

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
MATEMATINĖS INFORMATIKOS KATEDRA

Projekto vadovė **Aistė Čiplytė**

**Lukas Tutkus**

**Julius Daušas**

**Dominykas Smaliukas**

**Robert Stankevič**

Bioinformatikos studijų programos grupė BioSawmill

**Dvimatė pjovimo optimizacija**  
**Projekto planas**

# Turinys

<b>1. „Dvimatė pjovimo optimizacija“ PROJEKTO VIZIJA. . . . .</b>	<b>3</b>
<b>2. Dydžio ir pastangų vertinimas . . . . .</b>	<b>4</b>
2.1. Projekto dydžio ir pastangų apskaičiavimas . . . . .	5
<b>3. Tvarkaraštis . . . . .</b>	<b>8</b>
3.1. Tvarkaraščio užsiėmimai . . . . .	8
3.2. Projekto planas . . . . .	8
3.3. Kontroliniai taškai . . . . .	8
<b>4. Konfiguracionė kontrolė . . . . .</b>	<b>9</b>
<b>5. Versijų kontrolė . . . . .</b>	<b>11</b>
<b>6. Pakeitimų valdymo procedūra . . . . .</b>	<b>12</b>
<b>7. Resursų planas . . . . .</b>	<b>13</b>
<b>8. Rizikos valdymas . . . . .</b>	<b>14</b>
8.1. Identifikavimas . . . . .	14
8.2. Valdymo strategija . . . . .	14
8.3. Rizikos sumažinimas, . . . . .	16
8.4. Ką stebėti, atsižvelgti padidėjus rizikai . . . . .	16
8.5. Rizikos valdymas . . . . .	18

# 1. „Dvimatė pjovimo optimizacija“ PROJEKTO VIZIJA.

Atsižvelgus į iškeltus užsakovo bazinius reikalavimus siūlome sukurti programą (naudojant WEB modelį) skirtą užsakovui ar jo klientams gauti optimizuotą medienos pjovimo informaciją.

Siūlome turėti dvi šios sistemos versijas „BETA“ (terminuotą laiką, klientams išbandyti pagrindinius sistemos funkcionalumus) ir PILNĄJĄ (su visais įrankio funkcionalumais)

Sistema administruojama jos užsakovo administratoriaus teisėmis (galimybė matyti sistema besinaudojančių klientų duomenis (e-mail, aktyvacijos pradžios/pabaigos laikas, teisių lygis, pastabos), kurti/redaguoti/trinti klientų duomenis. Užsakovo klientus siūlome skirstyti į privačius asmenis ir įmones, kuriems pagal teises būtų sukuriamą galimybę turėti prieigą prie sistemos apibrėžtam/nurodomam administratoriaus skaičiui vartotojų.

BETA versijos vartotojas užsiregistravęs savo elektorniniu paštu prisijungia prie sistemos su labai ribotomis teisėmis (optimizuoto pjovimo skaičiavimas, vartotojo aktyvacijos galiojimo peržiūra, PILNOS versijos užsakymo langas). Optimizuoto pjovimo skaičiavimo lange jis turi galimybę nurodyti reikalingų detalių ilgį (mm), aukštį (mm) ir kiekius. Sistema optimizuos detalių išpjovimą iš standartinių panelių 1200x2500 ir 1200x3050, be iššvės informaciją ekrane. Vartotojui pasirinkus norimą variantą, sistema parodys ekrane pjovimo planą (spausdinimo galimybės nėra).

PILNA versija priklausomai nuo teisių leidžia:

- Papildomų vartotojų administravimą (kūrimas/redagavimas/šalinimas)
- Optimizuoto pjovimo skaičiavimas (standartinis funkcionalumas)
- Papildomo standartinio panelių dydžių įvedimas/redagavimas/šalinimas
- Galimybė pasirinkti kurioms panelėms atlikti skaičiavimus
- Galimybė atsispausdinti/išsaugoti bendrinę informaciją/ataskaitą (Vartotojo įvestas ilgis, plotis, kiekis, panelės/-ių dydis/-žiai ir optimalus ruošinių kiekis pirmoje panelėje)
- Istorijos peržiūra (išsaugotų skaičiavimų istorija su galimybe atsispausdinti planą)
- Pasirinkto plano atvaizdavimas naujame ekrane (PDF formatu)
- Galimybė atsispausdinti optimizuotus pasirinkto panelės dydžio planą.

Dėl sistemos aplinkos ji gali būti lengvai versijuojama ir tobulinama atsižvelgiant į tolimesnius užsakovo poreikius nepažeidžiant klientų istorijos .

## 2. Dydžio ir pastangų vertinimas

Atlikdami vertinamą naudosisime rekomenduojama funkcinių taškų metodika. Kadangi kuriama "web" aplikacija, tai pirma naudosisime atitinkamas "web" projekto sričių metrikas, noredami nustatyti apytikslei aplikacijos sudėtingumą. Jų pavadinimai pateikiami angliškai:

1. **Number of static Web pages (NSW)**
2. **Number of dynamic Web pages (NDW)**
3. **Number of internal pages links (NIL)**
4. **Number of persistent data objects (NPDO)**
5. **Number of external systems interfaced (NESI)**
6. **Number of static content objects (NSC)**
7. **Number of dynamic content objects (NDC)**
8. **Number of executable functions (NEF)**

Antra naudodami "web" metrika užpildysime funkciniais taškams skaičiuoti skirtas metrikas. Jas pateikiama angliškai:

1. **Number of external inputs**
2. **Number of external outputs**
3. **Number of external inquiries**
4. **Number of internal logical files**
5. **Number of external interface files**

Trečia matuojamas projekto sudėtingumas siekiant tikslesnio dydžio vertinimo. Tai daroma atsakant į klausimus priskiriant kokybinius vertes nuo 0 - No influence iki 5 - Essential. Klausimai pateikiami angliškai.

1. **Does the system require reliable backup and recovery?**
2. **Are data communications required?**
3. **Are there distributed processing functions?**
4. **Is performance critical?**
5. **Will the system run in a existing, heavily utilized operational environment?**
6. **Does the system require on-line data entry?**

7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
9. Are the inputs, outputs, files or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Ketvirta apskaičiuojami funkciniai taškai pagal formulę

$$FP = count\ total \times [0.65 + 0.01 \times \sum(F_i)]$$

$FP$  - projekto dydžio vertinimas funkciniais taškais.

## 2.1 Projekto dydžio ir pastangų apskaičiavimas

NSW skaičiuojamas:

$$NSW \approx 3 \times NDW$$

NDW skaičiuojamas:

$$NDW \approx 5$$

NIL skaičiuojamas:

$$NIL \approx 4 + 2$$

NPDO skaičiuojamas:

$$NPDO \approx NDW \times 3$$

NESI skaičiuojamas:

$$NESI \approx 1$$

NSC skaičiuojamas:

$$NSC \approx (NSW + NDW) \times 5$$

NDC skaičiuojamas:

$$NDC \approx NDW \times 3$$

NEF skaičiuojamas:

$$NEF \approx 0$$

Number of external inputs:  
 NDW  
 Number of external outputs:  
 NDC  
 Number of external inquiries:  
 NESI  
 Number of internal logical files:  
 NPDO  
 Number of external interface files:  
 1

Domain	count	Weight	FP
External inputs	5	3	15
External outputs	15	1	15
External inquiries	1	10	10
Internal logical files	15	1	15
External interface files	1	7	7
Total:			62

1. 3  
 2. 1  
 3. 0  
 4. 1  
 5. 0  
 6. 5  
 7. 3  
 8. 3  
 9. 4  
 10. 5  
 11. 0  
 12. 0  
 13. 0  
 14. 3  
 Viso: 28

$$FP = 62 \times 0.93 = 57.66$$

EI:

$$S_1 = (4 + 4 \times 5 + 6) \div 6$$

EO:

$$S_2 = (12 + 4 \times 15 + 17) \div 6$$

EQ:

$$S_3 = (1 + 3 \times 1 + 2 \times 2) \div 6$$

ILF:

$$S_4 = (12 + 4 \times 15 + 17) \div 6$$

EIF:

$$S_5 = (1 + 4 \times 1 + 2) \div 6$$

Domain	Opt.	Likely	Pess.	Est.	Weight	FP
External inputs	4	5	6	5	3	15
External outputs	12	15	17	15	1	15
External inquiries	1	1	2	1.5	10	15
Internal logical files	12	15	15	0	1	15
External interface files	1	1	2	1	7	7
Total:						67

$$FP = 67 \times 0.93 = 62.31$$

Function Points per Person Monht table.

$$E = 62.31 \div 7.1 = 8.8$$

### 3. Tvarkaraštis

#### 3.1 Tvarkaraščio užsiėmimai

Darbuotojas	Darbas	Valandų skaičius
Projekto valdytojas	Prižiūri darbus, komunikuoja su užsakovu	

#### 3.2 Projekto planas

Diena	Planas
Pirmadienis	Darbų tęsimas, bei sprint'o savaitės planavimas užduočių visai savaitei planavimas
Antradienis	Peržiūrėjimas užduočių, jų atlikimas, siuntimas testuotojams
Trečiadienis	Peržiūrėjimas užduočių, jų atlikimas, siuntimas testuotojams
Ketvirtadienis	Peržiūrėjimas užduočių, jų atlikimas, siuntimas testuotojams
Penktadienis	sprint' o savaitės darbų pristatymas, atliktų neatliktų užduočių tobulinimas.

Projektas vykdomas Agile projektavimo tipu. Kadangi jis leidžia greitai spręsti iškilusias problemas bei beveik be pastangų tobulinti projektą.

Projekto kaina apskaičiuojama taip:

$Kaina(LTL) = Pastangos(mėnesiai žmogui) * Darbo mėnesio kaina(LTL/Mėnesį).$

Įmonės mėnesio darbo kaina, 175000 LTL, tad projekto kaina būtų:

$Kaina = 1,93 * 17500 = 33775 \text{ LTL}$

#### 3.3 Kontroliniai taškai

Projekto etapai savo ruožtu gali turėti keletą kontrolinių taškų. Juose nurodomi suplanuoti terminai, iki kada turėtų būti atlikti kažkurie projekto darbai, ir suplanuotos biudžeto išlaidos. Kontroliniai taškai padeda įvertinti projekto stovį duotais laikais momentais.

- Projekto planavimas ir kitų klausimų sprendimas - 1 savaitė (spalio 1d. - spalio 8d.)  
Išlaidos 5 tūkstančiai litų.
- Reliacinės duomenų bazės kūrimas - 1 savaitė (spalio 8d. – spalio 15d.) Išlaidos 4 tūkstančiai litų.



- Grafinės sąsajos ir serverio kūrimas – 1 savaitė (spalio 15d. – spalio 22d.) Išlaidos 4 tūkstančiai litų.
- Kreipinių kūrimo ir valdymo programinio komponento kūrimas - 1 savaitė (spalio 22d. – spalio 29d.) Išlaidos 4 tūkstančiai litų.
- Vartotojų valdymo programinio komponento kūrimas -1 savaitė (spalio 29d. – lapkričio 5d.) Išlaidos 4 tūkstančiai litų.
- Statistinių duomenų programinio paketo kūrimas - 1 savaitė (lapkričio 5d. – lapkričio 12d.) Išlaidos 4 tūkstančiai litų.
- Sistemų integracija į vieną sistemą, perkėlimas į U test aplinką ir testavimas - 2 savaitės (lapkričio 12d. - lapkričio 26d.) Išlaidos 10 tūkst ančiai litų.
- Sistemos pristatymas užsakovui ir paskutiniai testavimai - 2 savaitės (lapkričio 26d. - gruodžio 10d.) Išlaidos 7460 litų.

## 4. Konfiguracinė kontrolė

Kiekvieno produkto gyvavimo cikle atsiranda neišvengiamų pasikeitimų. Konfiguracinė kontrolė užtikrina, kad tie pasikeitimai būtų minimalūs. Šiame skyriuje aptarsiu galimus pasikeitimai iš užakovo pusės bei kaip juos kontroliuoti.

Užsakovas gali nuspręsti

- pakeisti produkto informaciją.
- kada ir kiek išplėsti, pagreitinti programos funkcionalumą.
- kokių servisų papildymo trūksta.

Kiekvina užsakovo sprendimą turime įvykdyti. Problema tame, kad reikia atsekti kada ir kur įvykdyti pakeitimą.

Problemos sprendimas ir būtų konfiguracionės kontrolės naudojimas.

Ji užtikrina, kad informacija saugojama, kontroliuojama sistemos kūrimo, keitimo ir testavimo metu ir taip pat ar atitinka vykdomus reikalavimus.

Taip pat užtikrinti kad informaciją apie projektą laisvai pasiekiamo.

Konfiguracinę kontrolės užduotys:

1. Rasti kokios projekto dalys gali keistis.
2. kontroliuoti vieną ar kelis rastus.

3. lengvinti skirtingų versijų aplikacijų konstrukciją.
4. užtikrinti kad projekto kokybė palaikoma, kai konfigūracija atnaujinama.

Konfigūracijos kontrolė užtikrina programos konfigūracijos elementų identifikavimą, pasikeitimo kontrolę, versijos kontrolę, konfigūracijos auditą ir ataskaita tiek projekto kūrimo metu, tiek palaikymo metu.

## **Projekte kontrolė užtikrinama**

1. Programos dokumentacijai.
2. Programos analizės failai.
3. Programos diegimo failam.
4. Programavimo kodą.
5. Internetinės svetainės kodas.

## 5. Versijų kontrolė

Versijų kontrolei naudojamas "Github".

Ne tik programos kodui versijonuoti bet taip pat ir apie tai informuoti bendradarbius.

### **Joje laikomi failai:**

1. Vadyba ( organizacijos struktūra, architektūros informacija ).
2. Modeliavimas ( analizė, dizainas ).
3. Konstrukcijos ( šaltinio kodas, kompiliavimo instrukcijos )
4. Testai ( skriptai, rezultatai, kokybės matavimai ).
5. Dokumentai (projekto, dizaino dokumentai, naudojimo žinynas ).
6. Projekto valdymas ( projekto trukmė, tvarkaraštis, auditas ).

## 6. Pakeitimų valdymo procedūra

### Pakeitimų skirstymas

1. funkcijos arba aplikacijos ar rezultatų atvaizdavimo patobulinimai ar klaidų juose pataisymas
2. funkcijos ar atvaizdavimo patobulinimai, kurie turi įtakos kitoms projekto programoms
3. funkcijų ir naudojamų duomenų pakeitimai, kurie stipriai pakeičia, patobulina ar supaprastina web aplikacijų ar rezultatų atvaizdavimą
4. dideli dizaino ar navigacijos web aplikacijoje pakeitimai, kurie stipriai įtakoti vartotoją.

Šioje pastraipoje yra aprašomas algoritmas, kaip bus veikama, išskylus poreikiui atlikti pakeitimus. Susidūrus su noru kažką pakeisti, jam, pirma, bus priskirta apimties/svarbumo klasė. Nepriklausomai nuo priskirtos klasės, jeigu pakeitimui įgyvendinti nėra nepritariama, tai jis bus pradedamas vykdyti pažingsniui:

1. taisytinas objektas (programos kodas) bus išsikeliamas (check-out)
2. atliekami pakeitimai
3. ištestuojami pakeitimai
4. pakeitimai įkeliami į repozitoriją (check-in)
5. paleidžiama web aplikacija su atliktu pakeitimu.

Priklausomai nuo priskirtos klasės, dar prieš išsikeliant taisytinus objektus, atliekamos tokios procedūros:

- 2-os kl. pakeitimui atlikti tikrinami visi objektai, kurie susiję su keistinu objektu. Įvertinama, kokią įtaką jiems tie pakeitimai turės. kiek tuomet pakeitimų (derinimų) reikės atlikti šiuose susijusiuose objektuose.
- 3-os kl. pakeitimui atlikti rašomas glaudus pakeitimo aprašymas.  
Pateikiamas visiems projekto vykdytojams, kad šie susipažintų ir pritartų,

ir nesant jų pritarimui, apie pakeitimą diskutuojama ieškant kompromiso arba jis visiškai atmetamas.

- 4-os kl. pakeitimui atlikti rašomas glaudus pakeitimo aprašymas.

Pateikiamas visiems užsakovams, kad šie peržiūrėtų ir pritartų, o nesant pritarimui, klausomasi jų siūlymų, ieškoma abi puses patenkinančio susitarimo ir jam susiformavus, pakeitimas atliekamas.

## 7. Resursų planas

Žmonės esantys kompanijoje, darbuotojai.

Darbuotojas/ Pareigos	Darbuotojų kiekis	Darbo pobūdis	Darbo valandų <sup>1</sup>
Administratorius	1	Sprendžia iškilusias techninės ir programinės įrangos problemas. Atsakingas už darbuotojams reikalingų resursų pasiekiamumą.	52
Analitikas	1	Tikrina ar programuotojų sukurtas sistemos funkcionalumas atitinka kliento lūkesčius. Iš dalies atlieka testuotojo vaidmenį. Ruošia dokumentaciją.	100
Dizaineris	1	Kuria pradinį svetainės dizainą, o projekto eigoje keičia dizainą pagal kliento reikalavimus. Suteikia profesionalias konsultacijas programuotojams.	44
Programuotojas-testuotojas	2	Rašo programinį kodą, o vėliau jį testuoja.	120x2
Programuotojas-testuotojas	1	Rašo programinį kodą, o vėliau jį testuoja. Paskutiniame etape bus atsakingas už produkto palaikymo užtikrinimą.	140
Projekto vadovas	1	Pagrindinis asmuo bendraujantis su klientu. Atsakingas už projekto vystymą, komandos narių parinkimą. Konsultuoja komandos narius.	100
Sistemos inžinierius	1	Pagal pateiktus ir išanalizuotus kliento poreikius sudaro sistemos architektūrą (įskaitant ir techninę dalį). Konsultuoja programuotojus.	100

**Pastaba:** laikoma, kad žmogus dirbantis nepilnu etatu, tuo pačiu metu dirba prie kitų įmonės projektų.

<sup>1</sup>Darbo valandų skaičius apskaičiuotas atsižvelgus į darbuotojų numatytas darbo valandas kiekviename etape.

Techninės įrangos poreikiai:

- Kekvienam darbuotojui suteikiamas asmeninis kompiuteris, kuriame sudiegta reikalinga programinė įranga.

## 8. Rizikos valdymas

### 8.1 Identifikavimas

Nr.	Rizikų priežastys	Rizikos	Kategorija	tikimybė	Įtaka
1.	Kalbos barjeras, užsakovo laiko trūkumas	Komunikavimo trūkumas	PR	didelė	1
2.	Nėra atsiskaitymo taškų, mažai pasitarimų	Produktas neatitinka reikalavimų	VR	didelė	2
3.	Neužtikrinamos geros darbo sąlygos	Darbuotojai palieka darbovietę, dažnas darbuotojų keitimas, produkto vėlinimas maišytas produktas	PR	didelė	3
4.	Trumpos apklausos, mažai testavimo priimanč naujus darbuotojus	Nekvalifikuoti darbuotojai	DR	vidutinė	2
5.	Galimybių pervertimas Neaiškus darbo planas.	Produkto kokybė pablogėja neišbaigtos produkto dalys	PR	maža	1
6.	Darbas viršvalandžiais, blogas darbu planas.	Klaidų įvelimas, Laiko praleidimas	PR	vidutinė	3
7.	Darbuotojų nekomunikavimas	Produkto kokybė blogesnė, programinis nesuderinamumas	PR	maža	3
8.	Atsiskaitymo data atkeliam anksčiau	Vėlavimas	VR	maža	3
9.	Dokumento rašymo atidejimai.	Produkto dokumentacija nepateikiama laiku	VR	maža	3

Įtakos dydžiai: 1 - Katastrofinis, 2 - Kritikalus, 3 - Nežymus.

PR - Produkto Rizika. DR - Darbuotojų Rizika. VR - Verslo rizika.

### 8.2 Valdymo strategija

Identifikavę galimas rizikų priežastis paruošiamą strategiją, kaip jas valdysim.

- Rizikos sumažinimas.

- Rizikos stebėjimas.
- Rizikos valdymas, nenumatytų atvejų planavimas.

## 8.3 Rizikos sumažinimas,

Stengiamasi sumažinti projekto valdymo galioje esančias rizikas.

### 1. Komunikavimo trūkumas .

Svarbu kalbėti, klausti, susitarti dėl vietos ir laiko susitikimam.

Sprendimas variantas būtų surasti žmogų, kuris galėtų sukomunikuoti arba pasiūlyti kita komunikavimo būdą pvz.: komunikuoti internetu.

### 2. Produktas neatitinka reikalavimų .

Sudaryti kontrolinius taškus, atsiskaitymus, kuriuose kiekvienas darbuotojas ar jų grupė parašo kiek nuveikė.

Daryti konferencijas, kuriose sprenžiamos atsiradusios problemos.

### 3. Darbo vietos neužtikrinimas.

Užtikrinti geras darbo salygas. Dažniau daryti apklausas apie darbą, įvertinti iš naujo darbuotojų atlyginimus.

Tam ivykus geriausia būtų pasamdyti atitinkamą darbuotoją ir užtikrinti, kad nepasikartotų senojo darbuotojo išėjimo priežastis, jei įmanoma.

### 4. Nekvalifikuoti darbuotojai.

Tinkamų apklausų užtikrinimas, irodančių įdarbinamojo sugebėjimus.

Peržvelgti projekto dokumentacija ištaisyti netikslumus.

Gavus nekvalifikuotą darbuotoją, svarbu suteikti reikalingą asmenį, kuris suteiktu jam reikalingą informaciją ir prižiūrėtų darbus.

### 5. Prasta produkto kokybe, neišbaigtos produkto dalys.

Įvertinti projekto sudėtingumą, nustatyti ar atitinka galimybes.

Per vėlai įvertinus būtina pranešti užsakovui ir ieškoti bendro sprendimo.

### 6. Klaidų įvėlimas, laiko praleidimas.

Derinti darbo grafika ir išnaudoti papildomą laiką.

Neišvengus reikėtų priskirti papildomą darbuotoją prie užsitiesusio darbo.

### 7. Produkto kokybe blogesnė, programinis nesusiderinamumas.

Kurti konferencijas, kuriose visi parodo ką įvykdė, ir kaip.

Atskiriem darbui priskirti darbuotoją, kuris sugeba komunikuoti.

### 8. Produkto dokumentacija nepateikiama laiku. Sudarinėti grafikus dokumento kūrimui, atsižvelgti į darbininkų norą jį kurti, sudaryti atsiskaitymų taškus, kuriose aiškinamos dokumente susidurtos problemos.

## 8.4 Ką stebėti, atsižvelgti padidėjus rizikai

### 1. Komunikavimo trūkumas .

Užsakovo reakciją į susitarimus, elgseną nuo komunikavimo būdo pakeitimo.



2. Produktas neatitinka reikalavimų .  
Kaip sprendžiamos problemos, kiek padaroma atsiskaitymuose. Kokie klausimai iškeliama konferencijose.
3. Darbo vietos neužtikrinimas.  
Konkurenciją tarp darbo vietų, darbo privalumus, santykius tarp bendradarbių, kaip sprendžiamos iškylusios bėdos esant dideliu spaudimui.
4. Nekvalifikuoti darbuotojai.  
Kaip naujokai sugeba susidoroti su iškylusiomis problemomis, kiek žmonių prašosi į darbą ir kiek patenką, ar apklausoje išangrinėjamos būtiniausios detalės. Taip pat kaip darbuotojai supranta dokumentaciją.
5. Prasta produkto kokybė, neišbaigtos produkto dalys.  
kas vyksta su darbų kokybę didėjant darbų spaudimui, artėjant projekto terminui, stebėti komandos narių spėdimus.
6. Klaidų įvėlimas, laiko praleidimas.  
Kaip sudarinėjamas darbo grafikas, kaip darbuotojai įvertina tai ką jiems reikia padaryti.
7. Produkto kokybė blogesnė, programinis nesusiderinamumas.  
Santykius tarp bendradarbių, komunikavimą esant grupėse, bendro darbo naudą ir atskiro darbo naudą.
8. Produkto dokumentacija nepateikiama laiku. Stebėti kaip rašomas taisomas dokumentas, kada pristabdomas dokumento rašymas ir kada rašant dokumentą atsiranda problemos.

## 8.5 Rizikos valdymas

1. Komunikavimo trūkumas .  
Keisti komuniavimo būdą, truputi pastumėti užsakova judėti pirmyn.
2. Produktas neatitinka reikalavimų .  
Spresti, keisti reikalavimus su užakovu, pastumėti komanda dirbti toliau.
3. Darbo vietos neužtikrinimas.  
Padaryti greitą apklausą kas napatinka darbo vietoje, išnagrinėti apklausą ir ištaisyti neprivatumus, padidinti mažus atlyginimus
4. Nekvalifikuoti darbuotojai.  
Peržiūrėti projekto dokumentą įsitikinti, kad visi projekto aspektai yra išaiškinami. Jei dokumentacija gera tai atleisti labiausiai nekvalifikuotus, kurie projektą labiausiai stabdo.  
Kitu atveju tiesiog patobulinti dokumentaciją .
5. Prasta produkto kokybė, neišbaigtos produkto dalys.  
Nusisamdyti žmogų gerai išmananti neišbaigtų produkto dalių kūrime, peržiūrėti projekto dokumentą įsitikinti, kad visi projekto aspektai yra išaiškinami.  
Susisiekti su užsakovu, keisti produktą.
6. Klaidų įvėlimas, laiko praleidimas.  
Peržvelgti darbų planus, įvertinti grafikus. Jei jie geri, tai atleisti daugiausiai klaidų priverlenčius, projektą stabdačius darbuotojus, nusisamdyti naujus darbuotojus. Jei ne tada įsakyti sudaryti gera darbų planą ir kartoti veiksmą nuo pradžių.
7. Produkto kokybė blogesnė, programinis nesusiderinamumas.  
Nusisamdyti papildomus darbuotojus gerai nusimanantičius ir komunikuojančius ir sudaryti įvesti į sudaryta grupę ar sudaryti naują
8. Produkto dokumentacija nepateikiama laiku.  
Nusisamdyti daugiau žmonių ruošti dokumentaciją.