceso sritį, kurios svarba buvo identifikuota atliekant DevOps procese taikomų praktikų analizę, nesusijusią su brandos modeliais.

Kokybės, valdymo ir diegimo proceso sritys yra priskiriamos visiems 5 brandos lygiams, tačiau, dalinimosi bei kultūros proceso sritys yra kitokios. Pastarosios proceso sritys užtikrina socialinį DevOps aspektą ir matavimas bei optimizavimas šių proceso sričių nėra reikalingas, tą patvirtina DevOps procese taikomų praktikų analizė, kurios metu nebuvo nustatyta praktikų, kurios padėtų matuoti bei optimizuoti organizacijos kultūrą ar dalinimąsi. Todėl, šios dvi proceso sritys yra priskiriamos pirmiems 3 brandos lygiams.

Galutinis kuriamo modelio proceso sričių sąrašas atrodo taip:

- Kokybė (pirmiems 5 brandos lygiams)
- Valdymas (pirmiems 5 brandos lygiams)
- Diegimas (pirmiems 5 brandos lygiams)
- Dalinimasis (pirmiems 3 brandos lygiams)

- Kultūra (pirmiems 3 brandos lygiams)

#### 4.4. Praktikos

Dauguma nagrinėtų DevOps brandos modelių pateikia praktikas, kurios yra susietos su proceso sritimis ir brandos lygiais – tik [Moh15, MM15] modeliai nepateikia jokių praktikų bei [Bea14] modelio praktikos nėra išreikštinai susiejamos su proceso sritimis. Taip pat, atlikus modelių analizę, buvo identifikuotas trūkumas, kad pateiktos praktikos yra neišsamios, t.y. nagrinėtų DevOps brandos modelių autoriai pateikė tik taikytinų praktikų pavadinimus, be jokio papildomo paaiškinimo, kuris leistų teisingai interpretuoti minimą praktiką. Tam, kad naujai kuriamame DevOps brandos modelyje būtų išvengta to paties trūkumo, praktikos modelyje turi būti pateiktos su paaiškinimais, kurie vienareikšmiškai nusakytų kokia tai praktika, kas turi būti atliekama – vien praktikų pavadinimų neužtenka.

Praktikos bus imamos iš praktikų analizėje surinktų praktikų sąrašo bei nagrinėtų DevOps brandos modelių praktikų. Praktikos turi būti priskiriamos proceso sritims bei atitinkamiems brandos lygiams, remiantis brandos lygio samprata. Praktikų detalizavimas, papildomas paaiškinimas, tam, kad būtų aišku, ką ta praktika reiškia, turi būti paremtas praktikos analize arba, jeigu reikia, papildomais šaltiniais – tai gali būti reikalinga, kadangi praktikos, kurios bus imamos iš nagrinėtų DevOps brandos modelių, neturi jokių paaiškinimų.

Taigi, praktikos su paaiškinimais bus priskiriamos visoms proceso sritims, kiekviename brandos lygyje, jeigu brandos lygis yra susietas su proceso sritimi, išskyrus pradinį, kuriam nėra keliami reikalavimai.

#### 4.5. Vertinimas

Vienas iš identifikuotų trūkumų nagrinėtuose DevOps brandos modeliuose yra vertinimo procedūros nebūvimas, t.y. neapibrėžta, kada laikoma, jog proceso sritis konkrečiame brandos lygyje yra įgyvendinta, ir kada laikoma, jog brandos lygis yra pasiektas. Siekiant ištaisyti šį nagrinėtų modelių trūkumą, reikia apsispręsti ir apibrėžti šiuos dalykus.

Kadangi negalima remtis nei vienu iš nagrinėtų DevOps brandos modelių, priimant sprendimą dėl vertinimo procedūros, remiamasi kitu, autoritetingu brandos modeliu – CMMI. Kuriamo DevOps brandos modelio pagrindinis vertinimo elementas – praktikos. CMMI turi vertinimo skalę, kuri padeda įvertinti kaip gerai įgyvendinta praktika [CMA14]. Praktika vertinama vienu iš penkių galimų įvertinimų: „Dar nepradėta“ (angl. Not Yet – NY), „Neįgyvendinta“ (angl. Not Implemented – NI), „Dalinai įgyvendinta“ (angl. Partially Implemented – PI), „Didžiąja dalimi įgyvendinta“ (angl. Largely Implemented – LI) ir „Pilnai įgyvendinta“ (angl. Fully Implemented – FI). Kiekvienas iš

įvertinimų turi aprašytus kriterijus, pavyzdžiui, artefaktų ar patvirtinimų, įrodančių praktikos įgyvendinimą, buvimas.

Naujai kuriamam DevOps brandos modeliui bus naudojama SCAMPI [CMA14] vertinimo skalė, tačiau su tam tikrais pakeitimais. Į kuriamo DevOps brandos modelio praktikų vertinimo skalę, neįtraukiamas įvertis „Dar nepradėta“. Šis įvertis skirtas praktikoms, kurių įgyvendinimas dar nebuvo pradėtas dėl sudarytos darbotvarkės ir yra numatomas ateityje. Siekiant, kad vertinimas naujai kuriamame DevOps brandos modelyje būtų kuo aiškesnis ir paprastesnis, praktikoms, kurios atitinka „Dar nepradėta“ įverčiu vertinamas praktikas, bus skiriamas įvertis – „Neįgyvendinta“. „Neįgyvendinta“ įverčio kriterijai tam neprieštarauja.

9 lentelėje yra pateikiama naujo DevOps brandos modelio praktikų vertinimo skalė, kuri yra sudaryta pagal SCAMPI [CMA14] praktikų vertinimo skalę, su anksčiau įvardintu pakeitimu. Pirmame stulpelyje pateikiamas pilnas įvertinimo pavadinimas, o skliausteliuose trumpinys (trumpiniai paliekami originalo kalba, dėl aiškumo ir unikalumo), antrame – įvertinimui gauti keliami kriterijai. Kriterijų skiltyje naudojami terminai – artefaktai ir patvirtinimai. Artefaktas yra praktikos įgyvendinimo rezultatas, kurį objektyviai galima vertinti, o patvirtinimas yra , pavyzdžiui, rašytinis ar žodinis patvirtinimas dėl vykdomos praktikos.

9 lentelė. Naudo DevOps brandos modelio praktikų vertinimo skalė.

Įvertinimas	Kriterijai
Pilnai įgyvendinta (FI)	Yra pakankamas kiekis artefaktų ir (arba) patvirtinimų, kurie yra vertinami kaip pakankami įrodyti praktikos įgyvendinimą, ir trūkumų nenustatyta.
Didžiąja dalimi įgyvendinta (LI)	Yra pakankamas kiekis artefaktų ir (arba) patvirtinimų, kurie yra vertinami kaip pakankami įrodyti praktikos įgyvendinimą, ir nustatytas vienas arba keli trūkumai.
Dalintai įgyvendinta (PI)	Kai kurių arba visų reikalingų duomenų nėra arba jie laikomi nepakankamais, kai kurie duomenys rodo, kad kai kurie praktikos aspektai yra įgyvendinti, ir nustatytas vienas ar keli trūkumai. ARBA Komandai pateikti duomenys (artefaktai ir (arba) patvirtinimai) konfliktuoja - kai kurie duomenys rodo, kad praktika įgyvendinta, o kai kurie duomenys rodo, kad praktika nėra įgyvendinta, ir nustatytas vienas ar keli trūkumai.
Neįgyvendinta (NI)	Kai kurių ar visų reikalingų duomenų nėra arba jie laikomi nepakankamais, pateikti duomenys nepatvirtina išvados, kad praktika įgyvendinta, ir nustatytas vienas ar keli trūkumai.

Vertinimas atliekamas su smulkiausiu modelio elementu – praktika. Vertinant praktiką, jai yra priskiriamas 1 iš 4 galimų įverčių. Tam, kad praktika būtų laikoma įgyvendinta, jos įvertis privalo būti „LI“ arba „FI“. Proceso sritis, kuriai priskirtas tam tikras brandos lygis, gali būti įgyvendinta

arba neįgyvendinta. Tam, kad proceso sritis būtų įgyvendinta, reikia, kad šios trys sąlygos būtų tenkinamos:

- Visos praktikos priskirtos proceso sričiai, tame brandos lygyje būtų įvertintos „LI“ arba „FI“.
- Identifikuotų trūkumų visuma negali turėti reikšmingos neigiamos įtakos proceso srities įgyvendinimui.
- Jeigu vertinamai proceso sričiai priskirtas aukštesnis nei 2 brandos lygis, žemesni brandos lygiai turi būti įgyvendinti.

Jeigu nors viena praktika įvertinama „LI“, o visos kitos „FI“ t.y. nustatoma bent vienas trūkumas, tokiu atveju vertintojas sprendžia ar proceso sritis yra įgyvendinta. Jeigu vertintojas nusprendžia, kad proceso sritis nėra įgyvendinta, jis turi pateikti paaiškinimą, kodėl nustatytas trūkumas ar trūkumai turi reikšmingos neigiamos įtakos proceso srities įgyvendinimui.

Tam, kad įgyvendinti brandos lygį, reikia, kad visos proceso sritys, kurios yra priskirtos tam brandos lygiui, būtų įgyvendintos tame brandos lygyje.

## 5. NAUJAS DEVOPS BRANDOS MODELIS

Šiame skyriuje pateikiamas naujas, praktiškai taikomas DevOps brandos modelis, kuris yra sudarytas pagal anksčiau darbe priimtus sprendimus, apimančius architektūrą, brandos lygius, proceso sritis, praktikas bei vertinimą.

### 5.1. Brandos lygių samprata

Brandos lygių samprata naujam DevOps brandos modeliui sudaryta atstatant [Moh15] DevOps modelio brandos lygių sampratą, kuri modelio autoriaus yra pateikta tik kaip patikslinta kiekvienai proceso sričiai atskirai. Pateikiama pilna naujame DevOps brandos modelyje naudojama brandos lygių samprata:

- *1 brandos lygis: Pradinis* – organizacijos, kuri yra pradiniam brandos lygyje, proceso sritys būna įgyvendinamos reaktyviai ir chaotiškai. Sėkmė tokiose organizacijose priklauso nuo konkrečių žmonių kompetencijų ir jų herojiškumo, bet ne nuo patikrintų procedūrų vykdymo. Taip pat, 1 brandos lygio organizacijos dažnai pasižymi tendencija nutraukti savo vykdomas iš anksto apibrėžtas procedūras išitikus krizei bei nesugeba pakartoti jau anksčiau buvusios sėkmės. Šiam brandos lygiui gali būti priskiriamos visos veiklą vykdančios organizacijos, nes reikalavimai nėra keliami.
- *2 brandos lygis: Valdomas* – 2 brandos lygyje proceso sritys vykdomos koordinuotai, laikantis iš anksto nustatytų taisyklių, tačiau procedūros ir jas apibrėžiančios taisyklės vis dar nėra tapę organizacijos standartais ir atskiruose organizacijos projektuose – gali būti vykdomos skirtingai. Vykdamas organizacijos projektus, į juos įtraukiamos reikiamos suinteresuotos šalys. Vykdomų procedūrų drausmė, kuri yra atspindima 2 brandos lygio, padeda užtikrinti, kad išitikusios krizės metu būtų nepertraukiamas jų vykdymas.
- *3 brandos lygis: Apibrėžtas* – 3 brandos lygyje organizacijoje vykdomos procedūros bei naudojami įrankiai yra aiškiai apibrėžti ir aprašyti organizacijos standartuose. Šie organizacijos standartai yra naudojami užtikrinti procedūrų vykdymo nuoseklumui organizacijoje tarp skirtingų komandų bei projektų. Kiekvieno projekto komanda turi galimybę nežymiai pakoreguoti procesą, esant specialioms poreikiams, tačiau tą darant, procesas vis tiek turi išlikti suderintas su organizacijoje priimtais standartais.
- *4 brandos lygis: Matuojamas* – 4 brandos lygyje organizacija turi nusistačiusi kiekybinius proceso sričių tikslus ir naudoja juos kaip kriterijus valdydama projektus. Renkamos metrikos yra svarbios tam, kad būtų galima matyti vykdomų procesų sričių kuriamą vertę, o kiekybiniai tikslai grindžiami kliento, galutinių vartotojų, organizacijos ir proceso sričių vykdytojų poreikiais.

- *5 brandos lygis: Optimizuotas* – 5 brandos lygyje organizacija rūpinasi bendru organizacijos rezultatyvumu, nuolat tobulina savo proceso sričių vykdymą, remdamasi kiekybiniu savo verslo tikslų ir veiklos rezultatų, surinktų iš skirtingų projektų, supratimu. Duomenų analizė nustato trūkumus ar spragas. Šios spragos naudojamos organizaciniam procesų optimizavimui skatinti, o tai lemia pastebimą veiklos tobulėjimą.

## 5.2. Proceso sritys

Šio skyriaus poskyriai suvokiami kaip kuriamo DevOps brandos modelio proceso sritys. Poskyriuose, pateikiama patikslinta brandos lygių samprata, kad būtų aiškiau, kaip brandos lygių samprata suprantama konkrečioje proceso srityje. Taip pat, pateikiamos taikytinos praktikos kiekviename brandos lygyje.

### 5.2.1. Kokybė

Šiame poskyryje pateikiama kokybės proceso srities brandos lygių samprata bei taikytinos praktikos kiekviename brandos lygyje. Kokybės proceso sritis yra priskiriama visiems 5 brandos lygiams.

Tikslas – užtikrinti organizacijos kuriamų programų sistemų kokybę.

#### 5.2.1.1. Brandos lygių samprata

Šiame poskyryje pateikta samprata yra patikslina taip, kaip ji turėtų būti suprantama kokybės proceso srityje:

- *Pradinis: 1 brandos lygis* – organizacijos, esančios šiame brandos lygyje, kokybės užtikrinimas paprastai būna įgyvendinamas reaktyviai ir chaotiškai. Kokybės užtikrinimas priklauso nuo konkrečių žmonių organizacijoje kompetencijų ir jų herojiškumo, bet ne nuo patikrintų procedūrų vykdymo. Organizacijoje nėra jokių kokybės standartų. Į gamybinę aplinką dažnai patenka netestuotas, klaidų turintis programinis kodas.
- *Valdomas: 2 brandos lygis* – 2 brandos lygyje kokybės užtikrinimas vykdomas laikantis iš anksto nustatytų taisyklių, taikomos gerosios praktikos, tačiau tai vis dar nėra standartizuota ir atskiruose projektuose gali būti vykdoma skirtingai. Vykdomuose organizacijos projektuose, atliekamas pagrindinių programų sistemų funkcionalumo testavimas.
- *Apibrėžtas: 3 brandos lygis* – 3 brandos lygyje organizacijos vykdomas kokybės užtikrinimas yra gerai apibrėžtas ir suprantamas bei aprašomas standartuose. Kokybės užtikrinimas, skirtinguose projektuose, vykdomas remiantis tais pačiais organizacijos



mastu priimtais standartais. Standartai yra pritaikomi kiekvienam organizacijoje vykdomam projektui individualiai, jeigu to reikia. Kokybės užtikrinimas atliekamas automatinėmis priemonėmis, rankinis testavimas vykdomas tik ten, kur to padaryti automatinėmis priemonėmis neįmanoma.

- *Matuojamas: 4 brandos lygis* – 4 brandos lygyje organizacija turi nusistačiusi kiekybinius kokybės užtikrinimo proceso srities tikslus ir naudoja juos kaip kriterijus užtikrinant kokybę. Kokybės užtikrinimo kiekybiniai tikslai grindžiami kliento, galutinių vartotojų ir organizacijos poreikiais.
- *Optimizuotas: 5 brandos lygis* – 5 brandos lygyje organizacija orientuojasi į nuolatinį kokybės užtikrinimo efektyvumo gerinimą, analizuojami turimi kiekybiniai duomenys, kuriais remiantis atliekamas laipsniškas procedūrų tobulinimas.

### 5.2.1.2. Praktikos

Šiame poskyryje pateikiamos praktikos, kurios yra priskirtos DevOps brandos modelyje esančiai kokybės proceso sričiai, kiekviename brandos lygyje bei tų praktikų detalizavimas:

- *Valdomas: 2 brandos lygis*
  - *Kodo versijų kontroliavimas* – programų sistemų kodo versijos organizacijoje yra valdomos naudojantis kodo versijavimo įrankiais, pavyzdžiui, git. Atlikus kodo pakeitimus, naudojantis versijavimo įrankiu, kodas įkeliamas į organizacijos kodo saugyklą (angl. repository), kodo versijos pakeitimas susiejamas su pakeitimą atlikusiu darbuotoju.
  - *Kodo prieigos kontroliavimas* – organizacijos programų sistemų kodas yra prieinamas tik tiems darbuotojams, kurie turi tam leidimą, pavyzdžiui, programuotojai. Darbuotojai, kuriems nėra poreikio peržiūrėti programų sistemos kodą neturi galėti prieiti prie kodo.
  - *Pagrindinio programų sistemos funkcijų identifikavimas* – identifikuojamas pagrindinis organizacijos kuriamų ir prižiūrimų programų sistemų funkcionalumas. Tai yra kritinis funkcionalumas, be kurio programų sistema netenka prasmės. Pagrindinio funkcionalumo identifikavimas reikalingas tam, kad būtų galima užtikrinti šio funkcionalumo nepertraukiamą veikimą.
  - *Pagrindinių modulių testavimas* – naudojamas automatinis modulių testavimas, tačiau privalomai reikalaujama, kad automatiniai testai padengtų tik pagrindinį funkcionalumą, tokiu būdu užtikrinant, kad nauji kodo pakeitimai nenutruks pagrindinių programų sistemų paslaugų teikimo. Modulių testai turi būti leidžiami kaskart pakeitus programų sistemos kodą ir jį įkėlus į kodo saugyklą.

Jeigu modulių testavimo metu aptinkama klaida, kodas negali būti suliejamas (angl. merge) su pagrindine kodo šaka (angl. branch) ir turi būti taisomas. Taip pat, automatiniai modulių testai turi būti leidžiami kaskart prieš diegiant naują kodą į gamybinę aplinką, jeigu aptinkama klaida, diegimas turi būti sustabdytas ir kodas taisomas.

- *Pagrindinio funkcionalumo integracinis testavimas* – naudojamas integracinis testavimas, kuris atliekamas aplinkoje, identiškoje gamybinei aplinkai. Privalomai reikalaujama, kad būtų testuojamas pagrindinis programų sistemos funkcionalumas, tokiu būdu užtikrinant, kad nauji kodo pakeitimai nenutruks pagrindinių programos sistemų paslaugų teikimo. Integracinis testavimas turi būti atliekamas kaskart prieš diegiant naują kodą į gamybinę aplinką. Jeigu integracinio testavimo metu aptinkama klaida, naujas kodas negali būti diegiamas į gamybinę aplinką ir turi būti taisomas.
- *Atskiros testavimo, kūrimo ir gamybos aplinkos* – programų sistemos integracinis testavimas vyksta atskiroje nuo gamybinės testavimo aplinkoje, kuri yra identiškos konfigūracijos. Programų sistema, kuri veikia testavimo aplinkoje, turi lygiai taip pat veikti ir gamybinėje aplinkoje. Programuotojai taip pat turi savo identišką kūrimo aplinką, kurioje gali atlikti savo testavimą, prieš įkeliant kodą į kodo saugyklą.
- *Kodo peržiūros* – pakeitus programų sistemos kodą ir jį įkėlus į kodo saugyklą, kodas yra peržiūrimas bent vieno kito programuotojo. Geriausia, kad kodą peržiūrėtų programuotojas iš kitos komandos. Programuotojas peržiūrintis kodą parašo pastabas ar pažymi klausimus keliančias vietas. Jeigu reikia, kodas tikslinamas, kol peržiūrintis programuotojas nepatvirtina, kad kodas yra geras jis negali būti suliejamas su pagrindine kodo šaka.
- *Incidentų valdymo sistema* – organizacija naudoja incidentų valdymo sistemą, kurioje yra registruojami identifikuotos problemos, užfiksuoti incidentai, jiems priskiriamos komandos ar individualūs asmenys. Registruotas incidentas yra uždaromas po to, kai problema būna išsprendžiama.
- *Testavimo metu identifikuotų problemų registravimas* – identifikuotos problemos, nekorektiškas programų sistemų veikimas yra registruojamas incidentų valdymo sistemoje.

- *Programų sistemos naudotojų identifikuotų problemų registravimas* – problemos, kurias identifikuoja programų sistemų naudotojai ir apie jas praneša, yra registruojamos incidentų valdymo sistemoje.
- *Sisteminio žurnalo įrašai* – programų sistemų apdorojamos užklauskos yra įrašomos į sisteminį žurnalą, taip pat į žurnalą rašomos užklauskų apdorojimo metu įvykusios klaidos.
- Apibrėžtas: 3 brandos lygis
  - *Kokybės proceso srities standartizavimas* – aktyviai vykdomas kokybės proceso srities standartizavimas visos organizacijos mastu. Organizacijoje vykdomos su kokybės užtikrinimu susijusios praktikos yra dokumentuojamos ir diegiamos kaip standartas.
  - *Privalomas naujų modulių testavimas* – visas naujai pridamas programų sistemos kodas yra padengiamas modulių testais.
  - *Automatizuotas integracinis testavimas* – integracinis testavimas vykdomas automatizuotu būdu, nebent tai nėra įmanoma, pavyzdžiui, reikia įvertinti atvaizduojamo lango atitikimą pagal numatytą dizainą.
  - *Privalomas naujo funkcionalumo integracinis testavimas* – visas naujai pridamas programų sistemos funkcionalumas yra padengiamas automatiniiais integraciniais testais.
  - *Automatinė statinė kodo analizė* – statinė kodo analizė gali būti vykdoma nebūtinai kodo surinkimo metu, bet periodiškai. Visi analizės metu identifikuoti saugumo bei kokybės trūkumai yra ištaisomi prieš programų sistemos išleidimą į gamybos aplinką.
  - *Kodo peržiūros tarp skirtingų komandų* – jeigu organizacija turi daugiau nei vieną projektą, vienų projektų komandos turi atlikti kodo peržiūras kitų projektų komandoms, jeigu projektas yra tik vienas, užtenka, kad kodo peržiūrą atliktų kolega iš tos pačios komandos.
  - *Apsaugos priemonių nuo potencialių kenkėjiškų atakų taikymas* – analizuojamos grėsmės, naujai atsirandančios saugumo spragos. Nustačius naują spragą, spraga yra registruojama incidentų valdymo sistemoje ir tvarkoma suinteresuotų asmenų.
- Matuojamas: 4 brandos lygis
  - *Informacijos poreikių dokumentavimas* – reikalingos informacijos, skirtos gerinti kokybės užtikrinimą, poreikiai dokumentuojami ir saugomi.

- *Dokumentuotų informacijos poreikių analizavimas* – dokumentuoti informacijos poreikiai yra analizuojami, peržiūrimi. Nustatomos reikiamos metrikos, skirtos gauti kuo tikslesnę norimą informaciją.
- *Dokumentuotų metrikų rinkimas* – dokumentuotos metrikos, skirtos gauti informacijai, kuri reikalinga gerinti kokybės užtikrinimui, yra renkamos.
- *Naudotojo elgsenos sistemoje stebėjimas* – programų sistema stebi naudotojų elgseną: kiek laiko praleidžiama kiekviename lange, kurie langai populiariausi, koks funkcionalumas naudojamas ar nenaudojamas. Tai leidžia kelti programų sistemos kokybę, atsisakyti nereikalingo funkcionalumo, sumažinti programų sistemos kodo kiekį.
- *Nesėkmingų automatinių testų kiekybinis stebėjimas* – nesėkmingi automatinių testų vykdymai yra kiekybiškai stebimi, taip matuojant bendrą kodo kokybę. Dažnas automatinių testų neveikimas, gali rodyti, jog atlikti kodo pakeitimus programų sistemoje darosi sudėtinga ir, matyt, kodo kokybė krenta.
- *Incidentų gamybinėje aplinkoje stebėjimas* – incidentai įvykę gamybinėje aplinkoje yra automatiškai užfiksuojami, įrašomi ir aiškiai matomi incidentų valdymo sistemoje. Incidentai įvykę gamybinėje aplinkoje turi būti išnagrinėjami ir pataisomi kuo greičiau.
- *Laiko nuo incidento atsiradimo iki jo panaikinimo stebėjimas* – tam, kad būtų galima tiksliau įvertinti žalą bei tobulėti, yra stebimas laikas nuo incidento atsiradimo iki jo panaikinimo.
- *Programų sistemos įprasto eksploatavimo slenksčių nustatymas* – nustatomi įprasto programų sistemos eksploatavimo slenksčiai (angl. thresholds). Peržengti nustatyti slenksčiai identifikuoja nekorektišką paslaugos teikimą. Slenksčiai nustatomi, programų sistemos atsako laikui, serverio apkrovai, duomenų bazių apkrovai ir kitiems su programų sistemos eksploatavimu susijusiems parametrams.
- *Programų sistemos eksploatavimo slenksčių stebėjimas* – renkamos eksploatavimo metrikos, kurios parodo serverio, kuriame įdiegta programų sistema, apkrovą, duomenų bazės serverio apkrovą, kartu su lėčiausiai veikiančių duomenų bazės užklausų sąrašu. Stebimi visi kiti valdomi su programų sistemos veikimu susiję serveriai.
- *Programų sistemos pasiekiamumo stebėjimas* – nuolatos stebimas programų sistemos pasiekiamumas. Jeigu programų sistema tampa nebesiekiamą

naudotojams apie tai turi būti informuojami suinteresuoti asmenys ir problema kuo greičiau sprendžiama.

- *Naudotojų atsiliepimų rinkimas* – programų sistemų naudotojai gali palikti atsiliepimą apie programų sistemą ir jos teikiamas paslaugas.
  - *Matavimo rezultatų apdorojimas* – surinkti matavimų rezultatai yra apdorojami ir paruošiami į lengvai suprantamą ir interpretuojamą formatą, pavyzdžiui, lentelės, grafikai ir t.t. Tai leidžia geriau suprasti surinktus duomenis ir efektyviau optimizuoti programų sistemą.
  - *Matavimo rezultatų analizavimas* – surinkti ir apdoroti matavimo rezultatai yra peržiūrimi ir aptariami suinteresuotų asmenų. Identifikuojamos rizikos ir galimybės bei padaromos išvados, ką reiktų daryti toliau remiantis gautais rezultatais.
- **Optimizuotas: 5 brandos lygis**
    - *Nuolatinis kokybės užtikrinimo optimizavimas* – remiantis turimomis, surinktomis, kokybės metrikomis, kokybės užtikrinimas yra laipsniškai optimizuojamas, šalinant identifikuotas spragas.
    - *Kodo pertvarkymas* – siekiant palaikyti aukštą programų sistemos kodo kokybę, reikia reguliariai atlikti kodo pertvarkymus, taip neleidžiant kodui pasenti, ir jį optimizuojant.
    - *Nuolatinis kodo padengimo testais gerinimas* – nuolatos gerinamas kodo padengimas testais, padengiant dar nepadengtą kodą arba pridodant papildomus testavimo scenarijus jau dalinai padengtam programų sistemos kodui.

## 5.2.2. Valdymas

Šiame poskyryje pateikiama valdymo proceso srities brandos lygių samprata bei taikytinos praktikos kiekviename brandos lygyje. Valdymo proceso sritis yra priskiriama visiems 5 brandos lygiams.

Tikslas – užtikrinti efektyvų komandų valdymą.

### 5.2.2.1. Brandos lygių samprata

Šiame poskyryje pateikta brandos lygių samprata yra patikslina taip, kaip ji turėtų būti suprantama valdymo proceso srityje:

- *Pradinis: 1 brandos lygis* – organizacijos, esančios šiame brandos lygyje, valdymas paprastai būna įgyvendinamas reaktyviai ir chaotiškai. Organizacijos veiklos rezultatai yra nenuspėjami.

- *Valdomas: 2 brandos lygis* – 2 brandos lygyje valdoma laikantis iš anksto nustatytų taisyklių, paremtų gerosiomis praktikomis, kurios ateina iš judriųjų metodikų, tačiau tai vis dar nėra standartizuota ir skirtinguose projektuose valdymas gali vykti skirtingai.
- *Apibrėžtas: 3 brandos lygis* – 3 brandos lygyje organizacijos vykdomas valdymas yra gerai apibrėžtas ir suprantamas bei aprašomas standartuose. Valdymas, skirtinguose projektuose, vykdomas remiantis tais pačiais organizacijos mastu priimtais standartais. Standartai yra pritaikomi kiekvienam organizacijoje vykdomam projektui individualiai jeigu to reikia.
- *Matuojamas: 4 brandos lygis* – 4 brandos lygyje organizacija turi nusistačiusi kiekybinius valdymo proceso srities tikslus ir naudoja juos kaip kriterijus valdydama projektus. Projektų valdymo kiekybiniai tikslai grindžiami kliento, galutinių vartotojų ir organizacijos poreikiais.
- *Optimizuotas: 5 brandos lygis* – 5 brandos lygyje organizacija orientuojasi į nuolatinį valdymo efektyvumo gerinimą, analizuojami turimi kiekybiniai duomenys, kuriais remiantis atliekamas laipsniškas procedūrų tobulinimas.

#### 5.2.2.2. Praktikos

Šiame poskyryje pateikiamos praktikos, kurios yra priskirtos DevOps brandos modelyje esančiai valdymo proceso sričiai, kiekviename brandos lygyje bei tų praktikų detalizavimas:

- *Valdomas: 2 brandos lygis*
  - *Produkto šeimininko rolė* – šios rolės buvimas užtikrina koordinuotą darbą. Rolės turėtojas rūpinasi, kad darbų sąrašas sutaptų su verslo reikalavimais ir kliento poreikiais, nustato darbų prioritetus, šalina iškilusias kliūtis.
  - *Darbas iteracijomis* – iš anksto numatyta, laike apribota, trumpa (trunkantis 1-3 savaites) ir pasikartojanti iteracija, kurios pabaigoje turimas darbo rezultatas gali būti pristatomas.
  - *Projekto reikalavimų sąrašas* – projektas turi savo reikalavimų sąrašą, kuriame yra visi reikalavimai, reikalingi, kad projektas būtų įgyvendintas. Jeigu atsiranda naujų reikalavimų, jie yra pridedami į projekto reikalavimų sąrašą.
  - *Reikalavimų analizavimas* – reikalavimai yra peržiūrimi, nustatomas jų poveikis bei poreikis produktui. Nustatomos analizuojamų reikalavimų rizikos. Reikalavimai prioretizuojami.
  - *Esamos iteracijos reikalavimų sąrašas* – kiekvienai iteracijai yra atliekamas planavimas ir sudaromas reikalavimų sąrašas, kuris yra įgyvendinamas iteracijos metu. Reikalavimai iteracijai imami iš projekto reikalavimų sąrašo,

pagal prioritetą. Reikalavimai yra įvertinami pagal sudėtingumą ir kiek reikalauja darbo laiko. Iteracijos reikalavimų sąrašas neturi keistis iteracijos metu.

- *Iteracijos rezultatų pristatymas komandos viduje* – kiekvienos iteracijos pabaigoje iteracijos rezultatai yra pristatomi komandai, pavyzdžiui, pademonstruojant, kaip veikia naujas funkcionalumas. Produkto šeimininkas gali išsakyti pastebėjimus, į kuriuos atsižvelgiama ir programų sistema taisoma.
- *Pristatyme identifikuotų problemų registravimas* – identifikuotos problemos, nekorektiškas ar neatitinkantis reikalavimų programų sistemų veikimas yra registruojamas, įdedant naujas užduotis į reikalavimų sąrašą.
- **Apibrėžtas: 3 brandos lygis**
  - *Valdymo proceso srities standartizavimas* – aktyviai vykdomas valdymo proceso srities standartizavimas visos organizacijos mastu. Organizacijoje vykdomos su valdymu susijusios praktikos yra dokumentuojamos ir diegiamos kaip standartas.
  - *Bendras organizacijos reikalavimų sąrašas* – visi organizacijoje vykdomi projektai turi reikalavimų sąrašą ir yra sąrašas, kuriame yra visų projektų reikalavimai. Tai leidžia atskiroms komandoms derinti darbus tarpusavyje, išvengti situacijų, kada tą pačią užduotį vykdo kelios skirtingos komandos.
  - *Sinchronizuotos iteracijos tarp projektų* – visi projektai dirba iteracijomis ir skirtingų projektų iteracijos prasideda tuo pačiu metu bei trunka tokį patį laiko tarpą.
  - *Iteracijos rezultatų pristatymas organizacijoje* – kiekvienos iteracijos pabaigoje, kiekvienas projektas pristato savo iteracijos darbo rezultatus viešai organizacijoje taip, kad tai galėtų matyti visų projektų komandos.
- **Matuojamas: 4 brandos lygis**
  - *Informacijos poreikių dokumentavimas* – reikalingos informacijos, skirtos gerinti projektų valdymą, poreikiai dokumentuojami ir saugomi.
  - *Dokumentuotų informacijos poreikių analizavimas* – dokumentuoti informacijos poreikiai yra analizuojami, peržiūrimi. Nustatomos reikiamos metrikos, skirtos gauti kuo tikslesnę norimą informaciją.
  - *Dokumentuotų metrikų rinkimas* – dokumentuotos metrikos, skirtos gauti informacijai, kuri reikalinga gerinti projektų valdymą, yra nuolatos renkamos.

- *Iteracijos progreso stebėjimas* – naudojantis reikalavimų įverčiais, stebimas esamos iteracijos progresas. Kiek darbo vienetų iš visų esančių iteracijoje yra atlikta, bei kiek iteracijos laiko jau praėjo bei liko.
- *Iteracijų atlikto darbo kiekio stebėjimas* – jau praėjusių iteracijų suplanuoto darbo vienetų kiekio ir įgyvendintų darbo vienetų kiekio stebėjimas. Tai padeda įsivertinti kaip gerai yra planuojamos iteracijos.
- *Matavimo rezultatų apdorojimas* – surinkti matavimų rezultatai yra apdorojami ir paruošiami į lengvai suprantamą ir interpretuojamą formatą, pavyzdžiui, lentelės, grafikai ir t.t. Tai leidžia geriau suprasti surinktus duomenis ir efektyviau optimizuoti programų sistemą.
- *Matavimo rezultatų analizavimas* – surinkti ir apdoroti matavimo rezultatai yra peržiūrimi ir aptariami suinteresuotų asmenų. Identifikuojamos rizikos, galimybės ir spragos, bei padaromos išvados, ką reiktų daryti toliau remiantis gautais rezultatais. Analizės rezultatai, kurie reikalauja tolimesnių veiksmų yra registruojami.
- Optimizuotas: 5 brandos lygis
  - *Nuolatinis projektų valdymo optimizavimas* – remiantis turimomis, surinktomis, valdymo metrikomis, projektų valdymas yra laipsniškai optimizuojamas, šalinant identifikuotas spragas, apsisaugant nuo galimų rizikų bei išnaudojant esamas galimybes.
  - *Iteracijų peržiūros* – rezultatų peržiūrėjimas siekiant įvertinti atliktą iteracijos planavimą – ar buvo tinkamai įvertinti reikalavimai? Ar į iteraciją nebuvo įtraukta per daug/per mažai reikalavimų? Jeigu ne viskas buvo kaip planuota, turi būti identifikuotos priežastys. Iteracijų peržiūros padeda gerinti kitų iteracijų planavimą. Neįvykdyti reikalavimai grąžinami atgal į projekto reikalavimų sąrašą. Analizuojamos surinktos metrikos.
  - *Retrospektyvos* – atliekamos tam, kad būtų galima patikrinti, kaip vyko pastaroji iteracija žmonių, santykių, procesų ir įrankių atžvilgiu. Retrospektyvų metu komanda turi identifikuoti gerinimo priemones, kurias ji įgyvendins kitos iteracijos metu.
  - *Identifikuotų problemų registravimas* – iteracijų peržiūrų ir retrospektyvų metu identifikuotos problemos yra registruojamos ir prioretizuojamos.
  - *Registruotų problemų sprendimas* – registruotos problemos yra sprendžiamos jų prioriteto tvarka. Analizuojami galimi problemos sprendimo veiksmai.



Sprendimas yra įgyvendinamas ir dokumentuojamas. Išsprendus problemą, ji panaikinama iš registruotų problemų sąrašo.

### 5.2.3. Diegimas

Šiame poskyryje pateikiama diegimo proceso srities brandos lygių samprata bei taikytinos praktikos kiekviename brandos lygyje. Diegimo proceso sritis yra priskiriama visiems 5 brandos lygiams.

Tikslas – užtikrinti automatizuotą naujų programų sistemų versijų leidimą, nepertraukiant programų sistemų veikimo gamybinėje aplinkoje.

#### 5.2.3.1. Brandos lygių samprata

Šiame poskyryje pateikta samprata yra patikslina taip, kaip ji turėtų būti suprantama diegimo proceso srityje:

- *Pradinis: 1 brandos lygis* – organizacijos, esančios šiame brandos lygyje, diegimą vykdo chaotiškai, dažniausiai rankiniu būdu. Atliekant diegimą dažnai neišvengiama ilgesnio prastovos.
- *Valdomas: 2 brandos lygis* – 2 brandos lygyje programų sistemos diegiamos laikantis iš anksto nustatytų taisyklių, nuosekliai, automatizuotu būdu, tačiau tai vis dar nėra standartizuota ir skirtinguose projektuose diegimas gali vykti skirtingai.
- *Apibrėžtas: 3 brandos lygis* – 3 brandos lygyje organizacijos vykdomas diegimas yra gerai apibrėžtas ir suprantamas bei aprašomas standartuose. Diegimas, skirtinguose projektuose, vykdomas remiantis tais pačiais organizacijos mastu priimtais standartais. Standartai yra pritaikomi kiekvienam organizacijoje vykdomam projektui individualiai jeigu to reikia.
- *Matuojamas: 4 brandos lygis* – 4 brandos lygyje organizacija stebi ir kaupia su diegimu susijusias metrikas, kurios parodo diegimo proceso srities efektyvumą.
- *Optimizuotas: 5 brandos lygis* – 5 brandos lygyje organizacija orientuojasi į nuolatinį diegimo efektyvumo gerinimą, analizuojami turimi kiekybiniai duomenys, kuriais remiantis atliekamas laipsniškas procedūrų tobulinimas.

#### 5.2.3.2. Praktikos

Šiame poskyryje pateikiamos praktikos, kurios yra priskirtos DevOps brandos modelyje esančiai diegimo proceso sričiai, kiekviename brandos lygyje bei tų praktikų detalizavimas:

- *Valdomas: 2 brandos lygis*
  - *Automatizuotas programinės įrangos aplinkų kūrimas* – aplinkos (testavimo, gamybos ir t.t.) sukūrimas yra automatizuotas ir atkartojamas, t.y. kiekvieną

kartą iš naujo kuriant aplinką turi būti sukuriama identiška aplinka. Aplinkos automatizuotam kūrimui gali būti naudojami tokie įrankiai kaip Ansible, Chef, Puppet ar kiti.

- *Automatizuotas kodo surinkimas* – programų sistemų kodas yra surenkamas automatiškai: paduodami reikiami atributai ir iškviečiamos atitinkamos komandos, reikalingos surinkti kodą.
- *Automatizuotas programų sistemos diegimas* – programų sistemos diegimas yra aprašytas kodu ir atkartojamas. Diegimo procesas gali būti iškviečiamas rankiniu būdu, tačiau jį paleidus rankinio įsikišimo neturi būti.
- *Planuojami ir valdomi versijų išleidimai* – naujos programų sistemų versijos išleidžiamos iš anksto numatytu laiku ir iš anksto numatytais pakeitimais. Į naują versiją įtraukiami tik planuoti programų sistemos pakeitimai bei kritiniai klaidų taisymai.
- *Programų sistemos diegimas į testavimo aplinką pagal poreikį* – yra galimybė įdiegti norimą programų sistemos versiją į testavimo aplinką bet kuriuo metu. Tam atlikti neturi reikėti tiesioginės prieigos prie serverio su testavimo aplinka.
- *Galimybė atstatyti anksčiau įdiegtą versiją* – yra galimybė atšaukti esamą programų sistemos versiją ir įdiegti ankstesnę versiją.
- *Programų sistemos diegimas be prastovos* – nauja programų sistemos versija yra diegiama taip, kad būtų išvengta prastovos.
- *Duomenų bazių migracijos be prastovos* – duomenų bazių migracijos yra atliekamos taip, kad būtų išvengta prastovos.
- Apibrėžtas: 3 brandos lygis
  - *Diegimo proceso srities standartizavimas* – aktyviai vykdomas diegimo proceso srities standartizavimas visos organizacijos mastu. Organizacijoje vykdomos su diegimu susijusios praktikos yra dokumentuojamos ir diegiamos kaip standartas.
  - *Planuojami ir valdomi skirtingų projektų versijų išleidimai* – naujų programų sistemų versijų išleidimas planuojamas visos organizacijos mastu, taip, kad skirtingų projektų atliekami pakeitimai būtų suderinami. Į naują versiją įtraukiami tik iš anksto suplanuoti ir skirtingų projektų tarpusavyje suderinti pakeitimai bei kritiniai klaidų taisymai.
  - *Skirtingos programų sistemos versijos testavimo aplinkoje* – galimybė vienu metu testavimo aplinkoje turėti skirtingas programų sistemos versijas.

- Matuojamas: 4 brandos lygis
  - *Informacijos poreikių dokumentavimas* – reikalingų metrikų, skirtų suprasti ir gerinti organizacijos diegimo proceso sritį, poreikiai dokumentuojami ir saugomi.
  - *Dokumentuotų informacijos poreikių analizavimas* – dokumentuoti informacijos poreikiai yra analizuojami, peržiūrimi. Nustatomos reikiamos metrikos, skirtos gauti kuo tikslesnę norimą informaciją.
  - *Dokumentuotų metrikų rinkimas* – dokumentuotos metrikos, skirtos gauti informacijai, kuri reikalinga gerinti diegimą, yra nuolatos renkamos.
  - *Diegimo dažnumo stebėjimas* – stebima, kiek dažnai arba kiek diegimų atliekama per dieną.
  - *Nepavykusių diegimų kiekio stebėjimas* – stebima, kiek nepavykusių diegimų atliekama per dieną, savaitę ar kitą organizacijai prasmingą laiko tarpą.
  - *Diegimo laiko stebėjimas* – stebima, kiek laiko trunka diegimas.
  - *Matavimo rezultatų apdorojimas* – surinkti matavimų rezultatai yra apdorojami ir paruošiami į lengvai suprantamą ir interpretuojamą formatą, pavyzdžiui, lentelės, grafikai ir t.t. Tai leidžia geriau suprasti surinktus duomenis ir efektyviau optimizuoti diegimą.
  - *Matavimo rezultatų analizavimas* – surinkti ir apdoroti matavimo rezultatai yra peržiūrimi ir aptariami suinteresuotų asmenų. Identifikuojamos rizikos, galimybės ir spragos, bei padaromos išvados, ką reiktų daryti toliau remiantis gautais rezultatais. Analizės rezultatai, kurie reikalauja tolimesnių veiksmų yra registruojami.
- Optimizuotas: 5 brandos lygis
  - *Nuolatinis diegimo optimizavimas* – remiantis turimomis, surinktomis, diegimo metrikomis, diegimas yra laipsniškai optimizuojamas, šalinant identifikuotas spragas, apsisaugant nuo galimų rizikų bei išnaudojant esamas galimybes.
  - *Funkcionalumo konfigūracija* – programų sistemų funkcionalumas kontroliuojamas naudojantis konfigūracija. Priklausomai nuo to, kokia yra konfigūracija, tam tikros programų sistemos kodo šakos yra įjungiamos arba išjungiamos. Tokiu būdu gamybinėje aplinkoje gali būti programų sistemų kodas, kuris dar nėra paruoštas naudojimui (konfigūracijoje išjungtas).

- *Nuolatinis diegimas į gamybos aplinką* – visi pakeitimai automatiškai diegiami į gamybos aplinką. Funkcionalumas apsaugomas konfigūracija. Naujausia programų sistemos versija visada yra „master“ šakoje.

## 5.2.4. Dalinimasis

Šiame poskyryje pateikiama dalinimosi proceso srities brandos lygių samprata bei taikytinos praktikos kiekviename brandos lygyje. Dalinimosi proceso sritis yra priskiriama pirmiems 3 brandos lygiams.

Tikslas – užtikrinti, kad svarbia informacija būtų dalinamasi ir reikiama informacija būtų lengvai gaunama.

### 5.2.4.1. Brandos lygių samprata

Šiame poskyryje pateikta samprata yra patikslina taip, kaip ji turėtų būti suprantama dalinimosi proceso srityje:

- *Pradinis: 1 brandos lygis* – organizacijos, esančios šiame brandos lygyje, dalinimasis paprastai būna įgyvendinamas reaktyviai ir chaotiškai. Darbuotojai dažnai neturi visos reikiamos informacijos arba turi pasenusią informaciją ir dėl to negali priimti geriausių bei naudingiausių sprendimų.
- *Valdomas: 2 brandos lygis* – 2 brandos lygyje dalinimasis vyksta sistemingai, skleidžiama svarbi su projektu susijusi informacija, pagrindinė komunikacijos forma - rašytinė, tokiu būdu išvengiant informacijos iškraipymo ar riboto paskleidimo. Taip pat, vyksta techninių bei kitų profesinių žinių dalinimasis komandų viduje. Tačiau šiame brandos lygyje informacija dalinimosi proceso sritis dar netampa standartu ir atskirose komandose ar projektuose gali būti įgyvendinama skirtingai.
- *Apibrėžtas: 3 brandos lygis* – 3 brandos lygyje organizacijos vykdomas informacijos dalinimasis yra gerai apibrėžtas ir suprantamas bei aprašomas standartuose. Informacijos dalinimasis, skirtinguose projektuose, vykdomas remiantis tais pačiais organizacijos mastu priimtais standartais. Standartai yra pritaikomi kiekvienam organizacijoje vykdomam projektui individualiai jeigu to reikia.

### 5.2.4.2. Praktikos

Šiame poskyryje pateikiamos praktikos, kurios yra priskirtos DevOps brandos modelyje esančiai dalinimosi proceso sričiai, kiekviename brandos lygyje bei tų praktikų detalizavimas:

- *Valdomas: 2 brandos lygis*

- *Rašytinė komunikacija* – organizacijoje yra naudojama rašytinė komunikacija elektroninėje erdvėje. Rašytinė komunikacija yra tikslesnė ir aiškesnė, išvengiama situacijų, kai perduota informacija yra pamirštama.
- *Greita tiesioginė komunikacija* – organizacija suteikia galimybę darbuotojams greitai ir tiesiogiai komunikuoti tarpusavyje. Tą galima įgyvendinti naudojantis programine įranga, tokia, kaip, pavyzdžiui, Slack, HipChat ir pan. Tokia komunikacija reikalinga efektyviam problemų sprendimui, siekiant gauti reikiamą informaciją prieš priimant sprendimą.
- *Projekto informacijos nuolatinis skleidimas* – informacija, susijusi su vykdomu projektu, yra nuolatos skleidžiama ir atnaujinama taip, kad visi darbuotojai, dirbantys prie konkretaus projekto, žinotų visą naujausią, su projektu susijusią informaciją ir galėtų priimti geriausius sprendimus, atitinkančius esamą projekto statusą.
- *Techninių ir kitų profesinių žinių nuolatinis skleidimas* – techninėmis ir kitomis profesinėmis žiniomis yra dalinamasi. Darbuotojai turintys daugiau žinių, tomis žiniomis dalinasi su mažiau žinių turinčiais darbuotojais. Skiriamas darbo laikas padėti, dirbama kartu. Taip pat, gali būti rengiami seminarai, kurių metu dalinamasi žiniomis.
- *Sisteminių žurnalo įrašų (angl. logs) matomumas* – sisteminio žurnalo įrašai yra lengvai randami ir matomi, tam, kad būtų galima efektyviai identifikuoti problemas ir greitai jas spręsti.
- **Apibrėžtas: 3 brandos lygis**
  - *Dalinimosi proceso srities standartizavimas* – aktyviai vykdomas dalinimosi proceso srities standartizavimas. Organizacijoje vykdomos su dalinimosi susijusios praktikos yra dokumentuojamos ir diegiamos kaip standartas.
  - *Atvira komunikacija* – organizacijos komunikacija vyksta organizacijos darbuotojams matomoje erdvėje, kur darbuotojai gali taip pat išsakyti savo nuomonę, palikti komentarą.
  - *Bendra kalba rašytinei komunikacijai* – rašytinė komunikacija vyksta bendra, visiems organizacijoje suprantama kalba. Pavyzdžiui, esant tarptautinei organizacijai, kurios bendra kalba yra anglų ir kurioje taip pat dirba ir lietuviai, jeigu lietuvis atlieka kodo peržiūrą kitam lietuviui – tai turi būti daroma anglų kalba.

- *Organizacijos renkamu metrikų komunikavimas* – visos organizacijos surenkamos metrikos susijusios su kuriamomis programų sistemomis yra lengvai randamos ir komunikuojamos suinteresuotiems asmenims organizacijoje.
- *Dokumentacija* – visos organizacijos mastu priimtas standartas, kad visas programų sistemos funkcionalumas ir kita susijusi informacija turi būti dokumentuoti, o dokumentacija lengvai randama ir matoma visų suinteresuotų asmenų organizacijoje.

## 5.2.5. Kultūra

Šiame poskyryje pateikiama kultūros proceso srities brandos lygių samprata bei taikytinos praktikos kiekviename brandos lygyje. Kultūros proceso sritis yra priskiriama pirmiems 3 brandos lygiams.

Tikslas – užtikrinti, kad DevOps kultūriniai aspektai organizacijoje būtų įgyvendinami.

### 5.2.5.1. Brandos lygių samprata

Šiame poskyryje pateikta samprata yra patikslina taip, kaip ji turėtų būti suprantama kultūros proceso srityje:

- *Pradinis: 1 brandos lygis* – organizacijos, esančios šiame brandos lygyje, DevOps kultūra praktiškai neegzistuoja.
- *Valdomas: 2 brandos lygis* – 2 brandos lygyje DevOps kultūra tampa svarbi organizacijos viduje. Atliekami reikiami kultūriniai pokyčiai, darbai pradedami daryti mažesnėmis apimtimis, atskiriant skirtingus darbus. Darbo aplinka yra saugi, darbuotojai nėra kaltinami už įvykusius incidentus, stengiamasi išsiaiškinti priežastis ir užkirsti kelią to pasikartojimui ateityje. Tačiau, DevOps kultūra dar nėra priimta visos organizacijos ir skirtingose komandose gali būti suprantama ir vykdoma skirtingai.
- *Apibrėžtas: 3 brandos lygis* – 3 brandos lygyje organizacijos DevOps kultūra yra gerai apibrėžta ir suprantama bei aprašomas standartuose. DevOps kultūra, skirtinguose projektuose, taikoma remiantis tais pačiais organizacijos mastu priimtais standartais.

### 5.2.5.2. Praktikos

Šiame poskyryje pateikiamos praktikos, kurios yra priskirtos DevOps brandos modelyje esančiai kultūros proceso sričiai, kiekviename brandos lygyje bei tų praktikų detalizavimas:

- Valdomas: 2 brandos lygis
  - *Izoliuoti pakeitimai* – vykdomi programų sistemos pakeitimai yra izoliuoti nuo kitų vykdomų pakeitimų, t.y. vienu kodo pakeitimu negali būti atliekami keli su skirtingu funkcionalumu susiję pakeitimai.
  - *Maži pakeitimai* – atliekami pakeitimai yra nedideli, tokie, kad juos būtų lengva peržiūrėti kolegoms bei esant poreikiui atstatyti atgal.
  - *Dažnas atgalinis ryšys* – atliekamas darbas gauna dažną atgalinį ryšį, nes yra matomas kitų kolegų ar klientų, taip išvengiant situacijų, kai ilgai atliekamas niekieno nematomas darbas būna neprasmingas ar net žalingas organizacijai.
  - *Peržiūros po incidentų be kaltinimų* – ši praktika įgalina komandas giliau pažvelgti į incidentą, siekiant išsiaiškinti, kas įvyko, kodėl tai įvyko, kaip komanda reagavo ir ką galima būtų padaryti siekiant užkirsti kelią pakartotiniams tokiems pat ar panašiams incidentams ateityje.
- Apibrėžtas: 3 brandos lygis
  - *Kultūros proceso srities standartizavimas* – aktyviai vykdomas kultūros proceso srities standartizavimas. Organizacijoje vykdomos su kultūra susijusios praktikos yra dokumentuojamos ir diegiamos kaip standartas.
  - *Darbuotojų įgalinimas priimti sprendimus* – darbuotojai turintys daugiausiai žinių susijusių su problema gali ir yra skatinami patys priimti sprendimus, užuot prašant sprendimo iš hierarchijoje esančio aukščiau darbuotojo, kuris galimai nėra taip įsigilinęs į problemą.
  - *Nuolatinis mokymasis* – organizacija užtikrina nuolatinį savo darbuotojų profesinį kvalifikacijos kėlimą. Darbuotojai gali būti siunčiami į jų profesijų atstovams aktualias konferencijas, apmokymus ir pan.
  - *Nuopelnų skyrimas komandoms* – organizacija pastebi ar apdovanoja už pasiekimus, bet pastebimi yra ne individualūs asmenys, o komandos. DevOps kultūra skatina bendradarbiavimą ir komandinį darbą.

### 5.3. Apibendrinimas

Remiantis priimtais sprendimais buvo sudarytas naujas DevOps brandos modelis. Darbe buvo nustatyta, kad šiuo metu nėra praktiškai taikomo DevOps brandos modelio. Nagrinėti DevOps brandos modeliai negali būti taikomi praktiškai dėl jų nepakankamo detalumo: modeliuose trūksta praktikų, jų paaiškinimų, nei vienas modelis nenurodo, kaip reikia atlikti vertinimą.

Kuriant naują DevOps brandos modelį, buvo išspręstos anksčiau identifikuotos problemos. Naujas DevOps brandos modelis turi apibrėžtą brandos lygių sampratą, kiekvienai proceso sričiai,

kiekviename brandos lygyje (išskyrus pirmąjį), yra priskirtos praktikos, su jų detalizavimu. Pirmam brandos lygiui nėra keliami reikalavimai, todėl visos organizacijos gali gauti įvertį. Taip pat, sudaryta praktikų vertinimo skalė ir nurodytos sąlygos, kada yra įgyvendinta proceso sritis bei brandos lygis. Naujas DevOps brandos modelis turi visus reikiamus modelio elementus, kad būtų galima atlikti DevOps proceso brandos vertinimą, tačiau nėra perkrautas pertekliniais reikalavimais, painiais išaiškinamais, kurie reikalautų modelio autorių pagalbos, tam, kad būtų galima atlikti vertinimą, kaip, pavyzdžiui, yra su CMMI-DEV [CMD10] brandos modeliu, kurio sudėtingumas yra toks aukštas, kad vertinimui atlikti reikalingi sertifikuoti vertintojai [CMA14].

Taigi, remiantis įvardintais argumentais, galima teigti, kad naujai pasiūlytas DevOps brandos modelis gali būti taikomas praktiškai.

10 lentelėje pateikiamas naujo DevOps brandos modelio apibendrinimas, apimantis visas proceso sritis bei brandos lygius su jų praktikomis.

10 lentelė. Naujo DevOps brandos modelio apibendrinimas.

	<b>Pradinis</b>	<b>Valdomas</b>	<b>Apibrėžtas</b>	<b>Matuojamas</b>	<b>Optimizuotas</b>
<b>Kokybė</b>	-	<ul style="list-style-type: none"> <li>• Kodo versijų kontroliavimas</li> <li>• Kodo prieigos kontroliavimas</li> <li>• Pagrindinio programų sistemos funkcijų identifikavimas</li> <li>• Pagrindinių modulių testavimas</li> <li>• Pagrindinio funkcionalumo integracinis testavimas</li> <li>• Atskiros testavimo, kūrimo ir gamybos aplinkos</li> <li>• Kodo peržiūros</li> <li>• Incidentų valdymo sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Kokybės proceso srities standartizavimas</li> <li>• Privalomas naujų modulių testavimas</li> <li>• Automatizuotas integracinis testavimas</li> <li>• Privalomas naujo funkcionalumo integracinis testavimas</li> <li>• Automatinė statinė kodo analizė</li> <li>• Kodo peržiūros tarp skirtingų komandų</li> <li>• Apsaugos priemonių nuo potencialių kenkėjiškų atakų taikymas</li> </ul>	<ul style="list-style-type: none"> <li>• Informacijos poreikių dokumentavimas</li> <li>• Dokumentuotų informacijos poreikių analizavimas</li> <li>• Dokumentuotų metrikų rinkimas</li> <li>• Naudotojo elgsenos sistemoje stebėjimas</li> <li>• Nesėkmingų automatinį testų kiekybinis stebėjimas</li> <li>• Incidentų gamybinėje aplinkoje stebėjimas</li> <li>• Laiko nuo incidento atsiradimo iki jo panaikinimo stebėjimas</li> </ul>	<ul style="list-style-type: none"> <li>• Nuolatinis kokybės užtikrinimo optimizavimas</li> <li>• Kodo pertvarkymas</li> <li>• Nuolatinis kodo padengimo testais gerinimas</li> </ul>



		<ul style="list-style-type: none"> <li>• Testavimo metu identifikuotų problemų registravimas</li> <li>• Programų sistemos naudotojų identifikuotų problemų registravimas</li> <li>• Sisteminio žurnalo įrašai</li> </ul>		<ul style="list-style-type: none"> <li>• Programų sistemos įprasto eksploataavimo slenksčių nustatymas</li> <li>• Programų sistemos eksploataavimo slenksčių stebėjimas</li> <li>• Programų sistemos pasiekiamumo stebėjimas</li> <li>• Naudotojų atsiliepimų rinkimas</li> <li>• Matavimo rezultatų apdorojimas</li> <li>• Matavimo rezultatų analizavimas</li> </ul>	
<b>Valdymas</b>	-	<ul style="list-style-type: none"> <li>• Produkto šeimnininko rolė</li> <li>• Darbas iteracijomis</li> <li>• Projekto reikalavimų sąrašas</li> <li>• Reikalavimų analizavimas</li> <li>• Esamos iteracijos reikalavimų sąrašas</li> <li>• Iteracijos rezultatų pristatymas komandos viduje</li> <li>• Pristatyme identifikuotų problemų registravimas</li> </ul>	<ul style="list-style-type: none"> <li>• Valdymo proceso srities standartizavimas</li> <li>• Bendras organizacijos reikalavimų sąrašas</li> <li>• Sinchronizuotos iteracijos tarp projektų</li> <li>• Iteracijos rezultatų pristatymas organizacijoje</li> </ul>	<ul style="list-style-type: none"> <li>• Informacijos poreikių dokumentavimas</li> <li>• Dokumentuotų informacijos poreikių analizavimas</li> <li>• Dokumentuotų metrikų rinkimas</li> <li>• Iteracijos progreso stebėjimas</li> <li>• Iteracijų atlikto darbo kiekio stebėjimas</li> <li>• Matavimo rezultatų apdorojimas</li> <li>• Matavimo rezultatų analizavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Nuolatinis projektų valdymo optimizavimas</li> <li>• Iteracijų peržiūros</li> <li>• Retrospektyvos</li> <li>• Identifikuotų problemų registravimas</li> <li>• Registruotų problemų sprendimas</li> </ul>

<b>Diegimas</b>	-	<ul style="list-style-type: none"> <li>• Automatizuotas programinės įrangos aplinkų kūrimas</li> <li>• Automatizuotas kodo surinkimas</li> <li>• Automatizuotas programų sistemos diegimas</li> <li>• Planuojami ir valdomi versijų išleidimai</li> <li>• Programų sistemos diegimas į testavimo aplinką pagal poreikį</li> <li>• Galimybė atstatyti anksčiau įdiegtą versiją</li> <li>• Programų sistemos diegimas be prastovos</li> <li>• Duomenų bazių migracijos be prastovos</li> </ul>	<ul style="list-style-type: none"> <li>• Diegimo proceso srities standartizavimas</li> <li>• Planuojami ir valdomi skirtingų projektų versijų išleidimai</li> <li>• Skirtingos programų sistemos versijos testavimo aplinkoje</li> </ul>	<ul style="list-style-type: none"> <li>• Informacijos poreikių dokumentavimas</li> <li>• Dokumentuotų informacijos poreikių analizavimas</li> <li>• Dokumentuotų metrikų rinkimas</li> <li>• Diegimo dažnumo stebėjimas</li> <li>• Nepavykusių diegimų kiekio stebėjimas</li> <li>• Diegimo laiko stebėjimas</li> <li>• Matavimo rezultatų apdorojimas</li> <li>• Matavimo rezultatų analizavimas</li> </ul>	<ul style="list-style-type: none"> <li>• Nuolatinis diegimo optimizavimas</li> <li>• Funkcionalumo konfigūracija</li> <li>• Nuolatinis diegimas į gamybos aplinką</li> </ul>
<b>Dalinimasis</b>	-	<ul style="list-style-type: none"> <li>• Rašytinė komunikacija</li> <li>• Greita tiesioginė komunikacija</li> <li>• Projekto informacijos nuolatinis skleidimas</li> <li>• Techninių ir kitų profesinių žinių nuolatinis skleidimas</li> </ul>	<ul style="list-style-type: none"> <li>• Dalinimosi proceso srities standartizavimas</li> <li>• Atvira komunikacija</li> <li>• Bendra kalba rašytinei komunikacijai</li> <li>• Organizacijos renkamų metrikų komunikavimas</li> </ul>	-	-

		<ul style="list-style-type: none"> <li>• Sisteminių žurnalo įrašų matomumas</li> </ul>	<ul style="list-style-type: none"> <li>• Dokumentacija</li> </ul>		
<b>Kultūra</b>	-	<ul style="list-style-type: none"> <li>• Izoliuoti pakeitimai</li> <li>• Maži pakeitimai</li> <li>• Dažnas atgalinis ryšys</li> <li>• Peržiūros po incidentų be kaltinimų</li> </ul>	<ul style="list-style-type: none"> <li>• Kultūros proceso srities standartizavimas</li> <li>• Darbuotojų įgalinimas priimti sprendimus</li> <li>• Nuolatinis mokymasis</li> <li>• Nuopelnų skyrimas komandoms</li> </ul>	-	-

## 6. CMMI GEBĖJIMO PROFILIAI

Siekiant identifikuoti naujai pasiūlyto DevOps brandos modelio trūkumus ir jį pagerinti, arba įsitikinti, kad naujas modelis yra tinkamas naudojimui, reikėjo atlikti vertinimą. Kadangi DevOps apjungia kūrimą bei eksploatavimą, nuspręsta vertinimą atlikti remiantis būtent kūrimo (CMMI-DEV [CMD10]) ir eksploatavimo (CMMI-SVC [CMS10]) modeliais.

Šio skyriaus poskyriuose yra pateikti apibendrinti vertinimo rezultatai. Vertinimas buvo atliktas įsivaizduojamai organizacijai, kuri tenkina 5 brandos lygį, pagal naują DevOps brandos modelį. Sudaryti gebėjimo profiliai pagal CMMI-DEV [CMD10] ir CMMI-SVC [CMS10] modelius.

### 6.1. Gebėjimo profilis pagal CMMI-DEV

Šiame poskyryje nustatyta koki gebėjimo profilių užtikrina naujai pasiūlytas DevOps brandos modelis pagal CMMI-DEV [CMD10] modelį.

1 priede, yra pateiktas vertinimas, 5 brandos lygį atitinkančios, pagal naują DevOps brandos modelį, organizacijos, pagal CMMI-DEV [CMD10]. Vertinimui buvo naudojamos visos CMMI-DEV [CMD10] proceso sritys, išskyrus: „Tiekėjų sutarčių valdymas“, kadangi ši proceso sritis akivaizdžiai nėra susijusi su DevOps.

11 lentelėje yra pateiktas 1 priede atlikto vertinimo rezultatų apibendrinimas. Specifiniai tikslai žymimi „SG“, nurodant specifinio tikslo numerį. „U“ rodo, kad specifinis tikslas nėra įgyvendintas, „S“ – įgyvendintas, o „-“ rodo, kad specifinio tikslo tokiu numeriu proceso srityje nėra. Taip pat, lentelėje nurodomas kiekvienos proceso srities vertinimas – gebėjimo lygis.

11 lentelė. 5 brandos lygio organizacijos vertinimas pagal CMMI-DEV [CMD10].

Proceso sritis	Gebėjimo lygis	SG 1	SG 2	SG 3
CM - Konfigūracijos valdymas	0	U	S	U
MA - Matavimas ir analizė	0	U	U	-
PMC - Projekto stebėjimas ir kontrolė	0	U	U	-
PP - Projekto planavimas	0	U	U	U
PPQA - Proceso ir produkto kokybės užtikrinimas	0	U	U	-
REQM - Reikalavimų valdymas	0	U	-	-
DAR - Sprendimų analizė ir nutarimas	0	U	-	-
IPM - Integruotas projektų valdymas	0	U	U	-
OPD - Organizacinio proceso apibrėžimas	0	U	-	-
OPF - Organizacinio proceso dėmesys	0	U	U	U
OT - Organizacinis mokymas	0	U	U	-
PI - Produkto integracija	0	U	U	U
RD - Reikalavimų kūrimas	0	S	U	U
RSKM - Rizikos valdymas	0	U	U	U
TS - Techninis sprendimas	0	U	U	U
VAL - Patvirtinimas	0	U	S	-
VER - Verifikavimas	0	U	U	U
OPP - Organizacinio proceso efektyvumas	0	U	-	-

QPM - Kiekybinis projekto valdymas	0	U	U	-
CAR - Priežastinio ryšio analizė ir nutarimas	0	U	U	-
OPM - Organizacijos našumo valdymas	0	U	U	U

Remiantis 11 lentelės duomenimis 6 paveiksluke yra pateiktas 5 brandos lygio organizacijos gebėjimo profilis pagal CMMI-DEV [CMD10].

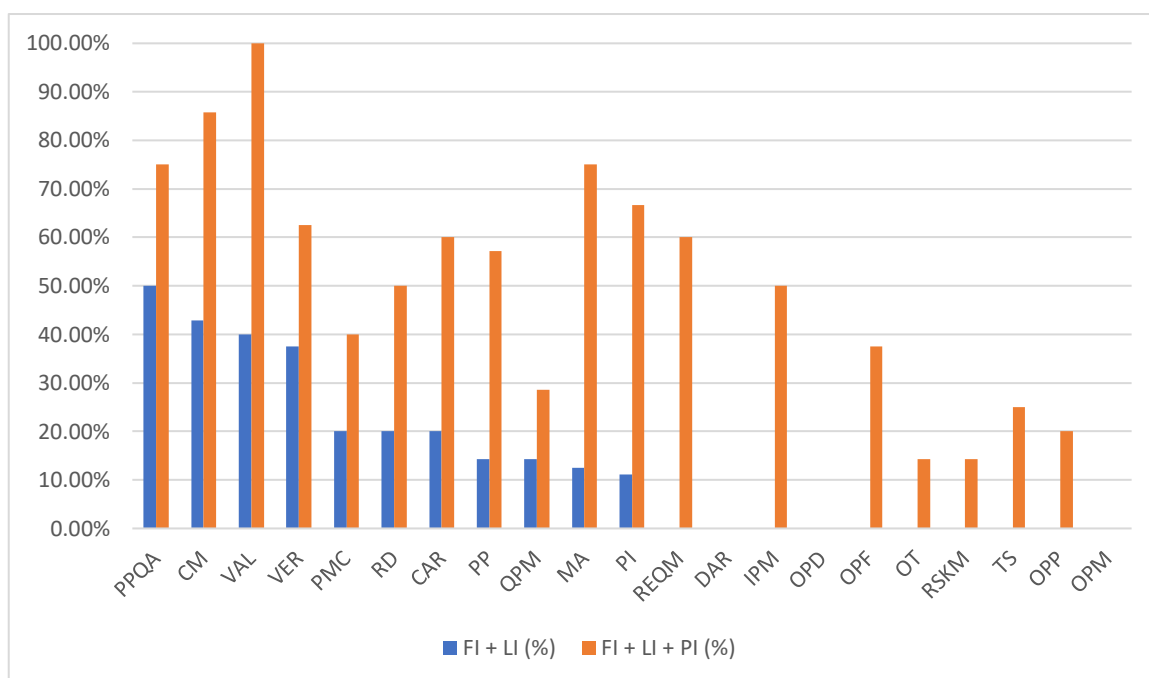


6 pav. 5 brandos lygio organizacijos CMMI-DEV [CMD10] gebėjimo profilis.

Nors vertinimo metu buvo nustatytas ne vienas įgyvendintas specifinis tikslas, tačiau nei viena proceso sritis nepasiekė 1 gebėjimo lygio. Tam, kad būtų įgyvendintas specifinis tikslas pagal CMMI, visos specifinės praktikos, susietos su specifiniu tikslu, turi būti įvertintos LI arba FI. Be to, jeigu yra nors viena specifinė praktika, kurios įvertis yra LI t.y. buvo nustatyti trūkumai – turi nebūti trūkumų, kurie neleistų pasiekti specifinio tikslo. CMMI-DEV [CMD10] modelis vertina daugiau proceso sričių ir ne visos jos yra tiesiogiai DevOps rūpestis, taip pat CMMI-DEV [CMD10] modelis yra griežtesnis – nurodo platesnę aibę taikytinų praktikų, su joms priskirtomis detaliosiomis praktikomis, kiekvienoje proceso srityje, nei naujas DevOps brandos modelis, dėl ko, daug specifinių praktikų yra tik dalinai įgyvendintos ir todėl 1 gebėjimo lygis nėra pasiekiamas. 12 lentelėje bei 7 paveiksluke pateikti duomenys, kiek specifinių praktikų, kiekvienoje proceso srityje, buvo įvertintos kiekvienu įverčiu bei pateikta, kokį procentą sudaro FI ir LI (kartu) įverčiai, bei kokį – FI, LI ir PI.

12 lentelė. Vertinimo pagal CMMI-DEV specifinių praktikų įverčių pasiskirstymas.

Proceso sritis	FI	LI	PI	NI	FI + LI (%)	FI + LI + PI (%)
PPQA - Proceso ir produkto kokybės užtikrinimas	1	1	1	1	50,00%	75,00%
CM - Konfigūracijos valdymas	0	3	3	1	42,86%	85,71%
VAL - Patvirtinimas	1	1	3	0	40,00%	100,00%
VER - Verifikavimas	1	2	2	3	37,50%	62,50%
PMC - Projekto stebėjimas ir kontrolė	0	2	2	6	20,00%	40,00%
RD - Reikalavimų kūrimas	2	0	3	5	20,00%	50,00%
CAR - Priežastinio ryšio analizė ir nutarimas	0	1	2	2	20,00%	60,00%
PP - Projekto planavimas	1	1	6	6	14,29%	57,14%
QPM - Kiekybinis projekto valdymas	0	1	1	5	14,29%	28,57%
MA - Matavimas ir analizė	1	0	5	2	12,50%	75,00%
PI - Produkto integracija	0	1	5	3	11,11%	66,67%
REQM - Reikalavimų valdymas	0	0	3	2	0,00%	60,00%
DAR - Sprendimų analizė ir nutarimas	0	0	0	6	0,00%	0,00%
IPM - Integruotas projektų valdymas	0	0	5	5	0,00%	50,00%
OPD - Organizacinio proceso apibrėžimas	0	0	0	7	0,00%	0,00%
OPF - Organizacinio proceso dėmesys	0	0	3	5	0,00%	37,50%
OT - Organizacinis mokymas	0	0	1	6	0,00%	14,29%
RSKM - Rizikos valdymas	0	0	1	6	0,00%	14,29%
TS - Techninis sprendimas	0	0	2	6	0,00%	25,00%
OPP - Organizacinio proceso efektyvumas	0	0	1	4	0,00%	20,00%
OPM - Organizacijos našumo valdymas	0	0	0	9	0,00%	0,00%



7 pav. Vertinimo pagal CMMI-DEV specifinių praktikų įverčių apibendrinimas.

Duomenys 6 paveiksliuke parodo gebėjimo profilį pagal CMMI-DEV, tačiau toks profilis nėra informatyvus, kadangi identišką profilį turėtų ir nieko bendro su informacinėmis technologijomis

neturinti organizacija. 7 paveiksluke esantys duomenys parodo, jog 5 DevOps brandos lygio organizacija iš tiesų vykdo su kūrimu susijusias praktikas, daugiau nei pusėje CMMI-DEV proceso sričių yra specifinės praktikos, kurios yra pilnai arba didžiąja dalimi įgyvendintos.

Vienas iš šio vertinimo tikslų, buvo įsitikinti, jog naujai pasiūlytame DevOps brandos modelyje netrūksta su kūrimu susijusių praktiku, kurios būtų DevOps tiesioginis rūpestis. Šiuo atveju, trūkumas būtų, jeigu dėl DevOps brandos modelyje trūkstamos kūrimo praktikos, nebūtų galima įgyvendinti kurios nors iš DevOps proceso sričių. Atlikus vertinimą pagal CMMI-DEV [CMD10], kūrimo modelį, nebuvo nustatyta trūkumų, susijusių su DevOps, o tai parodo, jog DevOps kūrimo dalį naujai siūlomas DevOps brandos modelis padengia.

## 6.2. Gebėjimo profilis pagal CMMI-SVC

Šiame poskyryje nustatyta kokį gebėjimo profilį užtikrina naujai pasiūlytas DevOps brandos modelis pagal CMMI-SVC [CMS10] modelį.

2 priede, yra pateiktas vertinimas, 5 brandos lygį atitinkančios, pagal naują DevOps brandos modelį, organizacijos, pagal CMMI-SVC [CMS10]. Vertinimui buvo naudojamos visos CMMI-SVC [CMS10] proceso sritys, išskyrus „Tiekėjų sutarčių valdymas“, kadangi ši proceso sritis nėra susijusi su DevOps.

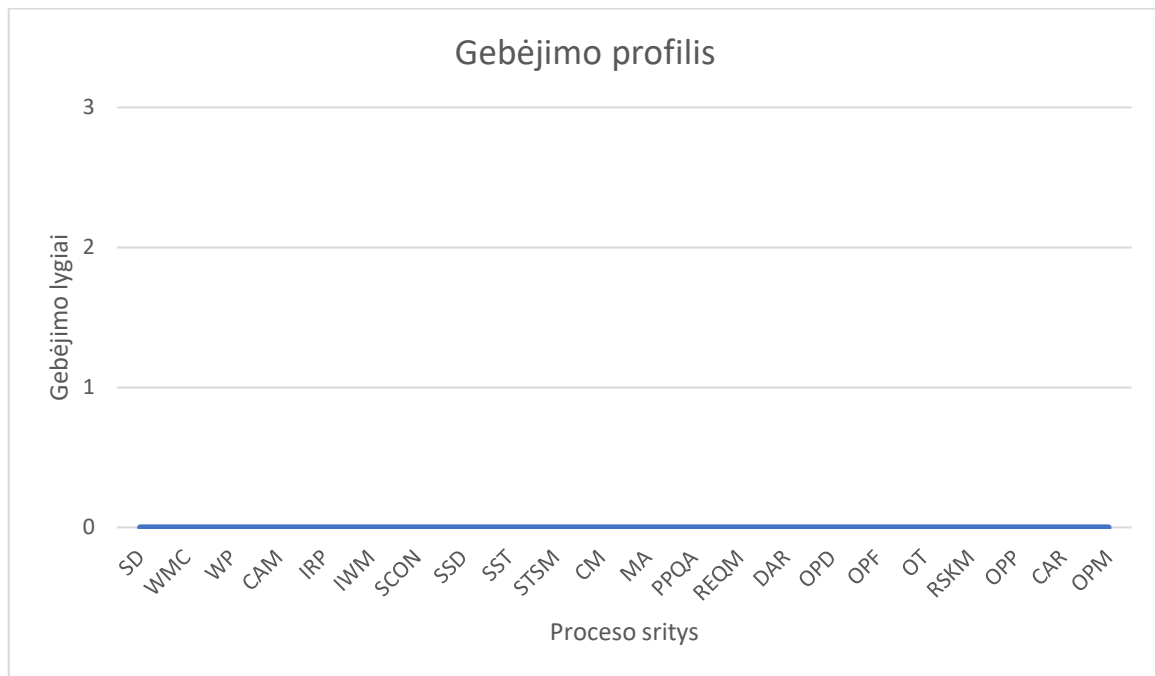
12 lentelėje yra pateiktas 2 priede atlikto vertinimo rezultatų apibendrinimas. Lentelėje naudojami trumpiniai yra tokie patys, kaip ir 11 lentelėje.

13 lentelė. 5 brandos lygio organizacijos vertinimas pagal CMMI-SVC [CMS10]

Proceso sritis	Gebėjimo lygis	SG 1	SG 2	SG 3
SD - Paslaugos teikimas	0	U	U	U
WMC - Darbo stebėjimas ir kontrolė	0	U	U	-
WP - Darbo planavimas	0	U	U	U
CAM - Pajėgumo ir prieinamumo valdymas	0	U	S	-
IRP - Incidentų sprendimas ir prevencija	0	U	U	U
IWM - Integruotas darbo valdymas	0	U	U	-
SCON - Paslaugos tęstinumas	0	U	U	-
SSD - Paslaugų sistemos kūrimas	0	U	U	U
SST - Paslaugų sistemos perėjimas	0	U	S	-
STSM - Strateginis paslaugų valdymas	0	U	U	-
CM - Konfigūracijos valdymas	0	U	S	U
MA - Matavimas ir analizė	0	U	U	-
PPQA - Proceso ir produkto kokybės užtikrinimas	0	U	U	-
REQM - Reikalavimų valdymas	0	U	-	-
DAR - Sprendimų analizė ir nutarimas	0	U	-	-
OPD - Organizacinio proceso apibrėžimas	0	U	-	-
OPF - Organizacinio proceso dėmesys	0	U	U	U
OT - Organizacinis mokymas	0	U	U	-
RSKM - Rizikos valdymas	0	U	U	U
OPP - Organizacinio proceso efektyvumas	0	U	-	-

CAR - Priežastinio ryšio analizė ir nutarimas	0	U	U	-
OPM - Organizacijos našumo valdymas	0	U	U	U

Remiantis 13 lentelės duomenimis 8 paveiksliuke yra pateiktas 5 brandos lygio organizacijos gebėjimo profilis pagal CMMI-SVC [CMS10].



8 pav. 5 brandos lygio organizacijos CMMI-SVC [CMS10] gebėjimo profilis

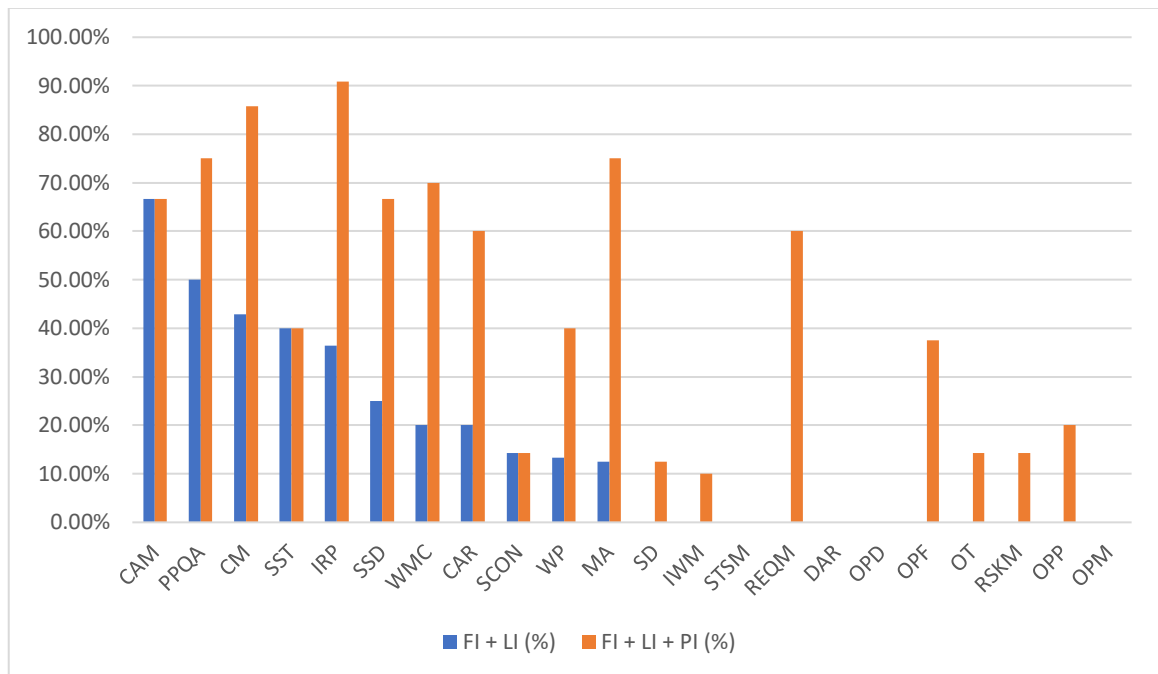
Nors vertinimo metu buvo nustatytas ne vienas įgyvendintas specifinis tikslas, tačiau nei viena proceso sritis, kaip ir CMMI-DEV vertinimo atveju, nepasiekė 1 gebėjimo lygio. CMMI-SVC [CMS10] modelis vertina daugiau proceso sričių ir ne visos jos yra pilnai tiesiogiai DevOps rūpestis, taip pat CMMI-SVC [CMS10] modelis yra griežtesnis – nurodo platesnę aibę taikytinų praktikų, su joms priskirtomis detaliosiomis praktikomis, kiekvienoje proceso srityje, nei naujas DevOps brandos modelis, dėl ko, daug specifinių praktikų yra tik dalinai įgyvendintos ir todėl 1 gebėjimo lygis nėra pasiekiamas. 14 lentelėje bei 9 paveiksliuke pateikti duomenys, kiek specifinių praktikų, kiekvienoje proceso srityje, buvo įvertintos kiekvienu įverčiu bei pateikta, kokį procentą sudaro FI ir LI (kartu) įverčiai, bei kokį – FI, LI ir PI.

14 lentelė. Vertinimo pagal CMMI-SVC specifinių praktikų įverčių pasiskirstymas.

Proceso sritys	FI	LI	PI	NI	FI + LI (%)	FI + LI + PI (%)
CAM - Pajėgumo ir prieinamumo valdymas	0	4	0	2	66,67%	66,67%
PPQA - Proceso ir produkto kokybės užtikrinimas	1	1	1	1	50,00%	75,00%
CM - Konfigūracijos valdymas	0	3	3	1	42,86%	85,71%
SST - Paslaugų sistemos perėjimas	1	1	0	3	40,00%	40,00%
IRP - Incidentų sprendimas ir prevencija	0	4	6	1	36,36%	90,91%
SSD - Paslaugų sistemos kūrimas	1	2	5	4	25,00%	66,67%
WMC - Darbo stebėjimas ir kontrolė	1	1	5	3	20,00%	70,00%



CAR - Priežastinio ryšio analizė ir nutarimas	0	1	2	2	20,00%	60,00%
SCON - Paslaugos testinumas	1	0	0	6	14,29%	14,29%
WP - Darbo planavimas	2	0	4	9	13,33%	40,00%
MA - Matavimas ir analizė	1	0	5	2	12,50%	75,00%
SD - Paslaugos teikimas	0	0	1	7	0,00%	12,50%
IWM - Integruotas darbo valdymas	0	0	1	9	0,00%	10,00%
STSM - Strateginis paslaugų valdymas	0	0	0	4	0,00%	0,00%
REQM - Reikalavimų valdymas	0	0	3	2	0,00%	60,00%
DAR - Sprendimų analizė ir nutarimas	0	0	0	6	0,00%	0,00%
OPD - Organizacinio proceso apibrėžimas	0	0	0	7	0,00%	0,00%
OPF - Organizacinio proceso dėmesys	0	0	3	5	0,00%	37,50%
OT - Organizacinis mokymas	0	0	1	6	0,00%	14,29%
RSKM - Rizikos valdymas	0	0	1	6	0,00%	14,29%
OPP - Organizacinio proceso efektyvumas	0	0	1	4	0,00%	20,00%
OPM - Organizacijos našumo valdymas	0	0	0	9	0,00%	0,00%



9 pav. Vertinimo pagal CMMI-SVC specifinių praktikų įverčių apibendrinimas.

Duomenys 8 paveiksliuke parodo gebėjimo profilį pagal CMMI-SVC, tačiau toks profilis nėra informatyvus, kadangi identišką profilį turėtų ir nieko bendro su informacinėmis technologijomis neturinti organizacija. 9 paveiksliuke esantys duomenys parodo, jog 5 DevOps brandos lygio organizacija iš tiesų vykdo su eksploatavimu susijusias praktikas, lygiai pusėje CMMI-SVC proceso sričių yra specifinės praktikos, kurios yra pilnai arba didžiąja dalimi įgyvendintos.

Vienas iš šio vertinimo tikslų, buvo įsitikinti, jog naujai pasiūlytame DevOps brandos modelyje netrūksta su eksploatavimu susijusių praktikų, kurios būtų DevOps tiesioginis rūpestis. Šiuo atveju, trūkumas būtų, jeigu dėl DevOps brandos modelyje trūkstamos eksploatavimo praktikos, nebūtų galima įgyvendinti kurios nors iš DevOps proceso sričių. Atlikus vertinimą pagal CMMI-SVC

[CMS10], eksploavimo modelį, nebuvo nustatyta trūkumų, susijusių su DevOps, o tai parodo, jog DevOps eksploavimo dalį naujai siūlomas DevOps brandos modelis padengia.

## 7. REZULTATAI IR IŠVADOS

Šiame darbe buvo išanalizuoti esami DevOps brandos modeliai, tam, kad būtų galima įsitikinti, jog šiuo metu nėra praktiškai taikomo DevOps brandos modelio, bei išanalizuotos DevOps procese taikomos praktikos, tam, kad būtų galima pasiūlyti DevOps brandos modelį, kurį būtų galima taikyti praktiškai.

Remiantis nagrinėtais DevOps brandos modeliais, CMMI brandos modeliais bei nustatytais DevOps procese taikomomis praktikomis, darbe buvo pasiūlytas naujas, praktiškai taikomas DevOps brandos modelis bei atliktas jo vertinimas, naudojantis kūrimo (CMMI-DEV [CMD10]) ir eksploataavimo (CMMI-SVC [CMS10]) modeliais, tam, kad būtų galima įsitikinti, jog naujame modelyje nėra trūkumų. Taip pat, remiantis atlikto vertinimo rezultatais buvo nustatyta, kokius gebėjimo profilius pagal CMMI-DEV ir CMMI-SVC užtikrina naujas DevOps brandos modelis.

Darbo rezultatai:

1. Pasiūlytas naujas praktiškai taikomas DevOps brandos modelis.
2. Gebėjimo profilis pagal CMMI-DEV, kurį užtikrina naujai pasiūlytas DevOps brandos modelis.
3. Gebėjimo profilis pagal CMMI-SVC, kurį užtikrina naujai pasiūlytas DevOps brandos modelis.

Išvados:

1. Šiuo metu nėra DevOps brandos modelio, kurio detalumas leistų tą modelį taikyti ne tik jo autoriams.
2. Naujai pasiūlyto DevOps brandos modelio detalumas leidžia šį modelį taikyti praktiškai.
3. Naujai pasiūlytas DevOps brandos modelis užtikrina su DevOps susijusių kūrimo ir eksploataavimo dalių įgyvendinimą, kadangi vertinimo, pagal kūrimo (CMMI-DEV [CMD10]) ir eksploataavimo (CMMI-SVC [CMS10]) modelius, metu nebuvo nustatyta kritinių trūkumų susijusių su DevOps.

## ŠALTINIAI

- [Alj18] ALJUNDI, Mohamed Khaled; Tools and Practices to Enhance DevOps Core Values. [lutpub.lut.fi/handle/10024/148944](http://lutpub.lut.fi/handle/10024/148944). 2018. Žiūrėta: 2020-03-11.
- [All12] ALLSPAW, John. Blameless postmortems and a just culture. [codeascraft.com/2012/05/22/blameless-postmortems](http://codeascraft.com/2012/05/22/blameless-postmortems). 2012. Žiūrėta: 2019-12-30.
- [Bah13] BAHRS, Paul. Adopting the IBM DevOps approach for continuous software delivery, IBM. [www.ibm.com/developerworks/library/d-adoption-paths](http://www.ibm.com/developerworks/library/d-adoption-paths), 2013-10-11. Žiūrėta: 2019-12-30.
- [BCC+13] BANG, Soon; CHUNG, Sam; CHOH, Young; DUPUIS, Marc. A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for devops. In Proceedings of the 2Nd Annual Conference on Research in Information Technology, RIIT, New York, NY, USA, 2013.
- [Bea14] BEAL, Helen. Ranger4 DevOps Maturity Assessment, [www.slideshare.net/DevOpstastic/ranger4-dev-ops-maturity-assessment](http://www.slideshare.net/DevOpstastic/ranger4-dev-ops-maturity-assessment), 2014-01-09. Žiūrėta: 2019-12-30.
- [BK17] BUCENA, Ineta; Kirikova, Marite. Simplifying the DevOps Adoption Process. [ceur-ws.org/Vol-1898/paper14.pdf](http://ceur-ws.org/Vol-1898/paper14.pdf). 2017. Žiūrėta: 2019-12-30.
- [Bto13] BTopham. DevOps and OpsDev: How Maturity Model Works. [www.docdroid.net/eY2HWPI/links-2013-agile-lifecycle-management-and.pdf](http://www.docdroid.net/eY2HWPI/links-2013-agile-lifecycle-management-and.pdf). 2013-04-26. Žiūrėta: 2019-12-30.
- [Cep05] CEPEDA, Sandra; CMMI - Staged or Continuous? [resources.sei.cmu.edu/asset\\_files/Presentation/2005\\_017\\_001\\_22691.pdf](http://resources.sei.cmu.edu/asset_files/Presentation/2005_017_001_22691.pdf). 2005. Žiūrėta: 2019-12-30.
- [CMA14] CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.3b. CMMI Institute, [cmmiinstitute.com/resources/standard-cmmi-appraisal-method-process-improvement-scampi-version-13b-method-definition](http://cmmiinstitute.com/resources/standard-cmmi-appraisal-method-process-improvement-scampi-version-13b-method-definition). 2014. Žiūrėta: 2020-04-07.
- [CMD10] CMMI for Development, Version 1.3. Carnegie Mellon University, [resources.sei.cmu.edu/asset\\_files/TechnicalReport/2010\\_005\\_001\\_15287.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf), 2010. Žiūrėta: 2019-12-30.
- [CMS10] CMMI for Services, Version 1.3. Carnegie Mellon University, [resources.sei.cmu.edu/asset\\_files/TechnicalReport/2010\\_005\\_001\\_15290.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15290.pdf), 2010. Žiūrėta: 2019-12-30.

- [Cuk13] CUKIER, Daniel. DevOps Patterns to Scale Web Applications using Cloud Services. [www.researchgate.net/publication/262404654\\_DevOps\\_patterns\\_to\\_scale\\_web\\_applications\\_using\\_cloud\\_services](http://www.researchgate.net/publication/262404654_DevOps_patterns_to_scale_web_applications_using_cloud_services). 2013. Žiūrėta: 2019-12-30.
- [DD16] DAVIS, Jennifer; DANIELS, Ryn. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale. USA, 2016. ISBN 978-1-491-92630-7.
- [DPY+19] DIAZ, Jessica; PEREZ, Jorge; YAGUE, Agustin ir kt. DevOps in Practice – A preliminary Analysis of two Multinational Companies. [arxiv.org/pdf/1910.07223.pdf](http://arxiv.org/pdf/1910.07223.pdf). 2019-09-23. Žiūrėta: 2019-12-30.
- [FBB+99] FOWLER, Martin, BECK, Kent; BRANT, John; OPDYKE, William; ROBERTS, Don. Refactoring: Improving the Design of Existing Code. Addison Wesley, 2nd edition. USA, 1999. ISBN 978-0134757599. [ieeexplore.ieee.org/abstract/document/1038431](http://ieeexplore.ieee.org/abstract/document/1038431). 2002. Žiūrėta: 2019-12-30.
- [FF14] FARROHA, Bassam; FARROHA, Deborah. A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment. [ieeexplore.ieee.org/abstract/document/6956773](http://ieeexplore.ieee.org/abstract/document/6956773). 2014. Žiūrėta: 2020-04-11.
- [Gee15] GEERLING, Jeff. Ansible for DevOps. First edition. USA, 2015. ISBN 978-0-9863934-0-2.
- [Htt12] HTERMANN Michael. Devops for Developers. Apress, Berkely, CA, USA, 2012. ISBN 978-1430245698.
- [JAP+16] JABBARI, Ramtin; ALI, Nauman; PETERSEN, Kai; TANVEER, Binish; What is DevOps? A Systematic Mapping Study on Definitions and Practices. [dl.acm.org/doi/abs/10.1145/2962695.2962707](http://dl.acm.org/doi/abs/10.1145/2962695.2962707). Association for Computing Machinery, New York, NY, USA, 2016. Žiūrėta: 2020-03-04.
- [JS05] JANZEN, David; SAIEDIAN, Hossein. Test-driven development concepts, taxonomy, and future direction. [ieeexplore.ieee.org/abstract/document/1510569](http://ieeexplore.ieee.org/abstract/document/1510569). 2005. Žiūrėta: 2019-12-30.
- [KHD+16] KIM, Gene; HUBMLE Jez; DEBOIS Patrick; WILLIS John. The DevOps Handbook. First edition. USA, 2016. ISBN 978-1942788003.
- [KS18] KONIG, Leon; STEFFENS, Andreas. Design Metric Indicator to Improve Quality Software Development. Towards a Quality Model for DevOps. [www.sciencedirect.com/science/article/pii/S1877050918315217](http://www.sciencedirect.com/science/article/pii/S1877050918315217). Aachen, Germany, 2018, p. 37. Žiūrėta: 2019-12-30.

- [Lwa17] LWAKATARE, Lucy Ellen. DevOps Adoption and Implementation in Software Development Practice. [jultika.oulu.fi/files/isbn9789526217116.pdf](http://jultika.oulu.fi/files/isbn9789526217116.pdf). University of Oulu. 2017. Žiūrėta: 2019-12-30.
- [Min14] MINICK, Eric. Continuous Delivery Maturity Model, IBM. [developer.ibm.com/urbancode/docs/continuous-delivery-maturity-model](http://developer.ibm.com/urbancode/docs/continuous-delivery-maturity-model), 2014-02-10. Žiūrėta: 2019-12-30.
- [MM15] MENZEL, Gunnar; MACAULAY, Andrew. DevOps - The Future of Application Lifecycle Automation, Capgemini. [www.capgemini.com/wp-content/uploads/2017/07/devops\\_pov\\_2015-12-18\\_final.pdf](http://www.capgemini.com/wp-content/uploads/2017/07/devops_pov_2015-12-18_final.pdf), 2015-12-18. Žiūrėta: 2019-12-30.
- [Moh15] MOHAMED, Samer. DevOps shifting software engineering strategy - value based perspective. [www.researchgate.net/publication/275224595\\_DevOps\\_shifting\\_software\\_engineering\\_strategy\\_-\\_value\\_based\\_perspective](http://www.researchgate.net/publication/275224595_DevOps_shifting_software_engineering_strategy_-_value_based_perspective), 2015. Žiūrėta: 2019-12-30.
- [NJT16] NICOLAU, Breno; JERONIMO, Helvio; TRAVASSOS, Guilherme Horta. Characterizing DevOps by Hearing Multiple Voices. [dl.acm.org/doi/10.1145/2973839.2973845](http://dl.acm.org/doi/10.1145/2973839.2973845). Maringá, Brazil, 2016. Žiūrėta: 2019-12-30.
- [OW15] OLSZEWSKA, Marta; WALDÉN, Marina. Devops meets formal modelling in high-criticality complex systems. In Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, ACM, New York, NY, USA, 2015, p. 7-12.
- [Par17] PARNIN, Chris. The Top 10 Adages in Continuous Deployment. IEEE Software, vol. 34, no. 3, May-Jun. 2017, p. 86-95.
- [RJ00] RISING, Linda; JANOFF, Norman. The Scrum Software Development Process for Small Teams. [faculty.salisbury.edu/~xswang/Research/Papers/SERelated/scrum/s4026.pdf](http://faculty.salisbury.edu/~xswang/Research/Papers/SERelated/scrum/s4026.pdf). 2000. Žiūrėta: 2019-12-30.
- [RW16] RAHMAN, Akond; WILLIAMS, Laurie. Security Practices in DevOps. [dl.acm.org/doi/10.1145/2898375.2898383](http://dl.acm.org/doi/10.1145/2898375.2898383). 2016. Žiūrėta: 2019-12-30.
- [SBO18] SENAPATHI, Mali; BUCHAN, Jim; OSMAN, Hady. DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. [www.researchgate.net/publication/326029173\\_DevOps\\_Capabilities\\_Practices\\_and\\_Challenges\\_Insights\\_from\\_a\\_Case\\_Study](http://www.researchgate.net/publication/326029173_DevOps_Capabilities_Practices_and_Challenges_Insights_from_a_Case_Study). Auckland, New Zealand, 2018. Žiūrėta: 2019-12-30.

- [Sol16] Solinea. The Solinea DevOps Maturity Model. [solinea.com/blog/solinea-devops-maturity-model](https://solinea.com/blog/solinea-devops-maturity-model), 2016-10-04. Žiūrėta: 2019-12-30.
- [Sta18] Statista. Extent of DevOps adoption by software developers worldwide in 2017 and 2018, Sauce Labs. [www.statista.com/statistics/673505/worldwide-software-development-survey-devops-adoption](https://www.statista.com/statistics/673505/worldwide-software-development-survey-devops-adoption). 2018. Žiūrėta: 2019-12-30.
- [Sto15] STORMS, Andrew. How Security can be the Next Force Multiplier in DevOps, RSAConference2015. [www.slideshare.net/AndrewStorms/how-security-can-be-the-next-force-multiplier-in-devops](https://www.slideshare.net/AndrewStorms/how-security-can-be-the-next-force-multiplier-in-devops). 2015. Žiūrėta: 2019-12-30.
- [Ver16] VERONA, Joakim; Practical DevOps. Birmingham, UK, 2016. ISBN 978-1-78588-287-6.
- [Vir15] VIRMANI, Manish. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. Fifth International Conference on the Innovative Computing Technology (INTECH 2015), Galcia, 2015, p. 78-82.
- [WBL14] WETTINGER, Johannes; BREITENBÜCHER, Uwe; LEYMANN, Frank. DevOpSlang – Bridging the Gap between Development and Operations. Service-Oriented and Cloud Computing: Third European Conference, ESOC 2014, Manchester, UK, September 2-4, 2014. Proceedings, p. 108-122.
- [WEH15] WALLER, Jan; EHMKE, Nils; HASSELBRING, Wilhelm. Including Performance Benchmarks into Continuous Integration to Enable DevOps. [www.researchgate.net/publication/274738961\\_Including\\_Performance\\_Benchmarks\\_into\\_Continuous\\_Integration\\_to\\_Enable\\_DevOps](https://www.researchgate.net/publication/274738961_Including_Performance_Benchmarks_into_Continuous_Integration_to_Enable_DevOps). 2015. Žiūrėta: 2019-12-30.
- [Wie18] WIEDEMANN, Anna. IT Governance Mechanisms for DevOps Teams – How Incumbent Companies Achieve Competitive Advantages. [scholarspace.manoa.hawaii.edu/bitstream/10125/50506/paper0619.pdf](https://scholarspace.manoa.hawaii.edu/bitstream/10125/50506/paper0619.pdf). 2018. Žiūrėta: 2019-12-30.
- [WZ06] WARD, Peter; ZHOU, Honggeng. Impact of information technology integration and lean/just-in-time practices on lead-time performance. 2006, 37(2), p. 177-203.

## PRIEDAI

### 1 priedas. Vertinimas pagal CMMI-DEV

Šiame priede pateikiamas 5 brandos lygio organizacijos pagal šiame darbe sukurtą DevOps brandos modelį vertinimas pagal CMMI-DEV brandos modelį. Vertinant prie įrodymų nurodoma praktika ir brandos lygis, todėl norint gauti žemesnio brandos lygio vertinimą, tereikia atmesti aukštesnio lygio įrodymus.

Specifiniai tikslai numeruojami pridedant priekyje SG, pavyzdžiui SG 1, SG 2 ir t.t.

Specifinės praktikos numeruojamos pridedant priekyje SP, o pirmas skaičius nurodo specifinio tikslo numerį, pavyzdžiui, SP 1.1, reiškia, jog tai yra pirmą specifinę praktiką susieta su pirmu specifiniu tikslu (SG 1).

Proceso sritis:		CM - Configuration Management	
Specifinės praktikos pagal tikslą	Įvertinimas (NI, PI, LI, FI)	Įrodymai	Trūkumai
SP 1.1 - Identify Configuration Items	PI	Kokybė (2 brandos lygis) – kodo versijų kontroliavimas. Identifikuojamas konfigūravimo elementas - kodo versijos. Nurodoma, kad kiekviena kodo versija saugoma organizacijos kodo saugykloje, kiekvienas pakeitimas susiejamas su pakeitimą atlikusiu darbuotoju.	Konfigūracijos elementams nesuteikiami unikalūs identifikatoriai, neapibrėžiamos jų svarbiosios savybės, nėra identifikuojami ryšiai tarp konfigūracijos elementų.
SP 1.2 - Establish a Configuration Management System	PI	Kokybė (2 brandos lygis) – kodo prieigos kontroliavimas. Tik leidimą turintys darbuotojai gali peržiūrėti programų sistemos kodą. Kokybė (2 brandos lygis) – kodo versijų kontroliavimas. Naudojantis kodo versijavimo įrankiu, galima saugoti ir peržiūrėti programų sistemų kodą.	Nėra sukurto mechanizmo, suteikiančio kelių lygių konfigūracijos valdymo galimybes. Konfigūracijos valdymo ataskaitos nėra kuriamos.
SP 1.3 - Create or Release Baselines	PI	Diegimas (2 brandos lygis) - automatizuotas programų sistemos diegimas	Organizacijoje CCB nėra sudaryta, bazinė konfigūracija nedokumentuojama
SP 2.1 - Track Change Requests	LI	Valdymas (2 brandos lygis) - reikalavimų analizavimas, esamos iteracijos reikalavimų sąrašas. Nauji reikalavimai patenka į reikalavimų sąrašą, reikalavimai peržiūrimi, analizuojami ir prioretizuojami. Iteracijos progresas yra stebimas ir rezultatas pristatomas.	Reikalavimai nėra skirstomi į kategorijas.
SP 2.2 - Control Configuration Items	LI	Kokybė (2 brandos lygis) – kodo versijų kontroliavimas.	Neregistruojamos pokyčių priežastys.
SP 3.1 - Establish Configuration	LI	Kokybė (2 brandos lygis) – kodo versijų kontroliavimas.	Nėra identifikuojamos konfigūracijos elementų versijos, naudojamos



Management Records		Kodo versijavimo sistemoje galima pažūrėti kodo versijų istoriją, kodo pokyčius tarp vienas po kito einančių versijų. Diegimas (5 brandos lygis) - nuolatinis diegimas į gamybos aplinką. Naujausia bazinė versija visada yra "master" kodo šakoje.	konkrečioje bazinėje versijoje. Nėra užtikrinama, kad visi suinteresuoti asmenys turėtų prieigą prie konfigūracijos bei konfigūruojamų elementų.
SP 3.2 - Perform Configuration Audits	NI	-	Nėra imamasi veiksmų, jog būtų užtikrinta, kad konfigūracijos valdymo įrašai teisingai identifikuotų konfigūracijos elementus. Konfigūracijos valdymo sistemos elementų struktūra bei integralumas nėra peržiūrimi. Nėra imamasi veiksmų, jog būtų užtikrinamas konfigūracijos valdymo sistemos elementų pilnumas, neprieštarinumas bei nuoseklumas. Nėra užtikrinamas suderinamumas su aktualiais konfigūracijos valdymo standartais ar procedūromis.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Baselines	Neįgyvendintas		
SG 2 - Track and Control Changes	Įgyvendintas		
SG 3 - Establish Integrity	Neįgyvendintas		

<b>Proceso sritis:</b>		MA - Measurement and Analysis	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Measurement Objectives	PI	Kokybė (4 brandos lygis) – informacijos poreikių dokumentavimas. Reikalinga informacija, skirta programų sistemoms gerinti, yra dokumentuojami.	Informacijos poreikiai nėra prioretizuojami. Matavimų tikslai nėra dokumentuojami.
SP 1.2 - Specify Measures	PI	Kokybė (4 brandos lygis) – dokumentuotų informacijos poreikių analizavimas. Nustatomos reikiamos metrikos, reikiamos gauti dokumentuotai reikiamai informacijai. Renkama informacija susiejama su metrikomis.	Nėra identifikuojamos esamos metrikos, kurios pateiktų norimą informaciją, metrikos nėra prioretizuojamos, peržiūrimos ar atnaujinamos.
SP 1.3 - Specify Data Collection and Storage Procedures	PI	Kokybė (4 brandos lygis) - informacijos rinkimas. Palaikomas automatinis duomenų rinkimas, kai tai yra įmanoma.	Nėra specifikuojama kaip turi būti surenkamos metrikos ir kaip jos saugomos. Metrikų surinkimo procedūros nėra prioretizuojamos, peržiūrimos ar atnaujinamos.
SP 1.4 - Specify Analysis Procedures	NI	-	Analizės procedūros nėra specifikuojamos
SP 2.1 - Obtain Measurement Data	PI	Kokybė (4 brandos lygis) – informacijos rinkimas, naudotojų elgsenos sistemoje stebėjimas.	Duomenų integralumo patikros nėra atliekamos.

SP 2.2 - Analyze Measurement Data	PI	Kokybė (5 brandos lygis) - matavimo rezultatų apdorojimas. Surinkti matavimų rezultatai yra apdorojami ir paruošiami į lengvai suprantamą ir interpretuojamą formatą, pavyzdžiui, lentelės, grafikai ir t.t. Kokybė (5 brandos lygis) - matavimo rezultatų aptarimas. Suinteresuoti asmenys peržiūri ir aptaria gautus rezultatus.	Nesudaromi preliminarūs rezultatai. Matavimo kriterijai nėra atnaujinami ateities analizėms.
SP 2.3 - Store Data and Results	NI	-	Duomenų saugojimo procedūros nėra apibrėžtos.
SP 2.4 - Communicate Results	FI	Kokybė (5 brandos lygis) - matavimo rezultatų aptarimas. Suinteresuoti asmenys informuojami apie matavimų rezultatus. Dalinimasis (2 brandos lygis) – greita tiesioginė komunikacija. Suinteresuotiems asmenims greitai pranešama apie esančius matavimo rezultatus. Kokybė (5 brandos lygis) - matavimo rezultatų apdorojimas. Suinteresuotiems asmenims padedama suprasti matavimo duomenis juos apdorojant ir paruošiant.	
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Align Measurement and Analysis Activities	Neįgyvendintas		
SG 2 - Provide Measurement Results	Neįgyvendintas		

<b>Proceso sritis:</b>		PMC - Project Monitoring and Control	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Monitor Project Planning Parameters	PI	Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas. Valdymas (4 brandos lygis) - iteracijų atlikto darbo kiekio stebėjimas. Periodiškai matuojama, kiek iš tikrųjų buvo panaudota pastangų. Valdymas (5 brandos lygis) - iteracijų peržiūros. Nustatomas nuokrypis nuo projekto plane numatytų sąnaudų.	Projekto kaina nėra stebima. Nėra stebimi projekto personalo žinios ir įgūdžiai Nėra dokumentuojami dideli nuokrypiai nuo projekto planavimo parametrų. Nėra stebimi, skiriami ir naudojami resursai.
SP 1.2 - Monitor Commitments	LI	Valdymas (2 brandos lygis) - produkto šeimininko rolė. Procesą prižiūri projekto šeimininkas Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas komandos viduje.	Nėra identifikuojami įsipareigojimai, kurie su didele tikimybe, kad bus neįgyvendinti.

		Valdymas (5 brandos lygis) - iteracijų peržiūros. Identifikuojami įsipareigojimai kurie nebuvo patenkinti.	
SP 1.3 - Monitor Project Risks	NI	-	Nėra periodiškai peržiūrima rizikų dokumentacija atsižvelgiant į dabartinį projekto statusą ir aplinkybes. Nėra atnaujinama rizikų dokumentacija, kai atsiranda papildoma informacija. Nėra iškomunikuojama rizikos statusas atitinkamiems suinteresuotiems asmenims.
SP 1.4 - Monitor Data Management	NI	-	Nėra peržiūrimos duomenų valdymo veiklos. Nėra nustatomos ir dokumentuojamos svarbios problemos ir jų poveikiai. Nėra dokumentuojami duomenų valdymo veiklos peržiūrų rezultatai.
SP 1.5 - Monitor Stakeholder Involvement	NI	-	Nėra peržiūrimas suinteresuoto asmens dalyvavimo statusas. Nėra nustatomos ir dokumentuojamos svarbios problemos ir jų poveikiai. Nėra dokumentuojami suinteresuoto asmens dalyvavimo peržiūrų rezultatai.
SP 1.6 - Conduct Progress Reviews	LI	Kokybė (5 brandos lygis) - matavimo rezultatų aptarimas. Matavimo rezultatus peržiūri produkto šeimininkas. Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas komandos viduje. Produkto šeimininkas peržiūri ir aptaria rezultatus. Valdymas (3 brandos lygis) - iteracijos rezultatų pristatymas organizacijoje. Vykdomas pristatymas, kurio metu parodomas sukurtas funkcionalumas visiems organizacijoje. Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas. Projekto reikalavimų pakeitimai stebimi iki jų uždarymo.	Nėra nustatomos ir dokumentuojamos svarbios problemos ir nuokrypiai nuo plano. Nėra dokumentuojami peržiūrų rezultatai.
SP 1.7 - Conduct Milestone Reviews	PI	Valdymas (3 brandos lygis) - iteracijos rezultatų pristatymas organizacijoje.	Nėra peržiūrimi įsipareigojimai, projekto planas, statusas ir projekto rizikos. Nėra nustatomos ir dokumentuojamos svarbios problemos ir jų poveikiai. Nėra dokumentuojami peržiūrų, veiksmų ir nuosprendžių rezultatai.
SP 2.1 - Analyze Issues	NI	-	Nėra renkamos problemos analizei Nėra analizuojamos problemos
SP 2.2 - Take Corrective Action	NI	-	Nėra nusprendžiama ir dokumentuojama, kokių reikią tinkamų veiksmų nustatytoms problemoms. Nėra peržiūrima ir sutariama su tinkamais suinteresuotais asmenimis dėl veiksmų atlikimo. Nėra derinami pokyčiai į vidinius ar išorinius įsipareigojimus.

SP 2.3 - Manage Corrective Actions	NI	-	Nėra stebimi taisymo veiksmai iki jų užbaigimo. Nėra analizuojami taisymo veiksmų rezultatai siekiant nustatyti jų efektyvumą. Nėra nustatomi ir dokumentuojami taisymo veiksmai siekiant sutaisyti nuokrypius nuo planuotų rezultatų vykdant taisymo veiksmus.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Monitor the Project Against the Plan	Neįgyvendintas		
SG 2 - Manage Corrective Action to Closure	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>PP - Project Planning</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Estimate the Scope of the Project	PI	Valdymas (3 brandos lygis) - bendras organizacijos reikalavimų sąrašas. Nurodo aukšto lygio reikiamų atlikti darbų struktūrą. Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas, reikalavimų analizavimas.	Nėra apibrėžtas produktų ar produkto komponentų nustatymas įsigijimui. Nėra nustatomi darbo produktai, kurie bus pernaudojami.
SP 1.2 - Establish Estimates of Work Product and Task Attributes	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas. Įvertinami darbo produktų ir užduočių atributai.	Nėra nustatomi techniniai sprendimai projektui įgyvendinti. Nėra naudojami atitinkami metodai darbo produkto atributams nustatyti.
SP 1.3 - Define Project Lifecycle Phases	FI	Valdymas (2 brandos lygis) - darbas iteracijomis.	-
SP 1.4 - Estimate Effort and Cost	LI	Matuojamas (4 brandos lygis) - iteracijų atlikto darbo kiekio stebėjimas. Renkami duomenis, skirti geriau įsivertinti darbą ateityje.	Nėra konkrečiai nurodoma, kad nustatant kaštus yra atsižvelgiama į infrastruktūrą.
SP 2.1 - Establish the Budget and Schedule	PI	Valdymas (3 brandos lygis) - iteracijos rezultatų pristatymas organizacijoje. Iteracijos gale pristatomas naujas funkcionalumas. Valdymas (2 brandos lygis) - reikalavimų analizavimas, iteracijos reikalavimų sąrašas. Sudaromas preliminarus darbo grafikas.	Nėra apibrėžta, ar nustatomi sąryšiai tarp užduočių. Nėra palaikomas biudžetas ir darbo grafikas projekto eigoje. Nėra nustatomų taisymo veiksmų kriterijai.
SP 2.2 - Identify Project Risks	NI	-	Nėra nustatomos rizikos. Nėra dokumentuojamos rizikos. Nėra peržiūrima ir sutariama su suinteresuotais asmenimis dėl dokumentuotų rizikų išbaigtumo ir korektiškumo. Nėra patikrinama, ar rizikos yra tinkamos.

SP 2.3 - Plan Data Management	NI	-	Nėra nustatomi reikalavimai ir procedūros siekiant užtikrinti duomenų privatumą ir apsaugą. Nėra nustatytų mechanizmų skirtų archyvuoti duomenis ir pasiekti archyvuotus duomenis. Nėra nusprendžiama, kokie projekto duomenys bus nustatomi, surenkami ir paskirstomi. Nėra nustatomi reikalavimai siekiant suteikti prieigą prie duomenų bei juos paskirstyti susijusiems suinteresuotiems asmenims. Nėra nusprendžiama kokiems projekto duomenims ir planams reikia versijavimo sistemos.
SP 2.4 - Plan the Project's Resources	PI	Dalinimasis (2 brandos lygis) - rašytinė komunikacija, greita tiesioginė komunikacija. Dalinimasis (3 brandos lygis) - bendra kalba rašytinei komunikacijai.	Nėra nustatomi procesai, skirti projekto valdymui. Nėra apibrėžiami personalo reikalavimai. Nėra nustatomi įrangos, ir komponentų reikalavimai. Nėra nustatomi kitų išteklių resursų reikalavimų (pvz. elektra, ofiso reikmenys).
SP 2.5 - Plan Needed Knowledge and Skills	PI	Kultūra (3 brandos lygis) - nuolatinis mokymasis. Dalinimasis (2 brandos lygis) - techninių ir kitų profesinių žinių nuolatinis skleidimas.	Nėra nustatoma, kokių žinių ir įgūdžių reikia norint atlikti projektą. Nėra įvertinama, kokios yra pasiekiamos žinios ir įgūdžiai organizacijoje. Nėra įtrauktos parinktos priemonės į projekto planą.
SP 2.6 - Plan Stakeholder Involvement	NI	-	Nėra sudaromas suinteresuotų asmenų įtraukimo planas
SP 2.7 - Establish the Project Plan	PI	Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas, darbas iteracijomis, rezultatų pristatymas. Valdymas (5 brandos lygis) - iteracijų peržiūros, retrospektyvos.	Projekto plane nėra apibrėžiami: rizikos nustatymas, duomenų valdymas, resursų ir įgūdžių reikalavimai
SP 3.1 - Review Plans that Affect the Project	NI	-	Nėra peržiūrimi visi planai, kurie turi įtakos projektui, norint suprasti projekto įsipareigojimus.
SP 3.2 - Reconcile Work and Resource Levels	NI	-	Nėra suderinamas projekto planas, kad būtų galima suderinti laisvus ir numatytus resursus.
SP 3.3 - Obtain Plan Commitment	NI	-	Nėra nustatomas reikiama pagalba ir nėra tariamasi dėl įsipareigojimų su produkto šeimininku. Nėra dokumentuojami visi organizaciniai įsipareigojimai. Nėra peržiūrima ar vidiniai įsipareigojimai yra atitinka vadovybės lūkesčius.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Estimates	Neįgyvendintas		
SG 2 - Develop a Project Plan	Neįgyvendintas		

SG 3 - Obtain Commitment to the Plan	Neįgyvendintas
--------------------------------------	----------------

<b>Proceso sritis:</b>		PPQA - Process and Product Quality Assurance	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Objectively Evaluate Processes	LI	Dalinimasis (3 brandos lygis) - atvira komunikacija. Darbuotojams suteikiama galimybė ir yra skatinami pranešti apie matomas problemas vykdomame procese. Valdymas (5 brandos lygis) - retrospektyvos. Identifikuojamos išmoktos pamokos, kurios galėtų pagerinti procesus.	Nenaudojami kriterijai vertinti procesus, nes jie neapibrėžti. Nėra numatomi kriterijai proceso vertinimui.
SP 1.2 - Objectively Evaluate Work Products	PI	Kokybė (2 brandos lygis) - pagrindinio funkcionalumo integracinis testavimas. Darbo produktai vertinami prieš išleidžiant naują versiją.	Nėra atsirenkami produktai, kurie bus vertinami remiantis dokumentacija. Nėra nustatomi ir palaikomi aiškiai nurodyti vertinimo kriterijai pasirinktų darbo produktų vertinimui.
SP 2.1 - Communicate and Resolve Noncompliance Issues	NI	-	Nėra dokumentuojami neatitikimai, kurių neįmanoma išspręsti. Nėra eskaluojami neatitikimai, kurių neįmanoma išspręsti. Nėra analizuojami neatitikimai tam, kad būtų pastebėtos kokybės tendencijos. Nėra užtikrinama, kad suinteresuotieji sužinotų apie įvertinimo ir kokybės tendencijų rezultatus.
SP 2.2 - Establish Records.	FI	Kokybė (4 brandos lygis) - nesėkmingų automatinių testų kiekybinis stebėjimas. Automatinių testų rezultatai yra saugomi ir matomi.	-
<b>Specifinis tikslas</b>	<b>Įvertinimas (įgyvendintas, neįgyvendintas)</b>		
SG 1 - Objectively Evaluate Processes and Work Products	Neįgyvendintas		
SG 2 - Provide Objective Insight	Neįgyvendintas		

<b>Proceso sritis:</b>		REQM - Requirements Management	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Understand Requirements	PI	Valdymas (2 brandos lygis) - produkto šeimininko rolė. Reikalavimų teikėjas yra produkto šeimininkas.	Nėra nustatyti kriterijai, skirti įvertinti ir priimti reikalavimams.
SP 1.2 - Obtain Commitment to Requirements	NI	-	Nėra vertinama naujų reikalavimų įtaka esamiems įsipareigojimams. Nėra įrašomi ir svarstomi įsipareigojimai.

SP 1.3 - Manage Requirements Changes	PI	Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas. Valdymas (3 brandos lygis) - bendras organizacijos reikalavimų sąrašas. Reikalavimai dokumentuoti ir saugomi reikalavimų sąrašuose. Dalinimasis (2 brandos lygis) - projekto informacijos nuolatinis skleidimas. Projekto reikalavimai ir pasikeitimai nuolatos matomi.	Nėra sudaroma reikalavimų pakeitimų istorija. Nėra nustatoma reikalavimų pakeitimų įtaka. Nėra apibrėžta, kaip reikalavimų pakeitimai pakoreguos projekto planą (kainą, apimtį, trukmę).
SP 1.4 - Maintain Bidirectional Traceability of Requirements	NI	-	Nėra žemesnio ir aukštesnio lygio reikalavimų atsekamumo. Nėra reikalavimų ir žemesnio lygio reikalavimų atsekamumo. Nėra kuriama atsekamumo matrica.
SP 1.5 - Ensure Alignment Between Project Work and Requirements	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas komandos viduje. Kiekvienos iteracijos pabaigoje yra vykdomas per iteraciją sukurtu funkcionalumo pristatymas. Tokiu būdu produkto savininkas peržiūri funkcionalumo atitikimą reikalavimams.	Pateikus naujų reikalavimų ar pakoregavus esamus, nėra peržiūrimas projekto planas. Nėra nustatoma, kaip nauji reikalavimai pakoreguos projekto planą (kainą, apimtį, trukmę).
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Manage Requirements	Neįgyvendintas		

<b>Proceso sritis:</b>	DAR - Decision Analysis and Resolution		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Guidelines for Decision Analysis	NI	-	Nėra nustatomos gairės, kada reikia naudoti oficialų vertinimo procesą.
SP 1.2 - Establish Evaluation Criteria	NI	-	Kriterijai nėra apibrėžiami.
SP 1.3 - Identify Alternative Solutions	NI	-	Alternatyvos nėra identifikuojamos.
SP 1.4 - Select Evaluation Methods	NI	-	Negalima pasirinkti vertinimo metodo, kadangi jų nėra.
SP 1.5 - Evaluate Alternative Solutions	NI	-	Alternatyvūs sprendimai nėra formaliai vertinami.
SP 1.6 - Select Solutions	NI	-	Negalima pasirinkti sprendimo, kadangi nėra atliktas vertinimas.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Evaluate Alternatives	Neįgyvendintas		

<b>Proceso sritis:</b>	IPM - Integrated Project Management		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>

SP 1.1 - Establish the Project's Defined Process	PI	Organizacijoje naudojamas apibrėžtas procesas. Valdomas (2 brandos lygis) - reikalavimų analizavimas, darbas iteracijomis, iteracijų rezultatų pristatymas. Valdomas (5 brandos lygis) - iteracijų peržiūros, retrospektyvos.	Negalima pasirinkti proceso projektui, procesas projektui nėra dokumentuojamas.
SP 1.2 - Use Organizational Process Assets for Planning Project Activities	NI	-	Organizacijos matavimų duomenų bazė nėra naudojama įvertinti projekto planavimo parametrus.
SP 1.3 - Establish the Project's Work Environment	NI	-	Darbo aplinka atskirai projektui nėra planuojama ir kuriama. Nėra suteikiamas nuolatinis projekto darbo aplinkos palaikymas. Periodiškai neperžiūrima kaip gerai darbo aplinka atitinka projekto poreikius.
SP 1.4 - Integrate Plans	PI	Valdymas (2 brandos lygis) - produkto šeimininko rolė. Produkto šeimininkas užtikrina, kad projekto planai būtų suderinami suinteresuotų asmenų planais. Valdymas (3 brandos lygis) - bendras organizacijos reikalavimų sąrašas. Integracija tarp skirtingų projektų. Kokybė (2 brandos lygis) - kodo peržiūros. Dalinimasis (2 brandos lygis) - techninių ir kitų profesinių žinių nuolatinis sklaidimas. Darbuotojai kuriems reikia pagalbos dirbant prie projekto ją gauna iš didesnę patirtį turinčių kolegų.	Nėra identifikuojama kaip bus sprendžiami konfliktai tarp suinteresuotų šalių. Nėra projekto apmokymų planų. Į projekto planą nėra įtraukiama projekto valdymo priemonių ir matavimų apibrėžimai.
SP 1.5 - Manage the Project Using the Integrated Plans	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas.	Nėra organizacijos procesų bibliotekos. Nėra periodiškai peržiūrimi Nėra periodiškai peržiūrimi ir suderinami projekto rezultatai su esamais ir numatomais organizacijos, kliento ir galutinių vartotojų poreikiais, tikslais ir reikalavimais.
SP 1.6 - Establish Teams	NI	-	Komandų struktūra nėra dokumentuota.
SP 1.7 - Contribute to Organizational Process Assets	PI	Dalinimasis (3 brandos lygis) - atvira komunikacija, galimybė darbuotojams pasiūlyti kaip galima būtų gerinti procesą. Kultūra (2 brandos lygis) - peržiūros po incidentų be kaltinimų. Dokumentuojamos išmoktos pamokos. Valdymas (5 brandos lygis) - iteracijų peržiūros. Analizuojama kas išmokta iteracijos metu ir kaip kita iteracija galėtų būti geresnė.	Nėra matavimų saugojimo duomenų bazės. Nėra organizacijos procesų išteklių bibliotekos.



SP 2.1 - Manage Stakeholder Involvement	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Suinteresuoti asmenys gali įsitikinti, kad reikalavimai įgyvendinti tinkamai arba išsakyti pastabas.	Projekto plane nėra identifikuojami suinteresuoti asmenys. Nėra koordinuojami veiksmai kaip išspręsti su reikalavimais susijusius nesusipratimus, kai reikalavimas įgyvendintas netinkamai.
SP 2.2 - Manage Dependencies	NI	-	Priklausomybės nėra valdomos.
SP 2.3 - Resolve Coordination Issues	NI	-	Problemų su suinteresuotais asmenimis sprendimas nėra koordinuotas.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Use the Project's Defined Process	Neįgyvendintas		
SG 2 - Coordinate and Collaborate with Relevant Stakeholders	Neįgyvendintas		

<b>Proceso sritis:</b>		OPD - Organizational Process Definition	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Standard Processes	NI	-	Nėra praktikų skirtų nustatyti standartiniams procesams.
SP 1.2 - Establish Lifecycle Model Descriptions	NI	-	Nėra sudaromi gyvenimo ciklo modelio aprašai.
SP 1.3 - Establish Tailoring Criteria and Guidelines	NI	-	Nėra nustatomi proceso pritaikymo kriterijai ir gairės.
SP 1.4 - Establish the Organization's Measurement Repository	NI	-	Nėra organizacijos matavimų saugyklos.
SP 1.5 - Establish the Organization's Process Asset Library	NI	-	Nėra organizacijos proceso išteklių bibliotekos.
SP 1.6 - Establish Work Environment Standards	NI	-	Nėra svarstomi komerciškai prieinami darbo aplinkos standartai. Nėra kuriami nauji darbo aplinkos standartai.
SP 1.7 - Establish Rules and Guidelines for Teams	NI	-	Nėra komandų taisyklių ir gairių.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Organizational Process Assets	Neįgyvendintas		

<b>Proceso sritis:</b>		OPF - Organizational Process Focus	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Organizational Process Needs	NI	-	Nėra nurodoma politika, standartai ir verslo tikslai, kurie yra taikomi organizacijos procesams.

			Nėra nagrinėjami susiję procesų standartai siekiant išsiaiškinti gerąsias praktikas. Nėra nustatomi organizacijos proceso efektyvumo tikslai. Nėra apibrėžiami esminiai organizacijos procesų požymiai. Nėra dokumentuojami organizacijos procesų poreikiai ir tikslai.
SP 1.2 - Appraise the Organization's Processes	PI	Valdymas (5 brandos lygis) - retrospektyvos, vykdomos pagal iš anksto numatytą tvarkaraštį, vertinamas procesas.	Nėra gaunamas palaikymas iš vyresniosios vadovybės. Nėra apibrėžiama proceso vertinimo apimtis. Proceso vertinimas ir jo išvados nėra dokumentuojami.
SP 1.3 - Identify the Organization's Process Improvements	PI	Valdymas (5 brandos lygis) - retrospektyvos, nustatoma kas blogai su procesu ir kaip tai galėtų būti gerinama.	Galimi proceso pagerinimai nėra prioretizuojami. Identifikuoti proceso pagerinimai nėra dokumentuojami. Nėra planuojamų proceso pagerinimų sąrašo.
SP 2.1 - Establish Process Action Plans	NI	-	Nėra nurodomos strategijos, metodai ir veiksmai, susiję su nustatytais proceso patobulinimais. Nėra sudaromos proceso gerinimo veiksmų komandos veiksmams įgyvendinti. Nėra dokumentuojami proceso gerinimo veiksmų planai. Proceso gerinimo veiksmų planai nėra peržiūrimi ir derinami su interesuotais asmenimis.
SP 2.2 - Implement Process Action Plans	NI	-	Nėra proceso gerinimo planų.
SP 3.1 - Deploy Organizational Process Assets	NI	-	Nėra diegiami organizacijos proceso ištekliai visoje organizacijoje. Nėra dokumentuojami organizacijos proceso išteklių pokyčiai. Nėra teikiamos konsultacijos kaip naudotis organizacijos procesų ištekliais.
SP 3.2 - Deploy Standard Processes	PI	Kokybė (3 brandos lygis) - organizacijos kokybės standartizavimas. Valdymas (3 brandos lygis) - organizacijos valdymo standartizavimas. Dalinimasis (3 brandos lygis) - organizacijos dalinimosi standartizavimas. Diegimas (3 brandos lygis) - organizacijos diegimo standartizavimas. Kultūra (3 brandos lygis) - organizacijos kultūros standartizavimas.	Nėra identifikuojami prasidedantys projektai. Nėra identifikuojami esami projektai kuriems būtų naudinga įdiegti standartą. Nėra padedama projektams prisitaikyti standartą pagal jų poreikius.
SP 3.3 - Monitor the Implementation	NI	-	Nėra stebimas standartinių procesų įgyvendinimas.

SP 3.4 - Incorporate Experiences into Organizational Process Assets	PI	Valdymas (5 brandos lygis) - retrospektyvos, periodiškai peržiūrimas organizacijos standartinių procesų veiksmingumas ir tinkamumas. Gaunamas atgalinis ryšys. Dalinimasis (3 brandos lygis) - atvira komunikacija, gaunamas atgalinis ryšys apie organizacijoje vykdomus procesus. Apie išmuktos pamokas diskutuojama ir dalinamasi atvirai.	Nėra proceso užtikrinančio, kad geriausi metodai, įrankiai ir procesai būtų prieinami organizacijos darbuotojams. Proceso gerinimo pasiūlymai nėra valdomi.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Determine Process Improvement Opportunities	Neįgyvendintas		
SG 2 - Plan and Implement Process Improvements	Neįgyvendintas		
SG 3 - Deploy Organizational Process Assets and Incorporate Experiences	Neįgyvendintas		

<b>Proceso sritis:</b>		OT - Organizational Training	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Strategic Training Needs	NI	-	Strateginiai apmokymų tikslai nėra nustatomi.
SP 1.2 - Determine Which Training Needs Are the Responsibility of the Organization	NI	-	Nėra nustatomi kurie apmokymų tikslai yra organizacijos atsakomybė.
SP 1.3 - Establish an Organizational Training Tactical Plan	NI	-	Nėra sudaromas organizacijos apmokymų taktinis planas.
SP 1.4 - Establish a Training Capability	NI	-	Nėra nustatomas apmokymų pajėgumas.
SP 2.1 - Deliver Training	PI	Dalinimasis (2 brandos lygis) - techninių ir kitų profesinių žinių nuolatinis skleidimas. Darbuotojai turintys daugiau žinių, tomis žiniomis dalinasi su mažiau žinių turinčiais darbuotojais. Skiriamas darbo laikas padėti, dirbama kartu.	Nėra sudaromas mokymo planas. Nėra stebimas mokymo plano įgyvendinimas.
SP 2.2 - Establish Training Records	NI	-	Nėra sudaromi apmokymų įrašai.

SP 2.3 - Assess Training Effectiveness	NI	-	Nėra vertinamas apmokymų efektyvumas.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Establish an Organizational Training Capability	Neįgyvendintas		
SG 2 - Provide Training	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>PI - Product Integration</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish an Integration Strategy	PI	Kokybė (2 brandos lygis) - pagrindinio funkcionalumo integracinis testavimas. Identifikuojami pagrindiniai komponentai.	Nėra identifikuojamos alternatyvios komponentų integracijos strategijos. Kadangi nėra identifikuojamos alternatyvos, pasirinkimo tarp integracijos strategijų nėra. Integracijos strategijos nėra peržiūrimos. Pagrindimai priimtiems sprendimams nėra dokumentuojami.
SP 1.2 - Establish the Product Integration Environment	PI	Identifikuojamos procedūros ir kriterijai programų sistemos aplinkai: Diegimas (2 brandos lygis) - automatizuotas programinės įrangos aplinkų kūrimas, skirtingos aplinkos gamybai ir testavimui.	Nėra identifikuojami programų sistemos aplinkos reikalavimai. Nėra sprendžiama ar pirkti ar sukurti programų sistemos aplinką patiems. Nėra procedūros skirtos išmesti nereikalingas aplinkos dalis.
SP 1.3 - Establish Product Integration Procedures and Criteria	LI	Diegimas (2 brandos lygis) - automatizuotas kodo surinkimas. Kokybė (2 brandos lygis) - pagrindinių modulių testavimas, pagrindinio funkcionalumo integracinis testavimas. Kiekvieno ciklo metu atliekamas testavimas, naujas funkcionalumas integruojamas tik jeigu jis pereiną visus testavimo atvejus.	Integracijos kriterijai nėra dokumentuojami.
SP 2.1 - Review Interface Descriptions for Completeness	NI	-	Interfeisų duomenys nėra peržiūrimi. Komponentai ir interfeisai nėra žymimi. Nėra stebimi interfeisų aprašų pasikeitimai.
SP 2.2 - Manage Interfaces	NI	-	Sąsajos nėra valdomos
SP 3.1 - Confirm Readiness of Product Components for Integration	PI	Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas. Kai funkcionalumas įgyvendinamas, užduotis pažymima kaip atlikta. Kokybė (2 brandos lygis) - pagrindinių modulių testavimas, pagrindinio funkcionalumo integracinis testavimas. Naujas funkcionalumas negali būti integruotas kol nepraeina testavimo scenarijų.	Produktų komponentų gavimas nėra patvirtinamas. Komponentai nėra aprašomi, todėl nėra ir tikrinama, ar jie atitinka savo aprašą. Nėra tikrinami konfigūracijų statusai. Komponentų interfeisai prieš jų jungimą tikrinami nėra.

SP 3.2 - Assemble Product Components	PI	Kokybė (2 brandos lygis) - atskiros testavimo, kūrimo ir gamybos aplinkos. Aplinkos yra identiškos, todėl jeigu atlikus testavimą programų sistema veikia, ji veiks ir gamybos aplinkoje.	Prieš integravimą yra atliekamas testavimas, tačiau integravimo kriterijai ir strategija nėra apibrėžtos. Integracijos strategijos ir kriterijų nėra. Procedūros nėra peržiūrimos.
SP 3.3 - Evaluate Assembled Product Components	NI	-	Vertinimas nėra atliekamas. Nėra dokumentuojami vertinimo rezultatai.
SP 3.4 - Package and Deliver the Product or Product Component	PI	Diegimas (2 brandos lygis) - automatizuotas programų sistemos diegimas. Diegimas (5 brandos lygis) - nuolatinis diegimas į gamybos aplinką.	Nėra ieškoma efektyviausių būdų sukurtam produktui pristatyti. Reikalavimai produkto pristatymui nėra apibrėžiami.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Product Integration	Neįgyvendintas		
SG 2 - Ensure Interface Compatibility	Neįgyvendintas		
SG 3 - Assemble Product Components and Deliver the Product	Neįgyvendintas		

<b>Proceso sritis:</b>		RD - Requirements Development	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Elicit Needs	FI	Valdymas (2 brandos lygis) - produkto šeimininko rolė. Reikalavimai yra gaunami tiesiogiai ir pastoviai bendraujant su produkto šeimininku. Kadangi produkto savininkas atstovauja visus galutinius naudotojus, yra pakankama reikalavimus rinkti tik iš produkto šeimininko.	-
SP 1.2 - Transform Stakeholder Needs into Customer Requirements	FI	Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas. Reikalavimai laikomi reikalavimų sąrašė nuo svarbiausio iki mažiausiai svarbaus. Valdymas (2 brandos lygis) - reikalavimų analizavimas. Reikalavimai detalai išsiaiškinami ir įdedami į iteracijos reikalavimų sąrašą. Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Rezultatai pristatomi, juos peržiūri produkto šeimininkas. Kokybė (3 brandos lygis) - naujo funkcionalumo	-

		integracinis testavimas, naujų modulių testavimas. Naujam funkcionalumui turi būti parašyti testavimo scenarijai ir jie turi veikti.	
SP 2.1 - Establish Product and Product Component Requirements	NI	-	Nėra parengiami techniniai reikalavimai, būtini programų sistemos komponentų projektavimui. Nėra pridedami nauji reikalavimai atlikus galimo įgyvendinimo pasirinkimą. Nėra išskiriami reikalavimai ir papildomas reikalavimų sąrašas atlikus sistemos projektavimą. Nėra nustatomas ryšys tarp reikalavimų.
SP 2.2 - Allocate Product Component Requirements	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas, esamos iteracijos reikalavimų sąrašas. Reikalavimai yra analizuojami ir priskiriami iteracijai.	Reikalavimai nėra grupuojami pagal funkcijas. Reikalavimai nėra priskiriami komponentams ir architektūriniais elementams. Reikalavimai nėra grupuojami pagal architektūrinius komponentus. Ryšiai nėra dokumentuojami ryšiai tarp reikalavimų.
SP 2.3 - Identify Interface Requirements	NI	-	Sąsajų reikalavimai nėra identifikuojami.
SP 3.1 - Establish Operational Concepts and Scenarios	NI	-	Nėra kuriami veikimo scenarijai. Nėra atsižvelgiama į aplinkos, kurioje veiks kuriamas produktas apribojimus ir kitas sąlygas. Nėra daromi detalūs veikimo scenarijai.
SP 3.2 - Establish a Definition of Required Functionality and Quality Attributes	PI	Kokybė (4 brandos lygis) - naudotojo elgsenos sistemoje stebėjimas. Analizuojama ir kiekybiškai vertinama naudotojų funkcionalumo poreikiai.	Nėra nustatoma misija. Nėra nustatomi svarbiausieji architektūriniai kokybės atributai, remiantis misija. Reikalavimai nėra skaidomi į loginius ir funkcinis vienetus.
SP 3.3 - Analyze Requirements	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas.	Nėra nustatoma ar reikalavimai atitinka produkto savininko lūkesčius atliekant reikalavimų analizę. Nėra nustatoma, ar reikalavimai yra pilni, įgyvendinami ir verifikuojami. Nėra išskiriami svarbiausi reikalavimai. Nėra identifikuojama metrikos, kurios bus matuojamos.
SP 3.4 - Analyze Requirements to Achieve Balance	NI	-	Nėra naudojami įrodymo metodai, prototipai, simuliacijos, kad būtų nustatytas reikalavimų įgyvendinamumas, esant apribojimams. Nėra vertinama rizika. Nėra nustatoma, kokią riziką reikalavimų įgyvendinamumui turės projekto gyvavimo ciklas. Nėra nustatoma, kokią įtaką darys nefunkciniai reikalavimai projekto kainai ir kokia bus rizika.
SP 3.5 - Validate Requirements	NI	-	Nėra nustatoma rizika, kad galutinis produktas netinkamai veiks gamybinėje aplinkoje atliekant reikalavimų analizę. Nėra tikrinamas reikalavimų pilnumas.

		Nėra vertinamas įgyvendinimas
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>	
SG 1 - Develop Customer Requirements	Įgyvendintas	
SG 2 - Develop Product Requirements	Neįgyvendintas	
SG 3 - Analyze and Validate Requirements	Neįgyvendintas	

<b>Proceso sritis:</b>		<b>RSKM - Risk Management</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Determine Risk Sources and Categories	NI	-	Nėra proceso skirto nustatyti rizikoms. Nėra proceso skirto nustatyti rizikų kategorijoms.
SP 1.2 - Define Risk Parameters	NI	-	Nėra apibrėžiami kriterijai nustatyti rizikos tikimybei. Nėra apibrėžiamos kiekvienos rizikų kategorijos ribos. Nėra apibrėžtos ribos, pagal kurias slenksčiai taikomi kategorijai ar jos viduje.
SP 1.3 - Establish a Risk Management Strategy	NI	-	Projektų rizikos valdymo strategija nėra sudaroma.
SP 2.1 - Identify Risks	PI	Kokybė (2 brandos lygis) - programų sistemos testavimas, neištestuotas kodas negali patekti į gamybos aplinką, identifikuojama potencialiai nekorektiškai veikiančios programų sistemos rizika. Kokybė (5 brandos lygis) - apsaugos priemonių nuo potencialių kenkėjiškų atakų taikymas. Identifikuojama rizika dėl kenkėjiškų atakų.	Nėra nustatomos rizikos susijusios su kaina, planų ir našumu. Nėra nustatomi aplinkos elementai, kurie gali paveikti projektą. Nėra dokumentuojamas kontekstas, sąlygos ir potencialios pasekmės kiekvienai rizikai. Nėra identifikuojami suinteresuoti asmenys susiję su kiekviena rizika.
SP 2.2 - Evaluate, Categorize, and Prioritize Risks	NI	-	Rizikos nėra vertinamos, kategorizuojamos ir prioretizuojamos.
SP 3.1 - Develop Risk Mitigation Plans	NI	-	Nėra kuriamas rizikų šalinimo planas.
SP 3.2 - Implement Risk Mitigation Plans	NI	-	Nėra rizikų šalinimo plano.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Risk Management	Neįgyvendintas		
SG 2 - Identify and Analyze Risks	Neįgyvendintas		
SG 3 - Mitigate Risks	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>TS - Technical Solution</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Develop Alternative Solutions and Selection Criteria	NI	-	Nėra ruošiami alternatyvūs sprendimai ir pasirinkimo kriterijai.
SP 1.2 - Select Product Component Solutions	NI	-	Nėra pasirenkami produkto komponentų sprendimai.
SP 2.1 - Design the Product or Product Component	NI	-	Nėra projektuojamas produktas ar produkto komponentas.
SP 2.2 - Establish a Technical Data Package	NI	-	Nėra kuriamas išsamus produkto ar gaminio komponento aprašymas.
SP 2.3 - Design Interfaces Using Criteria	NI	-	Sąsajos nėra kuriamos naudojantis iš anksto nustatytais kriterijais.
SP 2.4 - Perform Make, Buy or Reuse Analyses	NI	-	Nėra atliekama analizė.
SP 3.1 - Implement the Design	PI	Kokybė (2 brandos lygis) - kodo peržiūros. Kokybė (3 brandos lygis) - privalomas naujų modulių testavimas.	Nėra apibrėžti metodai kaip turi būti kuriami produkto komponentai. Nėra užtikrinama, kad būtų laikomasi kūrimo standartų.
SP 3.2 - Develop Product Support Documentation	PI	Dalinimasis (3 brandos lygis) - dokumentacija. Sudaroma dokumentacija.	Dokumentacija nėra peržiūrima kolegų. Nėra naudojami apibrėžti metodai dokumentacijos sudarymui. Nėra sudaroma preliminarinė dokumentacijos versija.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Select Product Component Solutions	Neįgyvendintas		
SG 2 - Develop the Design	Neįgyvendintas		
SG 3 - Implement the Product Design	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>VAL - Validation</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Select Products for Validation	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Validuojamas naujas funkcionalumas įgyvendintas iteracijos metu. Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Validuojamas naujas funkcionalumas atliekant integracinį testavimą.	Nėra nustatoma kokios kategorijos naudotojų poreikiai yra validuojami. Nėra pasirenkamas validavimo metodas produktui ar komponentui. Nėra peržiūrimas validavimo metodo pasirinkimas su suinteresuotais asmenimis.



SP 1.2 - Establish the Validation Environment	PI	Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Identifikuojama, kad turi būti testuojamas naujas funkcionalumas Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas.	Nėra identifikuojami klientų tiekiami produktai. Nėra identifikuojama testavimo įranga. Nėra identifikuojami validavimo išteklių, kuriuos būtų galima panaudoti pakartotinai. Nėra detalai planuojamas išteklių prieinamumas.
SP 1.3 - Establish Validation Procedures and Criteria	PI	Dalinimasis (3 brandos lygis) - dokumentacija. Programų sistemos funkcionalumas dokumentuotas, pagal dokumentaciją galima tikrinti ar funkcionalumas įgyvendintas teisingai.	Produkto reikalavimai nėra peržiūrimi. Programų sistemos projekcija nėra peržiūrima.
SP 2.1 - Perform Validation	FI	Valdymas (2 brandos lygis) - produkto šeimininko rolė. Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Produkto šeimininkas peržiūri įgyvendintą funkcionalumą, gali pateikti pastabas ir grąžinti patikslintus reikalavimus į reikalavimų sąrašą kitai iteracijai. Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas. Validuojamas funkcionalumas automatiniais testais.	-
SP 2.2 - Analyze Validation Results	LI	Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas. Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Lyginami realūs rezultatai su tuo ko tikimasi. Testavimo scenarijai identifikuoja programų sistemos dalis kurios neveikia taip kaip turėtų. Produkto šeimininkas identifikuoja programų sistemos dalis kurios neveikia taip kaip turėtų. Produkto šeimininkas nurodo kaip defektai turėtų būti pataisyti.	Programų sistemos vykdymo techninės specifikos nėra tikrinamos.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Validation	Neįgyvendintas		

SG 2 - Validate Product or Product Components	Igyvendintas
-----------------------------------------------	--------------

Proceso sritis:		VER - Verification	
Specifinės praktikos pagal tikslą	Įvertinimas (NI, PI, LI, FI)	Įrodymai	Trūkumai
SP 1.1 - Select Work Products for Verification	LI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas. Verifikuojamas yra kiekvienas naujas funkcionalumas. Valdymas (2 brandos lygis) - reikalavimų analizavimas. Identifikuojami reikalavimai kurie turi būti įgyvendinti.	Nėra apibrėžiami verifikavimo metodai kiekvienam darbo produktui atskirai.
SP 1.2 - Establish the Verification Environment	NI	-	Nėra identifikuojami reikalavimai verifikavimo aplinkai. Nėra identifikuojami verifikavimo ištekčiai. Nėra identifikuojami verifikavimo įrankiai. Nėra įsigijami verifikavimą gerinantys įrankiai ar aplinka.
SP 1.3 - Establish Verification Procedures and Criteria	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas. Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas.	Nėra verifikavimo kriterijų kūrimo procedūros. Nėra identifikuojama reikalinga įranga verifikavimui.
SP 2.1 - Prepare for Peer Reviews	NI	-	Nėra ruošiamasi kolegų peržiūroms.
SP 2.2 - Conduct Peer Reviews	FI	Kokybė (2 brandos lygis) - kodo peržiūros. Kodo peržiūros atliekamos kodo versijavimo sistemoje, surašant trūkumus ir pastebėjimus. Kodo peržiūra yra baigiama kai darbuotojas atliekantis peržiūrą patvirtina, kad potencialių problemų nemato.	-
SP 2.3 - Analyze Peer Review Data	NI	-	Kolegų peržiūrų duomenys nėra analizuojami.
SP 3.1 - Perform Verification	LI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Produkto šeimininkas peržiūri įgyvendintą funkcionalumą, gali pateikti pastabas ir	Nėra identifikuojami gauti rezultatai darbo produkto verifikacijos metu. Nėra dokumentuojamas "as-run" verifikacijos metodas ir nukrypimai nuo procedūrų.

		<p>grąžinti patikslintus reikalavimus į reikalavimų sąrašą kitai iteracijai.</p> <p>Kokybė (3 brandos lygis) - naujo funkcionalumo integracinis testavimas.</p> <p>Kokybė (3 brandos lygis) - naujo funkcionalumo modulių testavimas.</p> <p>Validuojamas funkcionalumas automatiniais testais.</p> <p>Kokybė (2 brandos lygis) - testavime identifikuotų problemų registravimas.</p> <p>Valdymas (2 brandos lygis) - peržiūroje identifikuotų problemų registravimas.</p>	
SP 3.2 - Analyze Verification Results	PI	<p>Kokybė (2 brandos lygis) - testavime identifikuotų problemų registravimas.</p> <p>Valdymas (2 brandos lygis) - peržiūroje identifikuotų problemų registravimas.</p> <p>Baigus testavimą ir peržiūrą sukuriama naujos užduotys ir įdedamos į reikalavimų sąrašą.</p>	<p>Nėra lyginami esami rezultatai su norimais rezultatais.</p> <p>Nėra paruošiama rezultatų ataskaita.</p> <p>Verifikacijos rezultatai nėra naudojami faktiniams matavimams ir techninės eksploatacijos parametrų palyginti.</p> <p>Nėra suteikiama informacija kaip defektas gali būti išspręstas ir nėra inicijuojami teisingi veiksmai.</p>
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Verification	Neįgyvendintas		
SG 2 - Perform Peer Reviews	Neįgyvendintas		
SG 3 - Verify Selected Work Products	Neįgyvendintas		

<b>Proceso sritis:</b>		OPP - Organizational Process Performance	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish Quality and Process Performance Objectives	NI	-	<p>Nėra peržiūrimi organizacijos verslo tikslai, susiję su kokybe ir proceso našumu.</p> <p>Nėra apibrėžiami kiekybiniai organizacijos kokybės ir proceso rezultatų tikslai.</p> <p>Tikslai nėra prioretizuojami.</p> <p>Nėra gaunami suinteresuotų šalių įsipareigojimai organizacijos kokybės ir proceso rezultatų tikslams.</p>
SP 1.2 - Select Processes	NI	-	<p>Nėra sudaromi kriterijais pagal kuriuos būtų pasirenkami subprocesai.</p> <p>Nėra pasirenkami subprocesai ir dokumentuojamos priežastys kodėl jie buvo pasirinkti.</p> <p>Nėra sąsajų tarp pasirinktų subprocesų ir kokybės bei proceso našumo tikslų.</p> <p>Pasirinkimai nėra peržiūrimi.</p>
SP 1.3 - Establish Process Performance Measures	NI	-	<p>Nėra pasirenkamos metrikos, atitinkančios pasirinktų subprocesų atributus.</p>

			Nėra sukuriami pasirinktų priemonių apibrėžimai. Pasirinktas priemonės nėra įtraukiamos į organizacijos bendrųjų priemonių rinkinį.
SP 1.4 - Analyze Process Performance and Establish Process Performance Baselines	PI	Renkami duomenys apie proceso našumą: Kokybė (4 brandos lygis) - nesėkmingų automatinių testų kiekybinis stebėjimas. Kokybė (4 brandos lygis) - incidentų kiekio gamybinėje aplinkoje stebėjimas. Kokybė (4 brandos lygis) - programų sistemos eksploatavimo metrikų stebėjimas. Kokybė (4 brandos lygis) - laiko nuo incidento atsiradimo iki jo panaikinimo stebėjimas. Duomenys paskelbiami ir visiems matomi: Dalinimasis (3 brandos lygis) - organizacijos renkamų metrikų komunikavimas.	Nėra proceso skirta analizuoti surinktus duomenis. Nėra nustatomos procesų našumo bazinės linijos. Nėra peržiūrimos nustatytos bazinės linijos. Nėra gaunamas sutikimas dėl bazinių linijų iš suinteresuotų asmenų.
SP 1.5 - Establish Process Performance Models	NI	-	Nėra kuriami procesų našumo modeliai.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Performance Baselines and Models	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>QPM - Quantitative Project Management</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish the Project's Objectives	NI	Kokybė (4 brandos lygis) - informacijos poreikių dokumentavimas. Kokybė (4 brandos lygis) - dokumentuotų informacijos poreikių analizavimas.	Organizacijos tikslai susiję su kokybe ir procesų našumu nėra peržiūrimi. Nėra identifikuojami kokybės ir proceso našumo poreikiai ir prioritetai. Nėra nustatomos rizikos. Nėra sprendžiami konfliktai tarp skirtingų projektų.
SP 1.2 - Compose the Defined Processes	NI	-	Nėra nustatomi kriterijai, kuriuos reikia naudoti vertinant projekto alternatyvas. Nėra identifikuojami alternatyvūs projekto procesai ir papildomi procesai.
SP 1.3 - Select Subprocesses and Attributes	NI	-	Nėra pasirenkami subprocesai ir jų atributai.
SP 1.4 - Select Measures and Analytic Techniques	NI	Kokybė (4 brandos lygis) - informacijos poreikių dokumentavimas. Pagal projekto poreikius yra nustatoma kokių duomenų reikia ir kokios metrikos turėtų būti stebimos.	Nėra identifikuojamos bendrosios metrikos iš organizacijos procesų aprašų. Nėra identifikuojamos papildomos metrikos. Nėra identifikuojamos metrikos, kurios būtų naudojamos valdant subprocesus. Nėra nurodomos operatyvinės priemonių apibrėžtys.

			Nėra analizuojamas sąryšis tarp metrikų ir projekto kokybės ir procesų našumo tikslų. Nėra peržiūrimos metrikos ir statistinei analizei naudojamos technikos.
SP 2.1 - Monitor the Performance of Selected Subprocesses	NI	-	Nėra subprocessų.
SP 2.2 - Manage Project Performance	PI	Stebimas proceso progresas siekiant kokybės ir proceso našumo: Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas. Valdymas (4 brandos lygis) - iteracijų atlikto darbo kiekio stebėjimas. Periodiškai peržiūrimi rezultatai ir ar atitinka išsikeltus tikslus: Valdymas (5 brandos lygis) - iteracijų peržiūra.	Nėra subprocessų, todėl jie nėra periodiškai peržiūrimi. Nėra naudojami našumo modeliai suderinti su projekto duomenimis. Nėra įgyvendinami veiksmai susiję su nustatytais trūkumais.
SP 2.3 - Perform Root Cause Analysis	LI	Diagnozuojami proceso efektyvumo trūkumai: Valdymas (5 brandos lygis) - iteracijų peržiūra. Valdymas (5 brandos lygis) - retrospektyvos. Trūkumai registruojami: Valdymas (5 brandos lygis) - identifiкуotų problemų sprendimas. Analizuojami galimi veiksmai: Valdymas (5 brandos lygis) - registruotų problemų sprendimas.	Nėra įvertinamas veiksmų poveikis antrinių procesų našumui.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Quantitative Management	Neįgyvendintas		
SG 2 - Quantitatively Manage the Project	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>CAR - Causal Analysis and Resolution</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Select Outcomes for Analysis	PI	Renkami duomenys analizei: Kokybė (4 brandos lygis) - nesėkmingų automatinį testų kiekybinis stebėjimas. Kokybė (4 brandos lygis) - incidentų kiekio gamybinėje aplinkoje stebėjimas. Kokybė (4 brandos lygis) - programų sistemos eksploatavimo metrikų stebėjimas.	Nėra proceso skirta nuspręsti kuriuos duomenis reikia analizuoti toliau. Nėra formaliai apibrėžta analizės apimtis.

SP 1.2 - Analyze Causes	LI	Kultūra (2 brandos lygis) - peržiūros po incidentų be kaltinimų. Peržiūrimos priežastys.	Rezultatai nėra grupuojami.
SP 2.1 - Implement Action Proposals	PI	Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas. Produkto šeimininkas reikiamus atlikti veiksmus suformuluoja kaip užduotis ir įdeda į projekto reikalavimų sąrašą.	Atliktinų veiksmų pasiūlymai nėra prioretizuojami. Atliktinų veiksmų pasiūlymai nėra atrenkami. Nėra kuriamas veiksmų planas. Nėra ieškoma panašių situacijų kitose procesuose.
SP 2.2 - Evaluate the Effect of Implemented Actions	NI	-	Nėra atskirai vertinamas įgyvendintų veiksmų efektas.
SP 2.3 - Record Causal Analysis Data	NI	-	Nėra įrašoma analizės priežastis.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Determine Causes of Selected Outcomes	Neįgyvendintas		
SG 2 - Address Causes of Selected Outcomes	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>OPM - Organizational Performance Management</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Maintain Business Objectives	NI	-	Verslo tikslai nėra apibrėžiami.
SP 1.2 - Analyze Process Performance Data	NI	-	Proceso našumo duomenys nėra analizuojami verslo tikslų kontekste.
SP 1.3 - Identify Potential Areas for Improvement	NI	-	Galimos tobulinimo sritys nėra nustatomos, kadangi nėra atliekama analizė.
SP 2.1 - Elicit Suggested Improvements	NI	-	Nėra atrenkamos tobulinimo sritys.
SP 2.2 - Analyze Suggested Improvements	NI	-	Nėra analizuojamos siūlomos tobulinimo sritys.
SP 2.3 - Validate Improvements	NI	-	Tobulinimai nėra validuojami.
SP 2.4 - Select and Implement Improvements for Deployment	NI	-	Tobulinimai nėra atrenkami diegimui visoje organizacijoje.
SP 3.1 - Plan the Deployment	NI	-	Nėra planuojamas diegimas.
SP 3.2 - Manage the Deployment	NI	-	Tobulinimų diegimas nėra valdomas.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Manage Business Performance	Neįgyvendintas		

SG 2 - Select Improvements	Neįgyvendintas
SG 3 - Deploy Improvements	Neįgyvendintas

## 2 priedas. Vertinimas pagal CMMI-SVC

Šiame priede pateikiamas toks pats vertinimas kaip ir 1 priede, tačiau vertinama pagal CMMI-SVC brandos modelį. Šiame priede nėra tų CMMI-SVC modelio proceso sričių vertinimų, kurie sutampa su CMMI-DEV.

<b>Proceso sritis:</b>	<b>SD - Service Delivery</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Analyze Existing Agreements and Service Data	NI	-	Nėra peržiūrėti klientų ir naudotojų poreikių duomenys. Nėra peržiūrimos paslaugų teikimo ir aptarnaujančio personalo problemos. Nėra peržiūrimos galiojančios paslaugų ir tiekėjų sutartys. Nėra analizuojama galimybė teikti prašomas paslaugas.
SP 1.2 - Establish the Service Agreement	NI	-	Paslaugų sutartis nėra sudaroma.
SP 2.1 - Establish the Service Delivery Approach	NI	-	Nėra apibrėžiami paslaugų užklausų nustatymo kriterijai. Nėra apibrėžiami paslaugų užklausų kriterijai ir kriterijai paslaugų užklausų skirstymui į kategorijas. Nėra identifikuojami vienas ar daugiau mechanizmai, pagal kuriuos naudotojai galėtų teikti paslaugų užklausas. Nėra dokumentuojamas paslaugų pristatymo būdas.
SP 2.2 - Prepare for Service System Operations	NI	-	Nėra pasiruošama aptarnavimo sistemos operacijoms.
SP 2.3 - Establish a Request Management System	NI	-	Nėra užklausų valdymo sistemos.
SP 3.1 - Receive and Process Service Requests	NI	Įrašoma informaciją apie paslaugos užklausą: Kokybė (2 brandos lygis) - sisteminio žurnalo įrašai.	Nėra užtikrinama, kad paslaugų užklausos atitiktų paslaugos sutartį. Paslaugų užklausos nėra skirstomos į kategorijas ir analizuojamos. Nėra nustatoma kokių išteklių reikia, kad paslaugos užklausa būtų išspręsta. Paslaugos užklausos statusas nėra stebimas iki kol užklausa išsprendžiama. Paslaugos užklausa nėra uždaroma, o atlikti veiksmai ir gauti rezultatai nėra įrašomi.
SP 3.2 - Operate the Service System	PI	Valdomas paslaugų teikimo saugumas:	Nėra komunikuojamas paslaugų užklausų statusas iki jų uždarymo.

		<p>Kokybė (5 brandos lygis) - apsaugos priemonių nuo potencialių kenkėjiškų atakų taikymas.</p> <p>Atliekamas sistemos aptarnavimo komponentų stebėjimas:</p> <p>Kokybė (2 brandos lygis) - sisteminio žurnalo įrašai.</p> <p>Kokybė (4 brandos lygis) - programų sistemos eksploatavimo metrikų stebėjimas.</p> <p>Renkami sistemos naudotojų atsiliepimai:</p> <p>Kokybė (4 brandos lygis) - naudotojų atsiliepimų rinkimas.</p>	<p>Paslaugų užklauskos nėra sprendžiamos pagal paslaugų sutartį.</p> <p>Naudotojų atsiliepimai nėra renkami iškart po paslaugos suteikimo.</p>
SP 3.3 - Maintain the Service System	NI	-	<p>Palaikymo užklauskos nėra peržiūrimos.</p> <p>Palaikymo užklauskos nėra prioretizuojamos.</p> <p>Nėra kuriamas planas kaip bus vykdomas palaikymas.</p> <p>Nėra siunčiami palaikymo pranešimai suinteresuotiems asmenims.</p>
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Service Agreements	Neįgyvendintas		
SG 2 - Prepare for Service Delivery	Neįgyvendintas		
SG 3 - Deliver Services	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>WMC - Work Monitoring and Control</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Monitor Work Planning Parameters	PI	<p>Valdymas (4 brandos lygis) - iteracijų atlikto darbo kiekio stebėjimas. Stebimas esamos iteracijos atliktas darbas lyginant su planuotu atlikti darbu.</p>	<p>Nėra stebima atliekamo darbo kaina.</p> <p>Nėra stebimi suteikti ir panaudoti ištekliai.</p> <p>Nėra stebimos darbo grupės žmonių žinios ir įgūdžiai.</p> <p>Nėra dokumentuojami nukrypimai nuo plano.</p>
SP 1.2 - Monitor Commitments	LI	<p>Valdymas (5 brandos lygis) - iteracijų peržiūros. Reguliariai vykdomos įsipareigojimų ir realiai atliktų darbų peržiūros. Identifikuojami neįvykdyti reikalavimai.</p>	<p>Peržiūros nėra dokumentuojamos.</p>
SP 1.3 - Monitor Risks	NI	-	<p>Nėra periodiškai peržiūrimos dokumentuotos rizikos atliekamo darbo kontekste.</p> <p>Nėra atnaujinama rizikų dokumentacija kai atsiranda papildomos informacijos.</p> <p>Nėra komunikuojamas rizikos statusas susijusiems asmenims.</p>
SP 1.4 - Monitor Data Management	NI	-	<p>Nėra periodiškai peržiūrimos duomenų valdymo veiklos pagal jų aprašymus darbo plane.</p>



			Nėra identifikuojamos reikšmingos problemos ir jų poveikis. Nėra dokumentuojami duomenų valdymo veiklos peržiūrų rezultatai.
SP 1.5 - Monitor Stakeholder Involvement	NI	-	Nėra peržiūrimas suinteresuotų šalių įsitraukimas.
SP 1.6 - Conduct Progress Reviews	PI	Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas. Vykdomas darbas yra nuolatos stebimas produkto šeimininko.	Nėra dokumentuojamos reikšmingos problemos esančios darbo plane. Nėra dokumentuojami pakeitimų prašymai. Peržiūrų rezultatai nėra dokumentuojami.
SP 1.7 - Conduct Milestone Reviews	PI	Valdymas (5 brandos lygis) - iteracijų peržiūros. Valdymas (5 brandos lygis) - retrospektyvos.	Nėra dokumentuojamos identifikuotos problemos. Nėra dokumentuojami peržiūrų rezultatai.
SP 2.1 - Analyze Issues	FI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Valdymas (5 brandos lygis) - iteracijų peržiūros. Iteracijų peržiūrų bei iteracijos rezultatų pristatymų metu surenkamas problemų sąrašas. Produkto šeimininkas analizuoja problemas ir prideda reikiamas užduotis į užduočių sąrašą, kad problemos būtų išspręstos.	-
SP 2.2 - Take Corrective Action	PI	Valdymas (2 brandos lygis) - produkto šeimininkas. Valdymas (2 brandos lygis) - projekto reikalavimų sąrašas. Produkto šeimininkas nusprendžia kokiai veiksmi turi būti atlikti ir prideda reikalavimus į projekto reikalavimų sąrašą.	Veiksmai nėra peržiūrimi kartu su kitais susijusiais asmenimis. Nėra deramasi dėl pakeitimų vidiniuose ir išoriniuose įsipareigojimuose.
SP 2.3 - Manage Corrective Actions	PI	Valdymas (4 brandos lygis) - iteracijos progreso stebėjimas. Taisomieji veiksmai patenka į iteracijos darbų sąrašą ir yra stebimi iki kol užbaigiami.	Taisomieji veiksmai nėra analizuojami, tam, kad būtų galima suprasti jų efektyvumą.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Monitor the Work Against the Plan	Neįgyvendintas		
SG 2 - Manage Corrective Action to Closure	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>WP - Work Planning</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish the Service Strategy	NI	-	Nėra identifikuojami paslaugų tikslai. Nėra identifikuojami naudotini metodai pasiekti tikslams. Nėra dokumentuojami verslo svarba. Nėra identifikuojami pagrindiniai resursų poreikiai. Nėra identifikuojami pagrindiniai suinteresuoti asmenys.

			Nėra identifikuojamos rizikos.
SP 1.2 - Estimate the Scope of the Work	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas. Valdymas (4 brandos lygis) - esamos iteracijos reikalavimų sąrašas su įverčiais. Apibrėžiamos užduotys su vertinimais.	Nėra identifikuojami darbo produktai, kuriuos būtų galima panaudoti pakartotinai. Nėra identifikuojami produktai ar komponentai, kuriuos būtų galima įsigyti iš išorės.
SP 1.3 - Establish Estimates of Work Product and Task Attributes	PI	Valdymas (4 brandos lygis) - esamos iteracijos reikalavimų sąrašas su įverčiais.	Nėra vertinami užduočių atributai.
SP 1.4 - Define Lifecycle Phases	FI	Valdymas (2 brandos lygis) - darbas iteracijomis. Darbas vykdomas iteracijomis.	-
SP 1.5 - Estimate Effort and Cost	PI	Valdymas (4 brandos lygis) - esamos iteracijos reikalavimų sąrašas su įverčiais. Vertinama kiek darbo vienetų reikia vienai užduočiai.	Kainų vertinimas nėra atliekamas.
SP 2.1 - Establish the Budget and Schedule	NI	-	Nėra sudaromas biudžetas ir viso projekto grafikas.
SP 2.2 - Identify Risks	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas. Reikalavimų analizės metu identifikuojamos rizikos.	Rizikos nėra dokumentuojamos. Rizikos nėra aptariamoms su suinteresuotais asmenimis. Rizikos nėra peržiūrimos.
SP 2.3 - Plan Data Management	NI	-	Duomenų valdymas nėra planuojamas.
SP 2.4 - Plan the Resources	NI	-	Resursai nėra planuojami.
SP 2.5 - Plan Needed Knowledge and Skills	NI	-	Reikalingos žinios ir įgūdžiai nėra planuojami.
SP 2.6 - Plan Stakeholder Involvement	NI	-	Suinteresuotų asmenų įsitraukimas nėra planuojamas.
SP 2.7 - Establish the Work Plan	FI	Valdymas (2 brandos lygis) - esamos iteracijos reikalavimų sąrašas. Esamos iteracijos reikalavimų sąrašas yra iteracijos planas. Už planą atsakingas produkto šeimininkas, jis jį ir patvirtina.	-
SP 3.1 - Review Plans That Affect the Work	NI	-	Planai kurie gali paveikti darbą nėra peržiūrimi.
SP 3.2 - Reconcile Work and Resource Levels	NI	-	Darbo ir išteklių lygiai nėra derinami.
SP 3.3 - Obtain Plan Commitment	NI	-	Nėra gaunamas įsipareigojimas planui.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Estimates	Neįgyvendintas		
SG 2 - Develop a Work Plan	Neįgyvendintas		

SG 3 - Obtain Commitment to the Plan	Neįgyvendintas
--------------------------------------	----------------

<b>Proceso sritis:</b>		CAM - Capacity and Availability Management	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish a Capacity and Availability Management Strategy	NI	-	Nėra dokumentuojami išteklių ir paslaugų naudojimas, našumas ir prieinamumas. Nėra vertinami ateities išteklių ir paslaugų pajėgumo bei prieinamumo poreikiai. Nėra kuriama pajėgumo strategija. Nėra kuriama prieinamumo strategija. Strategija nėra peržiūrima.
SP 1.2 - Select Measures and Analytic Techniques	NI	-	Nėra identifikuojamos metrikos iš organizacinio proceso išteklių, kurie palaiko pajėgumo ir prieinamumo valdymo tikslus. Nėra identifikuojamos ir specifikuojamos papildomos metrikos. Nėra analizuojamas ryšys tarp identifikuotų metrikų ir paslaugų reikalavimų.
SP 1.3 - Establish Service System Representations	LI	Kokybė (4 brandos lygis) - programų sistemos įprasto eksploatavimo slenksčių nustatymas. Kokybė (4 brandos lygis) - programų sistemos eksploatavimo slenksčių stebėjimas.	Įprasto eksploatavimo slenksčiai nėra peržiūrimi ir nėra gaunamas leidimas iš suinteresuotų šalių.
SP 2.1 - Monitor and Analyze Capacity	LI	Kokybė (4 brandos lygis) - programų sistemos eksploatavimo slenksčių stebėjimas. Stebimi paslaugos teikimui reikalingi resursai, stebimas paslaugos atsako laikas. Identifikuojama kai slenksčiai peržengiami.	Nėra nustatomi taisomieji veiksmai, kurių reikia imtis. Nėra vertinamas poreikių pasikeitimas ateityje (mažiau/daugiau išteklių).
SP 2.2 - Monitor and Analyze Availability	LI	Kokybė (4 brandos lygis) - programų sistemos pasiekiamumo stebėjimas. Identifikuojama kai programų sistema tampa nepasiekiamą ir pranešama suinteresuotiems asmenims.	Nėra analizuojamos pasiekiamumo tendencijos. Nėra nustatomi taisomieji veiksmai, kurių reikia imtis.
SP 2.3 - Report Capacity and Availability Management Data	LI	Kokybė (4 brandos lygis) - incidentų gamybinėje aplinkoje stebėjimas Kokybė (4 brandos lygis) - programų sistemos eksploatavimo slenksčių stebėjimas. Kokybė (4 brandos lygis) - programų sistemos pasiekiamumo stebėjimas.	Nėra raportuojami duomenys pagal esamus duomenis lyginant su planuojamais augimo įverčiais.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		

SG 1 - Prepare for Capacity and Availability Management	Neįgyvendintas
SG 2 - Monitor and Analyze Capacity and Availability	Įgyvendintas

<b>Proceso sritis:</b>		<b>IRP - Incident Resolution and Prevention</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish an Approach to Incident Resolution and Prevention	PI	<p>Apibrėžiami kriterijai kas yra incidentas:  Kokybė (4 brandos lygis) - programų sistemos įprasto eksploataavimo slenksčių nustatymas.  Kokybė (4 brandos lygis) - incidentų gamybinėje aplinkoje stebėjimas.  Identifikuojamas mechanizmas kaip naudotojai galėtų pranešti apie incidentus.  Kokybė (4 brandos lygis) - naudotojų atsiliepimų rinkimas.</p>	<p>Nėra incidentų kategorijų ir kriterijų pagal kuriuos incidentai būtų priskiriami kategorijai.  Nėra apibrėžiami metodai ir įsigijami įrankiai, skirti incidentams valdyti.  Nėra dokumentuojami kriterijai pagal kuriuos sprendžiama kada incidentas turi būti uždarytas.</p>
SP 1.2 - Establish an Incident Management System	PI	<p>Kokybė (2 brandos lygis) - incidentų valdymo sistema.</p>	<p>Nėra užtikrinama galimybė perkelti incidentą tarp grupių incidentų valdymo sistemoje.  Nėra užtikrinamas incidentų valdymo sistemos ir jos turinio vientisumas.</p>
SP 2.1 - Identify and Record Incidents	LI	<p>Identifikuojamas mechanizmas kaip naudotojai galėtų pranešti apie incidentus:  Kokybė (4 brandos lygis) - incidentų gamybinėje aplinkoje stebėjimas.  Įrašoma informacija apie incidentus:  Kokybė (2 brandos lygis) - testavimo metu identifikuotų problemų registravimas.  Kokybė (2 brandos lygis) - programų sistemos naudotojų identifikuotų problemų registravimas.</p>	<p>Incidentai nėra kategorizuojami.</p>
SP 2.2 - Analyze Individual Incident Data	LI	<p>Valdymas (2 brandos lygis) - produkto šeimininkas. Produkto šeimininkas prideda naują užduotį į reikalavimų sąrašą.  Valdymas (2 brandos lygis) – reikalavimų analizavimas.  Išanalizuojamas incidentas, kuris pridėtas kaip nauja užduotis.</p>	<p>Nėra sprendžiama kuri grupė geriausiai išspręstų incidentą.</p>
SP 2.3 - Resolve Incidents	LI	<p>Valdymas (2 brandos lygis) – darbas iteracijomis. Incidento sprendimas įtraukiamas į iteracijos reikalavimų sąrašą ir yra sprendžiamas.</p>	<p>Nėra atskiro darbo proceso incidentams spręsti.</p>

SP 2.4 - Monitor the Status of Incidents to Closure	PI	Kokybė (2 brandos lygis) - incidentų valdymo sistema. Fiksuojamas incidento statusas ir uždaromas, kai incidentas būna išsprendžiamas.	Incidento sprendimas nėra dokumentuojamas. Sprendimas nėra peržiūrimas ir patvirtinamas suinteresuotų šalių.
SP 2.5 - Communicate the Status of Incidents	NI	-	Incidentų statusas nėra komunikuojamas.
SP 3.1 - Analyze Selected Incidents	LI	Kultūra (2 brandos lygis) - peržiūros po incidentų be kaltinimų. Identifikuojamos incidento priežastys.	Nėra dokumentuojamos incidento priežastys.
SP 3.2 - Establish Solutions to Respond to Future Incidents	PI	Kultūra (2 brandos lygis) - peržiūros po incidentų be kaltinimų. Randami sprendimai kaip ateityje išvengti įvykusių incidentų.	Nėra dokumentuojamas pakartotinai panaudojamas sprendimas. Nėra validuojamas pakartotinai panaudojamas sprendimas. Nėra komunikuojama suinteresuotiems asmenims.
SP 3.3 - Establish and Apply Solutions to Reduce Incident Occurrence	PI	Valdymas (2 brandos lygis) - iteracijos reikalavimų sąrašas. Sprendimai įtraukiami į iteracijų reikalavimų sąrašą ir yra įgyvendinami.	Nėra sprendžiama kuri grupė darbuotojų labiausiai tinkama įgyvendinti sprendimą. Nėra dokumentuojama kokie veiksmai buvo atlikti. Atliekami veiksmai nėra nuolatos komunikuojami suinteresuotoms šalims.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Įgyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Incident Resolution and Prevention	Neįgyvendintas		
SG 2 - Identify, Control, and Address Individual Incidents	Neįgyvendintas		
SG 3 - Analyze and Address Causes and Impacts of Selected Incidents	Neįgyvendintas		

<b>Proceso sritis:</b>	<b>IWM - Integrated Work Management</b>		
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Establish the Defined Process	NI	-	Negalima pasirinkti gyvavimo ciklo modelio.
SP 1.2 - Use Organizational Process Assets for Planning Work Activities	NI	-	Nėra organizacijos matavimų saugyklos. Nėra naudojama organizacijos matavimų saugykla, siekiant įvertinti planuojamą darbą.
SP 1.3 - Establish the Work Environment	NI	-	Nėra planuojama, projektuojama ir diegiama darbo aplinka.
SP 1.4 - Integrate Plans	PI	Valdymas (3 brandos lygis) - bendras organizacijos reikalavimų sąrašas. Darbai derinami.	Neatliekami apmokymai. Neplanuojama kolegų peržiūra. Nėra identifikuojama kaip konfliktai bus šalinami.

		Diegimas (3 brandos lygis) - planuojami ir valdomi skirtingų projektų versijų išleidimai. Skirtingų projektų versijų išleidimai derinami.	
SP 1.5 - Manage the Work Using Integrated Plans	NI	-	Darbas nėra valdomas remiantis integruotais planais.
SP 1.6 - Establish Teams	NI	-	Komandos nėra sudaromos.
SP 1.7 - Contribute to Organizational Process Assets	NI	-	Nėra prisidedama prie organizacijos procesų išteklių.
SP 2.1 - Manage Stakeholder Involvement	NI	-	Suinteresuotų šalių įsitraukimas nėra valdomas.
SP 2.2 - Manage Dependencies	NI	-	Priklausomybės nėra valdomos.
SP 2.3 - Resolve Coordination Issues	NI	-	Koordinavimo problemos nėra sprendžiamos.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Use the Defined Process for the Work	Neįgyvendintas		
SG 2 - Coordinate and Collaborate with Relevant Stakeholders	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>SCON - Service Continuity</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Identify and Prioritize Essential Functions	LI	Kokybė (2 brandos lygis) - pagrindinio programų sistemos funkcijų identifikavimas. Pirmenybė yra teikiama pagrindiniam funkcionalumui, pirma reikalaujama, kad būtų testuojamas pagrindinis funkcionalumas: Kokybė (2 brandos lygis) - pagrindinių modulių testavimas.	Nėra analizuojamas poveikis teikiamoms paslaugoms, jeigu pagrindinės funkcijos nustotų veikti.
SP 1.2 - Identify and Prioritize Essential Resources	NI	-	Nėra identifikuojami ir dokumentuojami vidinės ir išorinės priklausomybės. Nėra identifikuojamas ir dokumentuojamas pagrindinis personalas ir jų rolės. Nėra identifikuojamos ir dokumentuojamos organizacijos ir suinteresuotų šalių atsakomybės. Resursai nėra prioretizuojami. Nėra užtikrinama, kad įrašai ir duomenų bazės būtų apsaugotos, prieinamos ir naudojamos kritiniu atveju.
SP 2.1 - Establish Service Continuity Plans	NI	-	Nėra kuriamas paslaugos tęstinumo planai.

SP 2.2 - Establish Service Continuity Training	NI	Dalinimasis (2 brandos lygis) - techninių ir kitų profesinių žinių nuolatinis skleidimas. Daugiau patirties turintys darbuotojai apmoko mažiau patirties turinčius darbuotojus.	Nėra kuriami paslaugos tęstinumo mokymai.
SP 2.3 - Provide and Evaluate Service Continuity Training	NI	-	Nėra vykdomi paslaugos tęstinumo mokymai.
SP 3.1 - Prepare for the Verification and Validation of the Service Continuity Plan	NI	-	Nėra paslaugos tęstinumo plano.
SP 3.2 - Verify and Validate the Service Continuity Plan	NI	-	Nėra paslaugos tęstinumo plano.
SP 3.3 - Analyze Results of Verification and Validation of the Service Continuity Plan	NI	-	Nėra paslaugos tęstinumo plano.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Identify Essential Service Dependencies	Neįgyvendintas		
SG 2 - Prepare for Service Continuity	Neįgyvendintas		
SG 3 - Verify and Validate the Service Continuity Plan	Neįgyvendintas		

<b>Proceso sritis:</b>		<b>SSD - Service System Development</b>	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Develop Stakeholder Requirements	LI	Valdymas (2 brandos lygis) - produkto šeimininkas. Produkto šeimininkas nusako reikalavimus, ko tikimasi iš programų sistemos. Reikalavimai analizuojami, konkretinami: Valdymas (2 brandos lygis) - reikalavimų analizavimas.	Nėra apibrėžiami validavimo apribojimai.
SP 1.2 - Develop Service System Requirements	PI	Valdymas (2 brandos lygis) - produkto šeimininkas. Produkto šeimininkas atsakingas už reikalavimus, jis nusako kokie turi būti reikalavimai.	Nėra sudaromi ryšiai tarp reikalavimų. Reikalavimams nėra priskiriami loginiai vienetai (pavyzdžiui, komponentai). Nėra identifikuojamos išorinės ir vidinės sąsajos. Nėra kuriami reikalavimai identifikuotoms sąsajoms.

SP 1.3 - Analyze and Validate Requirements	PI	Valdymas (2 brandos lygis) - reikalavimų analizavimas. reikalavimai yra peržiūrėti, nustatomas jų poveikis bei poreikis produktui.	Nėra nustatomi kokybės atributai reikalavimams. Reikalavimai nėra validuojami.
SP 2.1 - Select Service System Solutions	NI	-	Nėra apibrėžiami kriterijai pasirinkimui. Nėra pateikiamos alternatyvos. Neatliekamas pasirinkimas.
SP 2.2 - Develop the Design	NI	-	Nėra atliekamas projektavimas.
SP 2.3 - Ensure Interface Compatibility	NI	-	Nėra įsitikinama dėl sąsajų suderinamumo.
SP 2.4 - Implement the Service System Design	NI	-	Nėra įgyvendinama suprojektuota sistema.
SP 2.5 - Integrate Service System Components	PI	Kokybė (3 brandos lygis) - privalomas naujo funkcionalumo integracinis testavimas. Įvykdoma sistemos komponentų integracija.	Sistemos integravimo strategija nėra sudaroma. Nėra vertinamas sistemos sąsajų suderinamumas.
SP 3.1 - Prepare for Verification and Validation	LI	Identifikuojami komponentai kurie turi būti tikrinami: Kokybė (2 brandos lygis) - pagrindinių modulių testavimas. Kokybė (3 brandos lygis) - privalomas naujų modulių testavimas. Paruošiama aplinka: Kokybė (2 brandos lygis) - atskiros testavimo, kūrimo ir gamybos aplinkos.	Nėra kuriami ir palaikomi validavimo kriterijai.
SP 3.2 - Perform Peer Reviews	PI	Kokybė (2 brandos lygis) - kodo peržiūros. Atliekamos kolegų peržiūros, nustatyta kada peržiūra yra baigta.	Nėra ruošiamasi kolegų peržiūrai. Nėra sudaromas kolegų peržiūrų grafikas. Nėra kolegų peržiūrų šablono pagal kurį turi būti atliekama peržiūra.
SP 3.3 - Verify Selected Service System Components	PI	Valdymas (2 brandos lygis) - iteracijos rezultatų pristatymas. Produkto šeimininkas prižiūri kad naujas funkcionalumas atitiktų reikalavimus.	Nėra įrašomi peržiūros rezultatai. Nėra dokumentuotas peržiūros metodas.
SP 3.4 - Validate the Service System	FI	Analizuojami validavimo duomenys: Kokybė (2 brandos lygis) - testavimo metu identifikuotų problemų registravimas. Atliekamas validavimas Kokybė (3 brandos lygis) - privalomas naujų modulių testavimas. Kokybė (3 brandos lygis) - privalomas naujo funkcionalumo integracinis testavimas.	-
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Develop and Analyze Stakeholder Requirements	Neįgyvendintas		



SG 2 - Develop Service Systems	Neįgyvendintas
SG 3 - Verify and Validate Service Systems	Neįgyvendintas

<b>Proceso sritis:</b>		SST - Service System Transition	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Analyze Service System Transition Needs	NI	-	Nėra analizuojama kaip sistema veikia esamoje pristatymo aplinkoje. Nėra analizuojami sistemos komponentai kurie bus atnaujinami. Nėra identifikuojamos potencialios rizikos.
SP 1.2 - Develop Service System Transition Plans	NI	-	Nėra apibrėžiamas diegimo metodas kiekvienam specifiniam sistemos atnaujinimui. Nėra nustatoma diegimo kaina, reikalingi resursai ir vykdymo grafikas. Nėra identifikuojami suinteresuoti asmenys.
SP 1.3 - Prepare Stakeholders for Changes	NI	-	Suinteresuoti asmenys nėra paruošiami pasikeitimams.
SP 2.1 - Deploy Service System Components	FI	Paruošiama programų sistemų versija kuri turi būti diegiama: Diegimas (2 brandos lygis) - planuojami ir valdomi versijų išleidimai. Įsitikinama, kad paruošta versija veikia korektiškai. Kokybė (2 brandos lygis) - atskiros testavimo, kūrimo ir gamybos aplinkos Atliekamas programų sistemų diegimas: Diegimas (2 brandos lygis) - automatizuotas programų sistemos diegimas. Po diegimo, stebima gamybinė aplinka dėl galimų incidentų buvimo: Kokybė (4 brandos lygis) - incidentų gamybinėje aplinkoje stebėjimas. Jeigu programų sistema veikia nekorektiškai, atstatoma ankstesnė versija: Diegimas (2 brandos lygis) - galimybė atstatyti anksčiau įdiegtą versiją.	-
SP 2.2 - Assess and Control the Impacts of the Transition	LI	Naudojami duomenų surinkimo metodai gauti informacijai apie naują programų sistemos versiją: Kokybė (4 brandos lygis) - naudotojų atsiliepimų rinkimas.	Incidentas nėra kontroliuojamas iki kol diegimo problemos yra išspręstos. Nėra atliekama peržiūra po diegimo.

		Nauja versija proaktyviai komunikuojama organizacijos viduje: Valdymas (3 brandos lygis) - iteracijos rezultatų pristatymas organizacijoje. Jeigu programų sistema veikia nekorektiškai, atstatoma ankstesnė versija: Diegimas (2 brandos lygis) - galimybė atstatyti anksčiau įdiegtą versiją.	
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Prepare for Service System Transition	Neįgyvendintas		
SG 2 - Deploy the Service System	Igyvendintas		

<b>Proceso sritis:</b>		STSM - Strategic Service Management	
<b>Specifinės praktikos pagal tikslą</b>	<b>Įvertinimas (NI, PI, LI, FI)</b>	<b>Įrodymai</b>	<b>Trūkumai</b>
SP 1.1 - Gather and Analyze Data	NI	-	Nėra renkami ir analizuojami duomenys apie organizacijos galimybes. Nėra renkami ir analizuojami duomenys apie organizacijos strateginius poreikius.
SP 1.2 - Establish Plans for Standard Services	NI	-	Nėra patvirtinami strateginiai verslo tikslai. Nėra rekomenduojami reikalavimai paslaugoms remiantis strateginiais verslo tikslais, galimybėmis ir strateginiais poreikiais. Nėra identifikuojami reikalingi atlikti veiksmai.
SP 2.1 - Establish Properties of Standard Services and Service Levels	NI	-	Nėra apibrėžiami kritiniai atributai kiekvienai paslaugai. Nėra nustatomos įprastos ir kintamos standartinių paslaugų dalys. Nėra organizuojamos paslaugas pagal poreikį.
SP 2.2 - Establish Descriptions of Standard Services	NI	-	Nėra kuriami standartinių paslaugų aprašai skirti visiems suinteresuotiems asmenims.
<b>Specifinis tikslas</b>	<b>Įvertinimas (Igyvendintas, neįgyvendintas)</b>		
SG 1 - Establish Strategic Needs and Plans for Standard Services	Neįgyvendintas		
SG 2 - Establish Standard Services	Neįgyvendintas		