

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Įmonės „Mėnuliukų technologijos“ programų
kūrimo proceso aprašas (Pirmas laboratorinis
darbas)**

**Description of the development process of the „Moon
Technologies” company (First laboratory work)**

Programų kūrimo proceso laboratorinis darbas

Atliko:	4 kurso 3 grupės studentai	
	Matas Savickis, Justas Tvarijonas, Džiugas Mažulis	(parašas)
	Greta Pyrantaitė, Andrius Bentkus	(parašas)
Darbo vadovas:	Saulius Ragaišis, Doc., Dr.	(parašas)

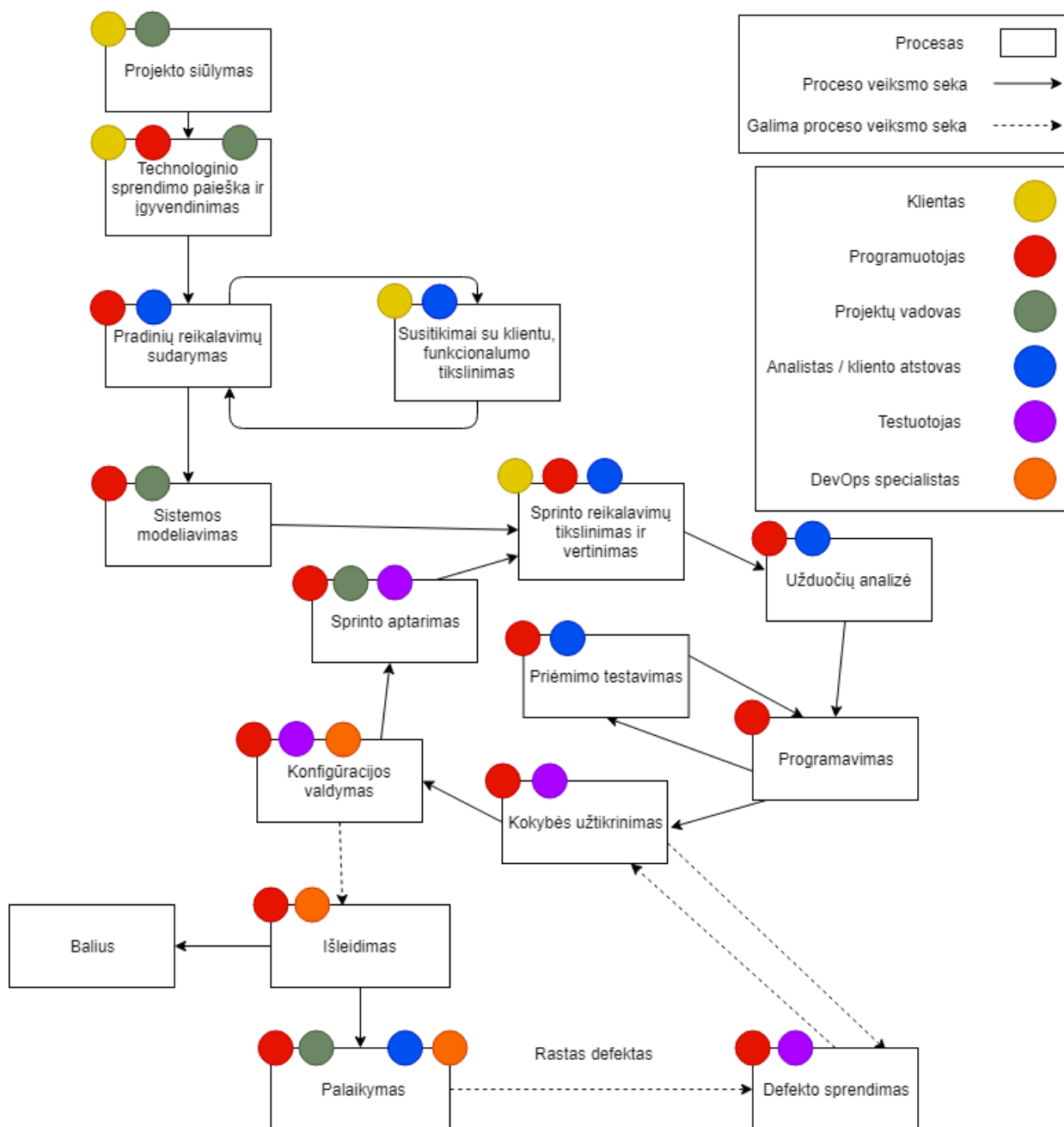
TURINYS

IVADAS	2
1. KŪRIMO PROCESAS	3
1.1. Projekto siūlymas	3
1.2. Technologinio sprendimo paieška ir įgyvendinimas.....	4
1.3. Reikalavimų ciklas	5
1.4. Sistemos modeliavimas	5
1.5. Sprintas	6
1.5.1. Sprinto užduočių tikslinimas ir vertinimas.....	6
1.5.2. Užduočių analizė	7
1.5.3. Programavimas	7
1.5.4. Priėmimo testavimas	8
1.5.5. Kokybės užtikrinimas	9
1.5.6. Konfigūracijos valdymas.....	9
1.5.7. Sprinto aptarimas.....	9
1.5.8. Defekto sprendimas	10
1.6. Išleidimas	11
1.7. Palaikymas	11
1.8. Balius/Post-mortem	12
ŽODYNĖLIS	13

Įvadas

Šiame darbe bus pristatytas „Mėnuliukų technologijų“ programų kūrimo procesas. Pats procesas yra paremtas Agile metodologija su minimaliais pakeitimais reikalavimų rinkime. Proceso pradžioje stengiamės su užsakovu išsiaiškinti norimus įgyvendinti funkcionalumus ir bendraujant kartu su užsakovu sudaryti reikalavimus. Sudarant reikalavimus yra diskutuojama ir sistemos ateities vizija, siekiant susidaryti geresnę perspektyvą sistemos ateičiai ir darbartiniams reikalavimams. Įmonė įvertina, kiek valandų užtruks kiekvieno funkcionalumo sukūrimas, o mokestis yra imamas už pradirbtas valandas. Klientui nesutikus su pateiktomis kainomis yra daromi susitikimai siekiant paaiškinti valandų vertinimą. Po susitikimų funkcionalumo įgyvendinimo valandos gali keistis, arba funkcionalumas bus atsisakytas. Kiekvieno sprinto pradžioje, po reikalavimų patikslinimo, yra sudaromi priėmimo testai, kuriuos praėjus kliento prašoma susimokėti už atliktus darbus.

1. Kūrimo procesas



1 pav. Sistemos kūrimo procesas

1.1. Projekto siūlymas

1. Pirmas susitikimas, kuriame aptariamas galimas projektas, kiekviena pusė išsako savo lūkesčius, pasidalinama idėjomis.
2. Po susitikimo įmonė paruošia pradininį pasiūlymą, į kurį įeina orientacinės finansų ribos, žmonių ištekliai, kurie galėtų būtų skiriami šiam projektui. Šis pasiūlymas aptariamas su

1 lentelė. Projekto siūlymo procesas

Pavadinimas:	Projekto siūlymas.
Tikslas:	Aptarti galimą projektą su galimu klientu.
Vykdytojai:	Projekto vadovas ir klientas.
Veiklos:	V1 - Aptariama sistemos aprėptis. V2 - Nustatomos kainos ribos iki pirmo sistemos išleidimo. V3 - Pirminės sutarties pasirašymas .
Naudojami produktai:	NP1 - Buvusių projektų dokumentai.
Sukuriami produktai:	SP1 - Pirminė sutartis sistemos projektavimui.

klientu, kartu su juo dokumentuojami funkcionalumai, kurių klientas nori pirmame sistemos išleidime. Sėkmingai tesiantis tolesnėms deryboms nutariama dėl pradinio technologinio sprendimo pasiūlymo datos, bei finansavimo jam.

3. Pasirašoma pradinė sutartis, kurioje dokumentuojama prieš tai aptarta informacija. Ši sutartis galioja iki pirmojo prototipo pasiūlymo, po kurio atnaujinamos derybos dėl tolesnio projekto vystymosi.

1.2. Technologinio sprendimo paieška ir įgyvendinimas

2 lentelė. Technologinio sprendimo paieškos ir įgyvendinimo procesas.

Pavadinimas:	Technologinio sprendimo paieška ir įgyvendinimas.
Tikslas:	Išskirti technologijas ir jų versijas, kurios bus naudojamos projekte.
Vykdytojai:	Projekto vadovas, klientas, programuotojas.
Veiklos:	V1 - Aptarti technologijas, šiuo metu naudojamas projekte. V2 - Nustatomos technologijų kainos, kurios bus naudojamos projekte. V3 - Nutariama dėl technologinių alternatyvų.
Naudojami produktai:	NP1 - Esamos sistemos dokumentacija, norimų naudoti technologijų dokumentacija ir kainynas.
Sukuriami produktai:	SP1 - Technologinių sprendimų ir jų alternatyvų dokumentas su preliminaromis technologijų licenzijų kainomis.

1. Aptiriamos technologijos, kurios šiuo metu naudojamos projekte - išskiriami technologiniai karkasai, duomenų bazės, programavimo kalbų versijos ir kitos naudojamos technologijos ir jų versijos. Įvertinamas esamų technologijų saugumo lygis, greitaveika ir ateities palaikymas. Pasiūlomi technologiniai sprendimai pagrindžiant jų naudą sistemai.
2. Nustatomos technologijų kainos, kurios bus naudojamos projekte - paskaičiuojamos dabartinių technologijų kainos ir naujų siūlomų technologijų kainos.
3. Nutariama dėl technologinių alternatyvų - jeigu įmanoma klientui pasiūlomi atviro kodo technologiniai sprendimai siekiant sutaupyti pinigų. Pateikiamas palyginimas tarp dabartinės sistemos technologijos, siūlomos technologijos ir atviro kodo technologijų sprendimų.

1.3. Reikalavimų ciklas

3 lentelė. Reikalavimų ciklo procesas.

Pavadinimas:	Reikalavimų ciklas.
Tikslas:	Suformuoti funkcinis ir nefunkcinis reikalavimus.
Vykdytojai:	Programuotojas, analistas, klientas.
Veiklos:	V1 - Iš kliento pateiktų verslo reikalavimų suformuojame funkcinis reikalavimus. V2 - Pristatome klientui sudarytus funkcinis reikalavimus ir tiksliname pateiktus verslo reikalavimus. V3 - Siūlomi nefunkciniai reikalavimai.
Naudojami produktai:	NP1 - Kliento pateiktas verslo reikalavimų dokumentas.
Sukuriami produktai:	SP1 - Funkcinių ir nefunkcinių reikalavimų dokumentas.

1. Iš kliento pateiktų verslo reikalavimų suformuojame funkcinis reikalavimus - mūsų įmonės verslo analitikas dirbdamas kartu su programuotojais suformuoja atsekamus (su indentifikacijos kodu) funkcinis reikalavimus.
2. Pristatome klientui sudarytus funkcinis reikalavimus ir tiksliname pateiktus verslo reikalavimus - suformavus funkcinis reikalavimus planuojami susitikimai su klientu, siekiant jam pristatyti suformuotus reikalavimus, patikslinti pateiktus verslo reikalavimus ir toliau tikslinti reikalavimus iki kol reikalavimai tenkis klientą ir bus suprantami programuotojų komandai, kuri dirbs prie kliento projekto.
3. Siūlomi nefunkciniai reikalavimai - jeigu klientas pats nepateikė nefunkcinių reikalavimų įmonė pati pateikia nefunkcinių reikalavimų siūlymus pagal esamo projekto apimtį ir biudžetą. Pateikti nefunkciniai reikalavimai yra aptariami ir tikslinami su klientu.

1.4. Sistemos modeliavimas

4 lentelė. Sistemos modeliavimo procesas

Pavadinimas:	Sistemos modeliavimas.
Tikslas:	Skurti pradinę sistemos versiją.
Vykdytojai:	Programuotojas, analistas.
Veiklos:	V1 - Reikalavimų analizė iš implementacijos pusės. V2 - Pradinės sistemos rašimas ir testavimas. V3 - Atnaujintos sutarties pasirašymas.
Naudojami produktai:	NP1 - Sistemos reikalavimai. NP2 - Architektūrinių sprendimų dokumentas.
Sukuriami produktai:	SP1 - Pradinė kodo bazė. SP2 - Atnaujinta projekto sutartis.

1. Pagal gautus reikalavimus programuotojų komanda sumodeliuoja pradinės sistemos implementaciją, pradedama rašyti kodo bazė ant kuriuos ateinančiuose sprintuose bus statoma visa sistema.
2. Sukurtas sistemos modelis yra pristatomas klientui, jeigu jį tenkina pasirinkta kryptis, yra pasirašoma sutartis tolimesniam bendradarbiavimui.

1.5. Sprints

Sprintu skaitome dvi savaites, kurių pabaigoje yra įvykdytas verslo analitiko ar komandos parinktas užduočių skaičius ir matomas apčiuopiamas rezultatas - veikiantis funkcionalumas. Sprinto ilgis - 2 savaitės - pasirinktas taip, kad nebūtų sunku numatyti ir suplanuoti užduočių tam laiko tarpui ir taip, kad būtų pakankamai laiko jas įvykdyti iki galo. Sprintas turi kelias veiklas, kurios skirtos palaikyti efektyvią programos kūrimo eigą ir užtikrinti, kad rezultatas būtų pasiektas laiku.

1.5.1. Sprinto užduočių tikslinimas ir vertinimas

5 lentelė. Sprinto užduočių tikslinimo ir vertinimo procesas

Pavadinimas:	Sprinto užduočių tikslinimas ir vertinimas.
Tikslas:	Įvertinti ir paanalizuoti užduotis.
Vykdytojai:	Programuotojas bei Įmonės analistas arba užsakovo atstovas, klientas
Veiklos:	V1 - Analistas paaiškina užduotis iš verslo perspektyvos. V2 - Užduočių vertinimo sesija tarp programuotojų. V3 - Įvertintų užduočių aptarimas su analistu
Naudojami produktai:	NP1 - Užduočių sąrašas.
Sukuriami produktai:	SP1 - Sprinto užduočių sąrašas. SP2 - Įvertintos užduotys backloge(vertimas?)

1. Pradinis užduočių aptarimas su analistu arba užsakovo atstovu, šiame aptarime paaiškinami kiekvienos užduoties scenarijai iš dalykinės srities pusės, paaiškinami reikalavimai, kurie turi būti įgyvendinami prieš pabaigiant užduotį. Diskusijos su programuotojais metu galimi užduočių ar jų priėmimo kriterijų pakeitimai arba, jeigu jų negalima atlikti nepasitarus su klientu, užduotis yra atnaujinama vėliau, pasitarus su klientu.
2. Atlikus pradinį aptarimą kartu su analistu arba verslo atstovu rengiama diskusija tarp programuotojų, kuriame kiekviena užduotis yra įvertinama taškais, vertinama fibonačio sekos skaičiais, o skaičiaus reikšmė yra viena programuotojo darbo diena. Šioje diskusijoje programuotojai diskutuoja apie galimą kiekvienos užduoties implementaciją, bei jos sudėtingumą, tada balsuojama dėl šiai užduočiai skiriamo balų skaičiaus. Taip aptariamos visos dar neįvertintos užduotys, tuo atveju, jeigu gilinantis į implementaciją atsiranda neaiškumų dėl

dalykinės srities, užduotis yra blokuojama paliekant komentarą, kad jį radęs analistas galėtų patikslinti užduotį.

3. Atlikus užduočių vertinimą analistas arba verslo atstovas sudeda sekančio sprinto struktūrą pagal galimą talpą (talpa lygi programuotojų skaičiui padauginta iš 8).

1.5.2. Užduočių analizė

6 lentelė. Užduociu analizė

Pavadinimas:	Užduočių analizė
Tikslas:	Surinkti visą reikalingą informaciją užduočiai įvykdyti
Vykdytojai:	Programuotojas
Veiklos:	V1 - Klausimų, į kuriuos reikia atsakymo, iškėlimas V2 - Informacijos rinkimas, atsakymas į klausimus V3 - Įgyvendinimo alternatyvų aprašymas V4 - Jei reikia, įrodymas, kad alternatyva veiks ar neveiks
Naudojami produktai:	NP1 - Užduočių sąrašas
Sukuriami produktai:	SP1 - Atsakymai į iškeltus klausimus SP2 - Implementacijos alternatyvų sąrašas ir aprašymas SP3 - Jei reikia, pasirinktos alternatyvos supaprastinta implementacija SP4 - Jei reikia, papildytas, patobulintas užduočių sąrašas

Turint pilnai aprašytas užduotis vyksta užduočių analizė, jei yra tam poreikis. Jei kyla daugiau neiš-kumų dėl projekto vykdymo ateities planų gali būti sukuriamos atskiros užduoties tam tikros srities išsiaiškinimui tam, kad geriau išsiaiškinti galimas implementacijos alternatyvas. Tokios analizės rezultatas - dokumentas, pateikiantis klausimus ir atsakymus, implementacijos alternatyvas ir ki-lusius kitus pastebėjimus. Po analizės turi būti aišku, kaip ir su kokiomis technologijomis užduotis bus įvykdoma ir, jei reikalinga, tikslinamos užduotys.

Jei analizė reikalinga mažesniu mastu, bet dar negalima iškart imti ir programuoti - asme-niškai, jau pasiėmus užduotį, aiškinamasi, kokių žinių trūksta ir kas jas galėtų suteikti. Tai gali būt pasikalbėjimas su kolegomis ar panašių užduočių implementacijos pavyzdžių ieškojimas. Šios veiklos pabaigoje jau galima pradėti programuoti.

1.5.3. Programavimas

1. Atlikus užduočių analizę ir išsiaiškinus kodo implementacijos kryptį, pradedama rašyti už-duoties implementacija. Atlikus implementaciją kodas yra peržiūrymas komandus narių, bet pagal jų rekomendacijas pamodifikuojamas.
2. Atlikus užduoties implementaciją programuotojas iš pradžių rankiniu būdu pratestuoja ar jo sukurtas funkcionalumas veikia tinkamai, o radęs netikslumų juos ištaiso.

7 lentelė. Programavimo procesas

Pavadinimas:	Programavimas.
Tikslas:	Suprogramuoti sprinto užduotis.
Vykdytojai:	Programuotojai.
Veiklos:	V1 - Kodo rašymas. V2 - Rankinis testavimas. V3 - Automatinių testų rašymas.
Naudojami produktai:	NP1 - Užduočių aprašymas.
Sukuriami produktai:	SP1 - Užduoties kodas. SP2 - Modulių testai. SP3 - Integraciniai testai.

3. Rankinio testavimo būdu pratestavus funkcionalumą yra rašomi automatiniai testai skirti patikrinti sudėtingesnius scenarijus, bei ateities regresiniam testavimui užtikrinti. Į šiuos testus įeina modulių, aptarnavimo bei integraciniai testai.

Šis žingsnis glaudžiai susijęs su prieš jį einančiomis fazėmis, jame vykdomas suprojektuotos sistemos dalies įgyvendinimas, kuris apima kodo rašymą, automatizuotų testų kūrimą, konfigūracijos pakeitimus ir duombazės kūrimą. Ši fazė įgyvendinama tik gerai išsiaiškinus ir susidokumentavus reikalavimus ir jau turint prototipą, kad kuriama programa atitiktų kliento lūkesčius ir reikalingų pakeitimų būtų mažiau.

1.5.4. Priėmimo testavimas

8 lentelė. Priėmimo testavimo procesas

Pavadinimas:	Priėmimo testavimas
Tikslas:	Patikrinti ar sistema atitinka verslo reikalavimus
Vykdytojai:	Analistas ir programuotojai
Veiklos:	V1 - Priėmimo testavimo planavimas V2 - Priėmimo testų klasifikavimas V3 - Priėmimo testavimas
Naudojami produktai:	NP1 - Verslo reikalavimų dokumentas
Sukuriami produktai:	SP1 - Priėmimo testų planas SP2 - Priėmimo testai

Priėmimo testavimo procesas vykdomas cikliškai su programavimo procesu kiekviename sprinte. Proceso metu, prieš sukurtos sistemos ar jos dalies pristatymą klientui, tikrinama, ar sukurta sistema atitinka verslo reikalavimus. Įmonė vykdo vidinį priėmimo testavimą, kurį atlieka įmonėje dirbantys, tačiau teisiogiai su projektu, programavimu ar testavimu nesusiję darbuotojai: programuotojui atlikus užduotį analitikas atlieka priėmimo testus. Tam tikrais atvejais naujai programos versijai gali būti vykdomas dūmų testas prieš kokybės užtikrinimo procesą. Priėmimo testavimo procesą sudaro priėmimo testavimo planavimas, testų klasifikavimas bei priėmimo testavimas.

1.5.5. Kokybės užtikrinimas

9 lentelė. Kokybės užtikrinimo procesas

Pavadinimas:	Kokybės užtikrinimas.
Tikslas:	Pasirūpinti, kad sistema su atnaujinta kodo baze veikia teisingai.
Vykdytojai:	Testuotojai ir programuotojai.
Veiklos:	V1 - Rašomi modulių testai V2 - Rašomi integraciniai testai V3 - Rašomi automatizuojami testai V4 - Vykdomas regresinis testavimas V5 - Testų klaidų analizė ir defektų aprašymas
Naudojami produktai:	NP1 - Užduočių sąrašas NP2 - Testai regresiniam testavimui
Sukuriami produktai:	SP1 - Nauji defektai SP2 - Testų rezultatų dokumentas SP3 - Papildytas naujais defektais užduočių sąrašas

Kokybės užtikrinimas skirtingas kiekvienam projektui. Tuose projektuose, kuriuose kuriamoje sistemoje egzistuoja vartotojo sąsaja, atliekamas rankinis testavimas kartu su automatizuotu regresiniu testavimu. Į kokybės užtikrinimą yra įtraukiami ir programuotojai, kurie yra atsakingi už modulių testų rašymą ir klaidų taisymą. Testuotojai atsakingi už rankinį testavimą, automatizuotų testų rašymą bei regresinių testų rezultatų apibendrinimą. Jeigu projektas yra iteracinis ir jau išleistas plačiam naudojimui, po sėkmingo testavimo kodas yra sudedamas į aukštesnę aplinką, kurioje yra papildomai praliduojamas prieš jį išleidžiant į produkciją. Jei testavimas parodė defektus - jie užregistruojami kaip užduotys ir jų taisymui, pagal jų sunkumą, skiriamas prioritetas.

Ir programuotojai ir testuotojai atsakingi už tvarkingą ir kokybišką kodą ir sklandų programos veikimą. Viena rečiau minima kokybės užtikrinimo dalis yra kodo peržiūra. Kiekvienas programuotojas yra atsakingas už kitų savo rato programuotojų kodo peržiūrą prieš jo kėlimą į aukštesnes aplinkas. Taip išvengiama kartais net labai sunkių klaidų dėl vieno ar kelių žmonių apsižiūrėjimo. Tuo pačiu išvengiama nedarbingo laiko, kuris atsirado dėl kitų programos dalių sulaužančio kodo įdėjimo į pagrindinę "master" kodo bazę.

1.5.6. Konfigūracijos valdymas

1. Srinto pabaigoje devOps specialistai peržvelgia atliktus konfigūracinius pakeitimus, kuriuos programuotojai pažymi sprinto metu. Šie konfigūracijos pakeitimai yra įdedami į aukštesnę aplinką. Taip pat po sprinto kodas yra sudedamas į aukštesnę aplinką tolimesniam testavimui.

1.5.7. Srinto aptarimas

10 lentelė. Konfigūracijos valdymo procesas

Pavadinimas:	Konfigūracijos valdymas.
Tikslas:	Atnaujinti sprinto konfigūraciją.
Vykdytojai:	devOps specialistai, programuotojai.
Veiklos:	V1 - Repozitorijos su konfigūracijomis atnaujinimas. V2 - Duomenų bazės atnaujinimas.
Naudojami produktai:	NP1 - Konfigūracijos pakeitimų sąrašas.
Sukuriami produktai:	SP1 - Pamodifikuota konfigūracijų repozitorija. SP2 - Atnaujinta duomenų bazė.

11 lentelė. Sprinto aptarimas

Pavadinimas:	Sprinto aptarimas
Tikslas:	Užtikrinti, kad sprintai būtų efektyvūs
Vykdytojai:	Komanda
Veiklos:	V1 - Problemų iškėlimas V2 - Problemų aptarimas V3 - Iškeliama pasiūlymai sekančio sprinto našumui gerinti
Naudojami produktai:	NP1 - Sprinto užduočių sąrašas
Sukuriami produktai:	SP1 - Užduotys komandai sekančio sprinto efektyvumui gerinti.

Komandos efektyvumui labai svarbu aptarti praėjusį sprintą: kas trukdė efektyviam darbui, kas gerai sekė, kokias programavimo veiklas tęsti, kokias nutraukti, kokios komandos nuotaikos ir to priežastys. Šioje veikloje dažniausiai dalyvauja tik komanda, o jos rezultatas - pagal išreikštus pastebėjimus išskirti tobulėjimo punktai, kurių laikomasi būsimame sprints siekiant geresnės ir greitesnės projekto implementacijos. Tai gali būti aktyvus kažkokių procesų vykdymo trukdžių sprendimas, komunikacija su kitomis komandomis ar klientu dėl iškilusių problemų.

1.5.8. Defekto sprendimas

12 lentelė. Defekto sprendimo procesas

Pavadinimas:	Defekto sprendimo
Tikslas:	Pašalinti defektą
Vykdytojai:	Programuotojai ir testuotojai
Veiklos:	V2 - Defekto analizė V3 - Defekto pašalinimas V4 - Patikrinimas ar defektas pašalintas
Naudojami produktai:	NP1 - Defekto aprašymas
Sukuriami produktai:	SP1 - Defekto analizė SP2 - Defekto sprendimas

1. Gavus defektą programuotojas pagal pateiktus žingsnius pakartoja defektą (jeigu jo pakartoti nepavyksta, jis grąžinamas atgal), bei išanalizuoja to defekto priežastis, bei galimą sprendimą.

2. Suradus sprendimą defektas yra pašalinamas, jeigu reikia papildomi automatiniai testai, kurie patikrina funkcionalumo atvejį, kuriame buvo gautas defektas.

1.6. Išleidimas

13 lentelė. Išleidimo procesas

Pavadinimas:	Išleidimas
Tikslas:	Išleisti naują sistemos versiją
Vykdytojai:	Programuotojai ir devOps specialistai
Veiklos:	V1 - Sistemos išleidimas V2 - Sistemos būsenos bei validumo stebėjimas
Naudojami produktai:	NP1 - Esama sistema
Sukuriami produktai:	SP1 - Išleista sistema

Išleidimo stadiją galima skaidyti į 2 skirtingus tipus - galimas naujas projekto išleidimas, kurio metu vartotojui pateikiama nauja sistema, kuri buvo tam tikrą laiką kuriama. Taip pat galimas variantas, kai egzistuoja veikianti sistema, kuri yra periodiškai atnaujinama (priklausomai nuo sprinto ilgio). Išleidimo metu įmonėje budi darbuotojų, atsakingi už greitą išleistos sistemos trikdžių pašalinimą.

1.7. Palaikymas

14 lentelė. Palaikymo procesas

Pavadinimas:	Palaikymas
Tikslas:	Užtikrinti korektišką sistemos veikimą po sistemos paleidimo
Vykdytojai:	Projektų vadovas, programuotojas, analistas, DevOps specialistas
Veiklos:	V1 - Analizuoti kliento pateiktus palaikymo darbus V2 - Registruoti palaikymo darbus V3 - Perduoti darbus defekto analizės procesui
Naudojami produktai:	NP1 - Esama sistema NP2 - Sistemos palaikymo sutartis NP3 - Užregistruoto defekto informacija NP4 - Vartotojo užregistruota palaikymo užduotis
Sukuriami produktai:	SP1 - Defekto aprašymas SP2 - Palaikymo darbo aprašymas

1. Analizuoti kliento pateiktus palaikymo darbus - klientui pateikus norimus palaikymo darbus analistas išanalizuoja, ar užregistruoti darbai įeiną į palaikymo sutartį, ar tai yra papildomi darbai, už kuriuos klientas turės susimokėti papildomai.

2. Registruoti palaikymo darbus - išanalizavus poreikį yra nusprendžiama kam darbas turi būti perduotas(programuotojams, testuotojams, DevOps) ir darbas yra užregistruojamas. Jeigu klientas nesutinka dėl palaikymo darbo statuso(jeigu mano, kad darbas priklauso palaikymui o ne papildomiems darbams) vykdomas susitikimas su klientu kurio metu aiškinamasis nesutarimas.
3. Perduoti darbus defekto analizės procesui - po analizės nutarus, kad darbas yra defektas jis yra perduodamas defekto analizės procesui.

Palaikymo procesui kuriamas palaikymo planas, susidedantis iš programos paruošimo, problemos identifikavimo bei produkto konfigūracijos valdymo. Problemos identifikavimas vykdomas tikrinant programos validumą, sukuriant problemos sprendinį bei išskiriant resursus modifikacijai įgyvendinti. Proceso patvirtinimas įgyvendinamas gavus patvirtinimą dėl ketinamų įgyvendinti pakeitimų iš užklauso autoriaus. Įmonė teikia dviejų tipų palaikymą: taisomąjį bei adaptacinį. Taisomasis palaikymas orientuotas į problemų, atrastų vartotojų arba vartotojų klaidų reportų analizės metu, taisymą. Adaptacinis palaikymas skirtas nūdienos standartų programose palaikymui. Įmonė laikosi „Boehm“ modelio, kuris pasižymi atitinkamai pakeitimų pasiūlymu, patvirtinimu bei įgyvendinimu.

1.8. Balius/Post-mortem

Procesas užbaigiamas komandos bei prie projekto prisidėjusių žmonių švente, kurioje aptariamas bei įvertinamas proceso pasisekimas ir daromos išvados.

Žodynėlis

- Klientas - žmogus arba žmonių grupė, kuri nusipirko programavimo paslaugas iš mūsų įmonės.
- Procesas - veiksmų seka reikalinga pasiekti užsibrėžtą tikslą.
- Užsakovas - žmogus arba žmonių grupė, kuri nusipirko programavimo paslaugas iš mūsų įmonės.
- Funkcinis reikalavimas - sistemos arba sistemos komponento veikimo apibrėžimas kuriame nurodytos elgesys tarp įvesties ir išvesties.
- Nefunkcinis reikalavimas - reikalavimas apibrėžiantis kriterijus pagal kuriuos galima spręsti apie sistemos veikimą, ne apie tikslią elgseną.
- Verslo reikalavimas - kritinės įmonės veiklos, kurios privalo būti įgyvendintos norint atitikti organizacijos tikslus tuo pačiu metu nepriklausant tuo konkrečiam sprendimui.
- Sprintas - viena tiksliai laiku apibrėžta iteracija kurios metu vyksta kūrimo ciklas.
- Susitikimas - mūsų įmonės ir kliento atstovų susitikimas gyvai apie nuotoliniu būdu.
- Programuotojas - įmonės darbuotojas atliekantis sistemos palaikymo ir kūrimo darbus.
- Analitikas - žmogus atsakingas už verslo poreikių analizę ir funkcinių reikalavimų sudarymą.
- Testuotojas - žmogus kuriantis testavimo dokumentaciją, automatinius, integracinius ir regresinius testus. Taip pat testuotojas atlieka rankinius testus.
- Projekto vadovas - žmogus atsakingas už bendrą projekto planavimą ir įgyvendinimą.
- DevOps specialistas - darbuotojas, kuris sujungia programavimo ir informacinių technologijų operacijų praktikas siekdamas sumažinti kūrimo gyvavimo cikla ir suteikti nenutrūkstamą funkcionalumo pristatymą ir aukštą programos kokybę.
- Karkasas - abstrakcija kuria programa pateikia funkcionalumą, kuriuo naudojantis galima kurti sistemas pridedant programuotojo parašyto kodo.
- Verslo atstovas - žmogus arba žmonių grupė atstovaujanti kliento įmonei. Atstovo tikslas yra užtikrinti projekto vykdymą ir reikalavimų aiškumą.
- Dalykinė sritis - sritis, kurioje naudojama sistema.