

1

2

3

4

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

5

6

**Įmonės „Mėnuliukų technologijos“ programų
kūrimo proceso aprašas**

7

8

**Description of the development process of the „Mėnuliukų
technologijos“ company**

9

Programų kūrimo proceso laboratorinis darbas

10

Atliko: 4 kurso 3 grupės studentai
Mėnuliukai (parašas)

Darbo vadovas: Saulius Ragaišis, Doc., Dr. (parašas)

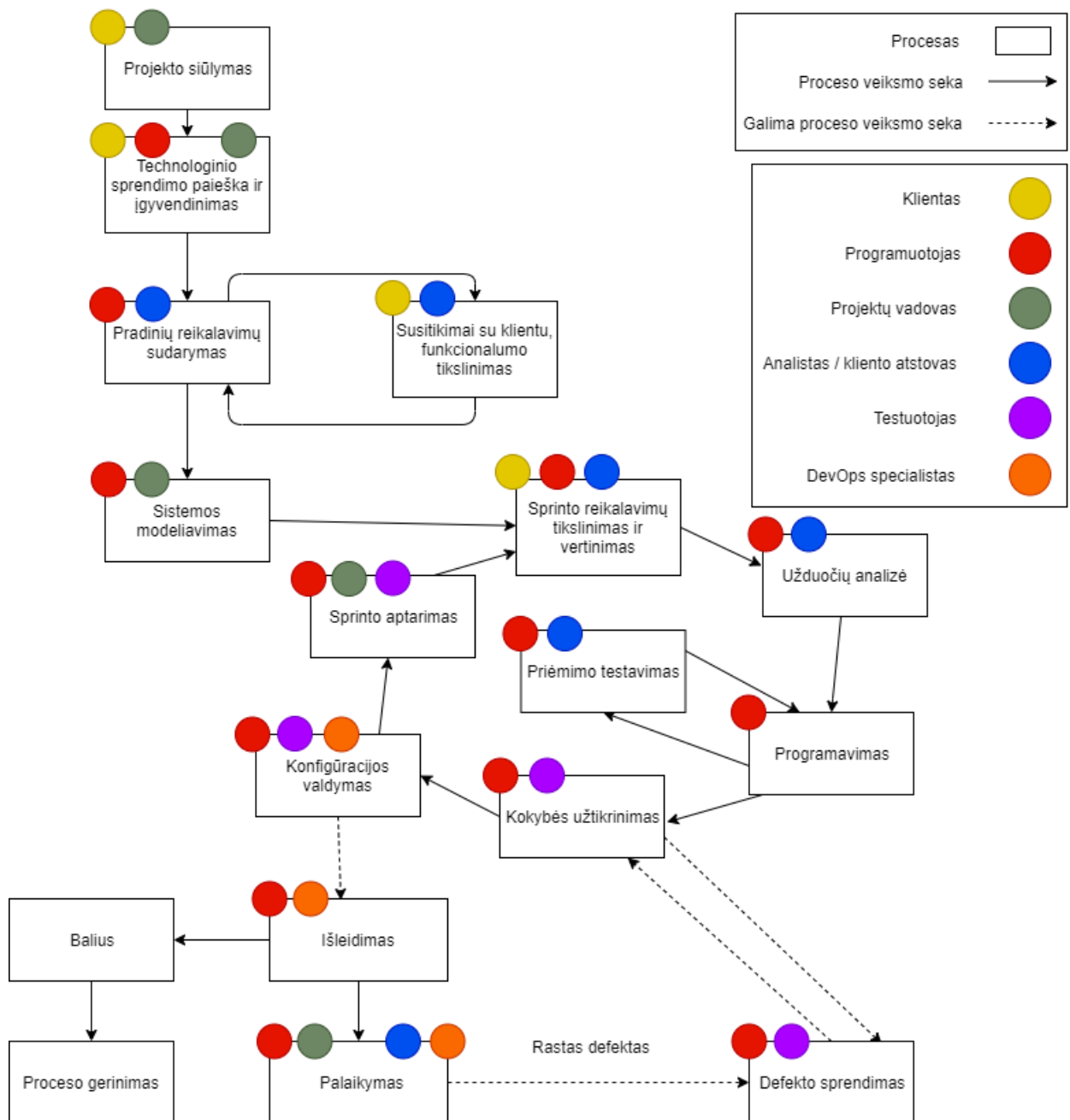
11

Vilnius – 2019

13 Įvadas

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

1. Kūrimo procesas



1 pav. Sistemos kūrimo procesas

25 1.1. Projekto siūlymas

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. |

26

- 27 1. Pirmas susitikimas, kuriame aptariamas galimas projektas, kiekviena pusė išsako savo lū-
28 kesčius, pasidalinama idėjomis.
- 29 2. Po susitikimo įmonė paruošia pradinį pasiūlymą, į kurį įeina orientacinės finansų ribos, žmo-
30 nių ištekliai, kurie galėtų būtų skiriami šiam projektui. Šis pasiūlymas aptariamas su klientu,
31 kartu su juo dokumentuojami funkcionalumai, kurių klientas nori pirmame sistemos išleidi-
32 me. Sėkmingai tęsiantis tolesnėms deryboms nutariama dėl pradinio technologinio sprendi-
33 mo pasiūlymo datos bei finansavimo jam.
- 34 3. Pasirašoma pradinė sutartis, kurioje dokumentuojama prieš tai aptarta informacija. Ši sutartis
35 galioja iki pirmojo prototipo pasiūlymo, po kurio atnaujinamos derybos dėl tolesnio projekto
36 vystymosi.

37 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. |

38

- 39 1. Aptariamos technologijos, kurios šiuo metu naudojamos projekte - išskiriami technologiniai
40 karkasai, duomenų bazės, programavimo kalbų versijos ir kitos naudojamos technologijos ir
41 jų versijos. Įvertinamas esamų technologijų saugumo lygis, greitaveika ir ateities palaikymas.
42 Pasiūlomi technologiniai sprendimai pagrindžiant jų naudą sistemai.
- 43 2. Nustatomos technologijų kainos, kurios bus naudojamos projekte - paskaičiuojamos dabar-
44 tinių technologijų kainos ir naujų siūlomų technologijų kainos.
- 45 3. Nutariama dėl technologinių alternatyvų - jeigu įmanoma klientui pasiūlomi atviro kodo tech-
46 nologiniai sprendimai siekiant sutaupyti pinigų. Pateikiamas palyginimas tarp dabartinės
47 sistemos technologijos, siūlomos technologijos ir atviro kodo technologijų sprendimų.

48 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. |
| | SP2 - Susaistytų šalių parašai reikalavimam patvirtinti. |

49

- 50 1. Iš užsakovo ir naudotojų pateiktų verslo reikalavimų suformuojame funkcinis ir nefunkci-
51 nius reikalavimus. Su visom partijom, įskaitant užsakovus ir naudotojus, susėdama, apta-
52 riami reikalavimai ir taip suderinami, kad visos susaistytos šalys būtų patenkintos. Iš visų
53 partijų surenkami parašai, tam kad vėliau nekiltų klausimų, kodėl reikalavimai neatitinka įsi-
54 vaizdavimų. Mūsų įmonės verslo analitikas dirbdamas kartu su programuotojais suformuoja
55 atsekamus (su identifikacijos kodu) funkcinis ir nefunkcinis reikalavimus.
- 56 2. Kompanija pateikia prieigą prie sistemos, su visu reikalavimų sąrašu, kur užsakovai gali
57 stebėti jų įvykdymą. Testuojant suprogramuotą funkcionalumą ir įgyvendinus reikalavimą,
58 sistemoje pažymimas jog reikalavimas įgyvendintas.
- 59 3. Išsiaiškinami baziniai reikalavimai, kurie yra patys svarbiausi ir turi būti pirmi įgyvendinti,
60 baziniai reikalavimai yra atitinkamai pažymimi sistemoje. Baziniai reikalavimai yra ypač
61 atidžiai analizuojami analistų, analizėje taip pat įvertinama, kokia tikimybė, jog pasikeis rei-
62 kalavimas ir atitinkamai suplanuojami veiksmai tokiu atveju.

- 63 4. Pristatome klientui sudarytus funkcinius reikalavimus ir tiksliname pateiktus verslo reika-
64 lavimus - suformavus funkcinius reikalavimus planuojami susitikimai su klientu, siekiant
65 jam pristatyti suformuotus reikalavimus, patikslinti pateiktus verslo reikalavimus ir toliau
66 tikslinti reikalavimus iki kol reikalavimai tenkins klientą ir bus suprantami programuotojų
67 komandai, kuri dirbs prie kliento projekto.
- 68 5. Siūlomi nefunkciniai reikalavimai - jeigu klientas pats nepateikė nefunkcinių reikalavimų
69 įmonė pati pateikia nefunkcinių reikalavimų siūlymus pagal esamo projekto apimtį ir biu-
70 džetą. Pateikti nefunkciniai reikalavimai yra aptariami ir tikslinami su klientu.
- 71 6. Jeigu klientas reikalavimą pakeičia programavimo fazėje tai reikalavimas yra atidžiai peržiū-
72 rimas dar kartą, įvertinama, kiek kitokio funkcionalumo reikės pakeisti kuriamoj programų
73 sistemoje, ir klientui pateikiama nauja sutartis, kurioje aprašoma, kokie darbai bus atlikti
74 tam, kad jau suprogramuotoje sistemoje pakeistas reikalavimas būtų įgyvendintas. Dar ne-
75 įgyvendintų reikalavimų keitimas yra pigesnis (nes nereikia perprogramuoti nieko) ir klientui
76 aiškiai išdėstoma kainos struktūra, kiek nauja sutartis kainuos, jeigu būs keičiami reikalavi-
77 mai, kurie jau įgyvendinti.

78 1.4. Sistemos modeliavimas

4 lentelė. Sistemos modeliavimo procesas

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

79

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

85 1.5. Sprintas

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

92 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 išanalizuoti 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 - Užduočių įvertinimas balais. |

93

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. Pasitikslinama ar reikalavimai pasikeitė, jeigu jie pasikeitė, atitinkamai užduotys yra keičiamos. 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 užduodys, tuo atveju, jeigu gilinantis į implementaciją atsiranda neaiškumų dėl dalykinės srities, užduotis yra blokuojama paliekant komentarą, kad ją 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. Užduočių 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 sukuriama 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.

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

122 1.5.3. Programavimas

7 lentelė. Programavimo procesas

| | |
|----------------------|---|
| Pavadinimas: | Programavimas. |
| Tikslas: | Suprogramuoti sprinto užduotis. |
| Vykdytojai: | Programuotojai. |
| Veiklos: | V1 - Kodo rašymas. V2 - Rankinis testavimas. V3 - PR sukūrimas. |
| Naudojami produktai: | NP1 - Užduočių aprašymas. |
| Sukuriami produktai: | SP1 - Užduoties kodas. SP2 - Užduoties PR. |

123

- 124 1. Atlikus užduočių analizę ir išsiaiškinus kodo implementacijos kryptį, pradedama rašyti už-
 125 duoties implementacija. Atlikus implementaciją kodas yra peržiūrimas komandos narių, bet
 126 pagal jų rekomendacijas pamodifikuojamas.
- 127 2. Atlikus užduoties implementaciją programuotojas sukuria PR (angl. Pull Request) prašymą,
 128 kad jo kodą įtrauktų į bendrą "master" kodą.

129 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. |

130

- 131 Priėmimo testavimo procesas vykdomas cikliškai su programavimo procesu kiekviename sprints.
- 132 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 dirban-
 tys, tačiau teisiogiai su projektu, programavimu ar testavimu nesusiję darbuotojai: programuotojai
 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. Praėjus te-
 stui, kuris pratestuoja reikalavimų funkcionalumą, reikalavimų sistemoje reikalavimas pažymimas
 kaip įgyvendintas, užsakovas gali pasižiūrėti tiksliai kada ir kaip įgyvendintas reikalavimas.

1.5.5. Kokybės užtikrinimas

9 lentelė. Kokybės užtikrinimo procesas

| | |
|----------------------|--|
| Pavadinimas: | Kokybės užtikrinimas. |
| Tikslas: | Pasirūpinti, kad sistema su atnaujinta kodo baze veiktų 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. SP4 - Vartotojo dokumentacija. |

141

Kokybės užtikrinimas skirtingas kiekvienam projektui. Tuose projektuose, kuriuose kuria-
 moje sistemoje egzistuoja vartotojo sąsaja, atliekamas rankinis testavimas kartu su automatizuotu
 regresiniu testavimu. Kiekvienas testas apibrėžia pradinis duomenis ir tikrina rezultatus. Šie su-
 daryti testai turi padengti visus užduoties bei programinės įrangos reikalavimus ir užtikrinti, kad
 rezultatai atitinka užduoties aprašymą bei nesugriauna kitų programos dalių. Į kokybės užtikrini-
 mą 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 rezulta-
 tų apibendrinimą. Jeigu projektas yra iteracinis ir jau išleistas plačiam naudojimui, po sėkmingo
 testavimo kodas yra sudedamas į aukštesnę aplinką, kurioje yra papildomai pralaiduojamas prieš
 jį išleidžiant į produkciją. Jei testavimas parodė defektus - jie užregistruojami kaip užduotys ir jų
 taisymui, pagal jų sunkumą, skiriamas prioritetas. Atlikus testavimus, jei vartotojo dokumentacija
 dar neparašyta vartotojo dokumentacija, ji parašoma implementuotom užduotim, o jei jos nėra, ji
 atnaujinama.

Ir programuotojai ir testuotojai atsakingi už tvarkingą ir kokybišką kodą ir sklandų prog-
 ramos veikimą. Viena rečiau minima kokybės užtikrinimo dalis yra kodo peržiūra. Kiekvienas

157 programuotojas yra atsakingas už kitų savo rato programuotojų kodo peržiūrą (angl. Pull Request
158 review) prieš jo kėlimą į aukštesnes aplinkas. Taip išvengiama kartais net labai sunkių klaidų dėl
159 vieno ar kelių žmonių apsižiūrėjimo. Tuo pačiu išvengiama nedarbingo laiko, kuris atsirado dėl
160 kitų programos dalių sulaužančio kodo įdėjimo į pagrindinę "master" kodo bazę.

161 1.5.6. Konfigūracijos valdymas

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. |

162

163 Sprinto pabaigoje devOps specialistai peržvelgia atliktus konfigūracinius pakeitimus, kuriuos prog-
164 ramuotojai pažymi sprinto metu. Šie konfigūracijos pakeitimai yra įdedami į aukštesnę aplinką.
165 Taip pat po sprinto kodas yra sudedamas į aukštesnę aplinką tolimesniam testavimui.

166 1.5.7. Sprinto aptarimas

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. |

167

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

174 1.5.8. Defekto sprendimas

12 lentelė. Defekto sprendimo procesas

| | |
|----------------------|--|
| Pavadinimas: | Defekto analizė. |
| 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. |

175

176 1. Defekto aprašymas pateikiamas ir sulyginamas su reikalavimų dokumentu, nustatoma ar
177 tikrai defektas yra validus. Gavus defektą programuotojas pagal pateiktus žingsnius pakar-
178 toją defektą (jeigu jo pakartoti nepavyksta, jis grąžinamas atgal), bei išanalizuoja to defekto
179 priežastis bei galimą sprendimą.

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

182 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. |

183

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

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. |

189

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 turi defektą, 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 katinamų įgyvendinti pakeitimų iš užklausos 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.

209 **1.8. Balius/Post-mortem**

210 Procesas užbaigiamas komandos bei prie projekto prisidėjusių žmonių švente, kurioje apta-
211 riamas bei įvertinamas proceso pasisekimas ir daromos išvados.

212 1.9. Proceso gerinimo procesas

213 Įvedamos procesų gerinimo metrikos. Pradėjus naują projektą kiekvieno proceso metu pro-
214 jektų vadovas žymisi metrikas apie procesą.

- 215 1. Proceso unikalūs indikatoriai (kiekvienam projektui skirtingas)
- 216 2. Kokia buvo numatyta proceso trukmė
- 217 3. Kiek laiko truko procesas
- 218 4. Ar visos numatytos šalys dalyvavo projekte
- 219 5. Kiek proceso veiklų buvo įgyvendinta
- 220 6. Proceso kaina
- 221 7. Atsiliepimai apie procesą

222 Pagal šias metrikas projekto pabaigoje per projekto aptarimą yra vykdomas pačio proceso aptarimas
223 ir, jeigu to reikia, gerinimas.

224 Jeigu projekto vadovas mato, kad pačio projekto vykdymo metu būtinas proceso pakeitimas
225 yra daromas susirinkimas su kompanijos savininkais ir suinteresuotais asmenimis siekiant spręsti
226 šį klausimą. Jeigu reikia, proceso modelio gerinimo procesas yra pradedamas ankščiau ir modelio
227 pakeitimai yra įgyvendinami kaip galima greičiau.

15 lentelė. Programų kūrimo proceso gerinimo procesas

| | |
|----------------------|---|
| Pavadinimas: | Proceso gerinimas. |
| Tikslas: | Aptarti praėjusio projekto programo kūrimo procesą ir pagerinti jį. |
| Vykdytojai: | Įmonės darbuotojai dirbę prie projekto ir įmonės vadovybė. |
| Veiklos: | V1 - Aptariamas kiekvienas modelio procesas. V2 - Apžvelgiamos metrikos. V3 - Surašomi gerinimo pasiūlymai. V4 - Atliekama proceso gerinimo atsiperkamumo analizė V5 - Sudaromas naujas programų sistemų kūrimo modelis |
| Naudojami produktai: | NP1 - Proceso gerinimo metrikos. |
| Sukuriami produktai: | SP1 - Pagerintas programų kūrimo proceso modelis. |

228

- 229 1. Bendrais bruožais aptariamas kiekvienas modelio procesas, kokia komandos nuomonė apie
230 tai, kaip sekėsi sekti proceso nurodymus jų pačių akimis, ir ar procesas jiems padėjo.
- 231 2. Apžvelgiamos metrikos ir kaip jos atsispindi tame, kas buvo kalbama pirmoje veikloje. Iš-
232 siaiškina, kodėl nebuvo sektas procesas, kodėl jis užtruko ilgiau arba trumpiau negu pla-
233 nuota ir kodėl buvo sunaudota daugiau arba mažiau pinigų negu planuota.

- 234 3. Kiekvienas susirinkimo dalyvis anonimiškai surašo savo pasiūlymus kaip gerinti procesą.
235 Pasiūlymai peržiūrimi susitinkimo dalyvių ir iškeliamą diskusiją trumpai aptariant visus ne-
236 sidubliuojančius pasiūlymus.
- 237 4. Pagal atrinktus pasiūlymus ir vadovybės išskirtus proceso gerinimo tikslus įvertinama, kiek
238 kiekviena gerinimo veikla kainuos implementuoti į modelį ir koks bus trumpalaikis ir ilga-
239 laikis veikos atsiperkamumas.
- 240 5. Atmetus nereikalingus tikslus sukuriamas naujas programų sistemos kūrimo proceso mo-
241 delis, kuriuo bus vadovaujama kituose projektuose.
- 242 Pastaba: jeigu modeliui pasikeitus projektas vis dar vyksta ir pakeitimai nėra kritiniai, lieka-
243 ma prie senos modelio versijos siekiant negluminti užsakovo.

244 Žodynėlis

- 245 • Klientas - žmogus arba žmonių grupė, kuri nusipirko programavimo paslaugas iš mūsų įmo-
246 nės.
- 247 • Procesas - veiksmų seka, reikalinga pasiekti užsibrėžtą tikslą.
- 248 • Užsakovas - žmogus arba žmonių grupė, kuri nusipirko programavimo paslaugas iš mūsų
249 įmonės.
- 250 • Funkcinis reikalavimas - sistemos arba sistemos komponento veikimo apibrėžimas, kuriame
251 nurodytas elgesys tarp įvesties ir išvesties.
- 252 • Nefunkcinis reikalavimas - reikalavimas, apibrėžiantis kriterijus, pagal kuriuos galima spręsti
253 apie sistemos veikimą, ne apie tikslią elgseną.
- 254 • Verslo reikalavimas - kritinės įmonės veiklos, kurios privalo būti įgyvendintos norint atitikti
255 organizacijos tikslus tuo pačiu metu, nepriklausant tuo konkrečiau sprendimo.
- 256 • Sprintas - viena tiksliai laiku apibrėžta iteracija, kurios metu vyksta kūrimo ciklas.
- 257 • Susitikimas - mūsų įmonės ir kliento atstovų susitikimas gyvai ar nuotoliniu būdu.
- 258 • Programuotojas - įmonės darbuotojas, atliekantis sistemos palaikymo ir kūrimo darbus.
- 259 • Analitikas - žmogus, atsakingas už verslo poreikių analizę ir funkcinį reikalavimų sudary-
260 mą.
- 261 • Testuotojas - žmogus, kuriantis testavimo dokumentaciją, automatinius, integracinius ir re-
262 gresinius testus. Taip pat testuotojas atlieka rankinius testus.
- 263 • Projekto vadovas - žmogus, atsakingas už bendrą projekto planavimą ir įgyvendinimą.
- 264 • DevOps specialistas - darbuotojas, kuris sujungia programavimo ir informacinių technologijų
265 operacijų praktikas siekdamas sumažinti kūrimo gyvavimo ciklą ir suteikti nenutrūkstamą
266 funkcionalumo pristatymą ir aukštą programos kokybę.
- 267 • Karkasas - abstrakcija, kuria programa pateikia funkcionalumą, kuriuo naudojantis galima
268 kurti sistemas pridedant programuotojo parašytą kodą.
- 269 • Verslo atstovas - žmogus arba žmonių grupė, atstovaujanti kliento įmonei. Atstovo tikslas yra
270 užtikrinti projekto vykdymą ir reikalavimų aiškumą.
- 271 • Dalykinė sritis - sritis, kurioje naudojama sistema.