A picture containing text, clock

Description automatically generated

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

**Studentų atstovybės rinkodaros ryšių duomenų bazė**

Integruotos programavimo aplinkos (su kursiniu darbu)

Kursinis darbas

Atliko PRIf-21/2 grupės studentai:

Emilis Keras

Danielius Šepkus

Erikas Bitovtas

Simona Žvikaitė

Augustas Muralis

Vilnius, 2022

**Turinys**

[Darbų organizavimo ir darbų pasiskirstymo komandoje ataskaita 3](#_Toc133799078)

[Naudotos santrumpos, terminų žodynas 4](#_Toc133799079)

[1. Įvadas 5](#_Toc133799080)

[1.1 Programų sistemos pavadinimas 5](#_Toc133799081)

[1.2 Dalykinė sritis 5](#_Toc133799082)

[1.3 Probleminė sritis 5](#_Toc133799083)

[1.4 Naudotojai 5](#_Toc133799084)

[1.5 Darbo pagrindas 5](#_Toc133799085)

[1.6 Pradinis sistemos langas 6](#_Toc133799086)

[1.7 Funkciniai reikalavimai 6](#_Toc133799087)

[1.8 Naudojamos sistemos 7](#_Toc133799088)

[2. Palyginamoji analizė 8](#_Toc133799089)

[2.1 Projekto organizavimas: reikalavimų perkėlimas į programavimo užduotis pasinaudojus Redmine sistemą Agile Scrum metodiką 8](#_Toc133799090)

[2.2 Aprašyti pasirinktos IDE privalumus, naudą 10](#_Toc133799091)

[2.2.1 Visual Studio 10](#_Toc133799092)

[2.2.2 Replit 11](#_Toc133799093)

[2.2.3 IntelliJ IDEA 13](#_Toc133799094)

[2.2.4 CLion 15](#_Toc133799095)

[2.3 Atlikti automatizuotą testavimo scenarijų ir pritaikyti jūsų sistemai 17](#_Toc133799096)

[2.4 Veikiančios sistemos klaidų paieška (klaidų suradimas pasinaudojus IDE – privalumai) 23](#_Toc133799097)

[2.5 Projekto išteklių susiejimas su kodo saugykla bei pakeitimų versijavimas 26](#_Toc133799098)

[3. Išvados 31](#_Toc133799099)

[4. Naudoti dokumentai ir literatūra 32](#_Toc133799100)

# Darbų organizavimo ir darbų pasiskirstymo komandoje ataskaita

Danielius Šepkus - Projektų išteklių susiejimas su kodo saugykla bei pakeitimų versijavimas. Duomenų bazės UI atvaizdavimas “Adobe Photoshop” įrankiu

Simona Žvikaitė - Projektų išteklių susiejimas su kodo saugykla bei pakeitimų versijavimas. Replit.com IDE privalumų ir trūkumų aprašymas.

Emilis Keras - Įvadas, naudotos santrumpos ir terminų žodyno suvedimas, veikiančios sistemos klaidų paieška, InteliJ IDEA privalumų ir trūkumų aprašymas. Išvados.

Erikas Bitovtas - Projekto organizavimas: reikalavimų perkėlimas į programavimo užduotis pasinaudojus Redmine sistemą Agile Scrum metodiką, CLion privalumų ir trūkumų aprašymas.

Augustas Muralis – Atlikti automatizuotą testavimo scenarijų ir pritaikyti jūsų sistemai, Visual Studio privalumu ir trūkumų aprašymas

# Naudotos santrumpos, terminų žodynas

Application - mobilioji aplikacija.

Chat’as - pokalbių sistema.

User story - naudotojų patirtis.

Sprint - Nustatytas laikotarpis, per kurį turi būti atliktos konkrečios užduotys ir padarytas naudojamas produktas.

Feature - Svarbi įrenginio ar programinės įrangos savybė/funkcija..

Google Cloud Platform (GCP) - platforma, kurioje duomenys, programos laikomos ir naudojamos internetu, vietoj laikymo vienoje fizinėje vietoje.

Testavimas - procesas, kurio metu patikrinama, ar programa visais įmanomais atvejais veikia taip kaip turėtų.

Saugykla - centrinė vieta, kurioje saugomi ir tvarkomi duomenys.

# Įvadas

## Programų sistemos pavadinimas

Studentų atstovybės rinkodaros ryšių duomenų bazė

## Dalykinė sritis

Studentų atstovybės rinkodaros ryšių duomenų bazės kūrimas ir analizė.

## Probleminė sritis

Šio darbo probleminė sritis yra studentų atstovybės rinkodaros ryšių duomenų saugojimo ir tvarkymo procesas. Atstovybė naudoja Google Sheets formatu sukurtą duomenų bazę ir sukuria naujus Sheets failus vietoj to, kad visus duomenis saugotų viename faile. Problema kyla dėl galimų informacijos praradimo ir klaidų rizikos. Taigi, šiame darbe bus nagrinėjama, kaip galima patobulinti duomenų saugojimo procesą ir sukurti efektyvesnę rinkodaros ryšių duomenų bazę. Pagrindinis tikslas yra sukurti efektyvesnę ir saugesnę duomenų bazę, kuri leistų atstovybei tvarkyti rinkodaros ryšius efektyviau ir sumažintų klaidų riziką.

## Naudotojai

Programėlės naudotojai bus:

1. Studentų atstovybės rinkodaros komiteto nariai, kurie turės galimybę pasiekti duomenų bazę surasti jiems norimus kontaktus arba juos pridėti, pasiekti informaciją apie prieš tai vykusias komunikacijas ir matyti savo priskirtas užduotis nuo koordinatorių.  
 2. Fakulteto Studentų atstovybės rinkodaros Koordinatoriai kurie turės visas prieš tai paminėtas teises bei galės sukurti bei pašalinti prisijungimus savo fakulteto komiteto nariams, pridėti naujus renginius ir prisegti tam tikras užduotis nariams ir matyti atliktų/neatliktų darbų statistiką.  
 3. Studentų atstovybės rinkodaros Vadovas/ė, turės visas prieš tai paminėtas teises, galės sukurti koordinatorių bei vadovų prisijungimus ir pasiekti visus duomenų bazės naujus/senus įrašus.  
 4. Plėtotojas/programuotojas, turės teisę matyti visą duomenų bazės kodą, taisyti/redaguoti jį, bei matyti IT pagalbos užklausas.

## Darbo pagrindas

Šio darbo tikslas yra sukurti efektyvią studentų atstovybės rinkodaros ryšių duomenų bazę, kuri leistų tvarkyti ir analizuoti atstovybės veiklą bei sumažinti klaidų riziką.

Darbo uždaviniai apima:

* Pasirinkti ir palyginti Integruotas programavimo aplinkas, kurios būtų patogiausios duomenų bazių kūrimui ir valdymui;
* Išsirinkti ir įdiegti tinkamiausią projektų valdymo programą, kuri naudoja Agile metodus, siekiant efektyviai valdyti darbą ir užtikrinti projekto sėkmę;
* Pasirinkti ir susieti versijų kontrolės platformą su pasirinkta Integruota programavimo aplinka, siekiant palengvinti programų testavimą ir integraciją;
* Kūrus duomenų bazę, atsižvelgti į gerą duomenų bazių kūrimo praktiką ir projektavimo principus, siekiant užtikrinti duomenų saugumą ir patikimumą;
* Ištirti ir įvertinti duomenų analizės galimybes ir funkcionalumą, siekiant užtikrinti patogų duomenų valdymą ir naudojimą.

Šie uždaviniai bus įgyvendinami naudojant efektyvias ir patogias Integruotas programavimo aplinkas, kad būtų pasiektas kuo didesnis produktyvumas ir sumažintos klaidų rizikos.

## Pradinis sistemos langas

Pradiniame sistemos lange matosi:

A screenshot of a computer

Description automatically generated with medium confidence

## Funkciniai reikalavimai

Funkciniai reikalavimai nurodo užduotis, kurios turi būti įgyvendinamos sistemoje, jog duomenų bazė veiktų korektiškai ir būtų paruošta naudojimui.

|  |  |
| --- | --- |
| Nr. | Reikalavimai |
| 1. | Studentų atstovybės vadovas/ė gali sukurti prisijungimus vadovo ir koordinatoriaus rolėms. |
| 2. | Studentų atstovybės koordinatoriai gali sukurti prisijungimus savo fakulteto komiteto nariams. |
| 3. | Duomenų bazės naudotojas turi tūrėti galimybę prisijungti prie duomenų bazės su naujais prisijungimais bei galimybę juos pakeisti. |
| 4. | Duomenų bazės naudotojas turi tūrėti galimybę pakeisti savo asmeninius duomenis. |
| 5. | Duomenų bazės naudotojas turi tūrėti galimybę pridėti naujos įmonės kontaktus. |
| 6. | Duomenų bazės naudotojas turi tūrėti galimybę pridėti naujus įrašus prie įmonės aprašymo apie nauja bendradarbiavimą. |
| 7. | Studentų atstovybės rinkodaros koordinatorius turi tūrėti galimybę sukurti naują renginį duomenų bazėje, priskirti reikiamo rėmimo renginiui, bei priskirti komiteto narius, kurie bus atsakingi už renginio rėmimo paiešką. |
| 8. | Duomenų bazės naudotojas turi turėti galimybę naudotis chat’u. |
| 9. | Duomenų bazės naudotojas turi turėti galimybę parašyti žinutę chat’e. |
| 10. | Studentų atstovybės koordinatorius turi tūrėti galimybę pasiekti savo fakulteto arba renginio darbų statistiką. |
| 11. | Duomenų bazės naudotojas turi turėti galimybę gauti techninę pagalbą. |
| 12. | Studentų atstovybės duomenų bazės Vadovas/ė turi tūrėti galimybę matyti visos atstovybės darbų statistiką |
| 13. | Duomenų bazės naudotojas turi tūrėti galimybę naudoti paieškos filtrus įmonių paieškai |
| 14. | Duomenų bazės naudotojas turi tūrėti galimybę matyti ar dėl tam tikro renginio buvo susiektą su įmonę |
| 15. | Duomenų bazės sistema turi automatiškai perkelti jau įvykusius renginius į “Praeiti renginiai”. |

## Naudojamos sistemos

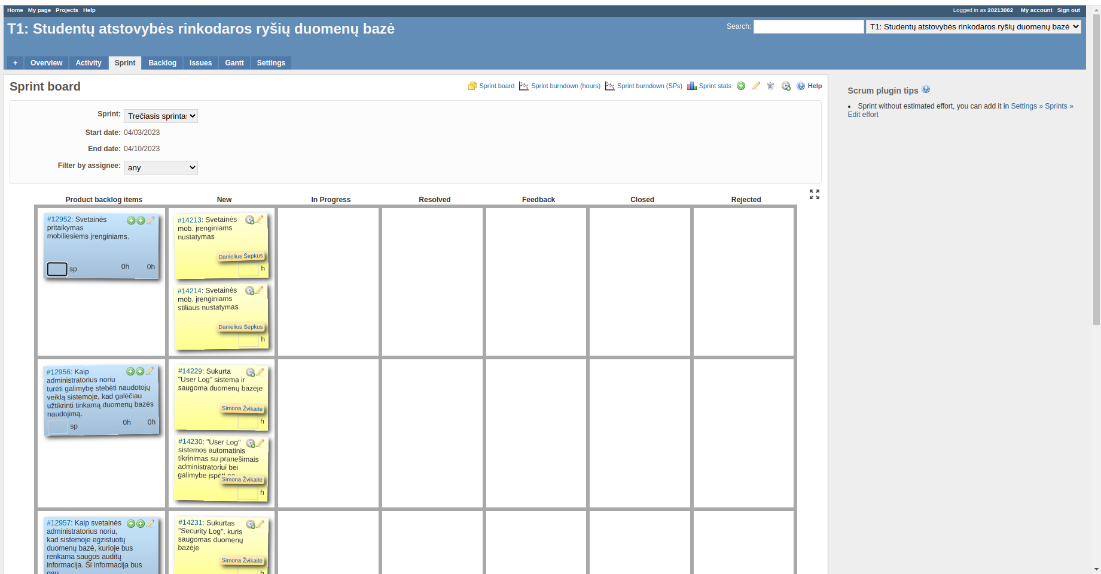
Kuriant maisto prekių užsakymo programėlę buvo naudojamas šios sistemos:

1. Redmine, ProjectMananger – tai lankstūs internetiniai projektų valdymo įrankiai. Šie įrankiai suteikia galimybę detaliai ir vaizdžiai sukonstruoti norimą projektą. Šiose sistemose detaliai ir nuosekliai išvardijama veiksmų seka ir atsakomybės.
2. GitLab,GIT – tai versijavimo aplinkos, kuriose programinis kodas yra nuosekliai atnaujinamas ir įtraukiamas į sistemą. Šios sistemos palengvina darbo įgyvendinimą, kadangi visi atliekami pakeitimai matomi sistemoje.
3. Ghost inspector, Automation – tai automatizuoti testavimo įrankiai, kurie padeda įsitikinti, jog sistema neturi klaidų, bei visos jos dalys būtų pasiekiamos vartotojo naudojimui.
4. Visual Studio, CLion, Replit, InteliJ IDEA – tai integruotos programavimo aplinka, kurios suteikia galimybę naudotis įvairiomis programavimo funkcijomis, kurios naudojamos būtent kuriant ir testuojant programą.

# Palyginamoji analizė

## 2.1 Projekto organizavimas: reikalavimų perkėlimas į programavimo užduotis pasinaudojus Redmine sistemą Agile Scrum metodiką

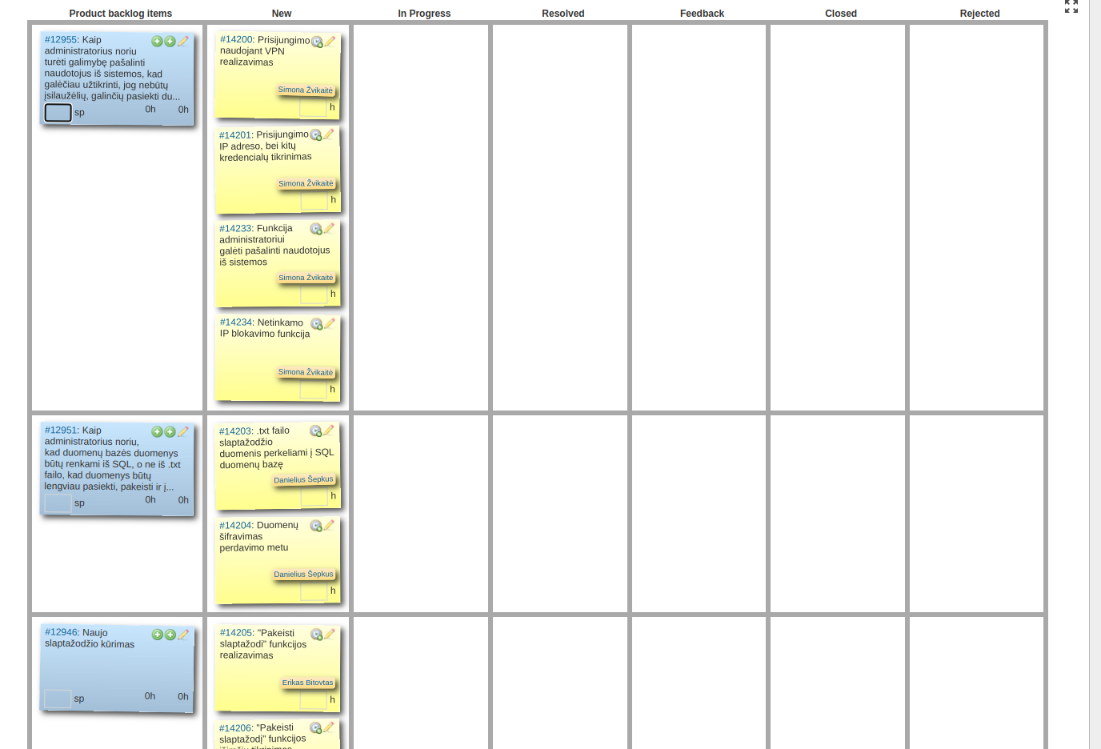
Darbo organizavimui pagal Agile Scrum metodiką pasitelkėme sistemą „Redmine“. Ji puikiai atitiko mūsų darbo būdo specifiką. „Redmine“ sistemoje darbas yra skirstomas į „Sprintus“, t.y. nurodytus laiko tarpus, per kuriuos turi būti realizuotos tam tikros programos funkcijos bei galimybės.



Kiekvienam „sprintui“ yra numatomas sąrašas galimybių, kurias reikia realizuoti programoje. Šios galimybės yra aprašomos „user story“, t.y. naudotojo istorijas. Jose yra aprašomas tikslinis galimas naudotojas, bendras reikalavimas programai iš jo pusės ir nauda, kurią jis gauna, kai šis reikalavimas bus įgyvendintas. „User story“ yra rašomos tokiais principais:

* Tikslinis naudotojas turi būti asmuo, kuris labiausiai tikėtina, jog naudosis mūsų programa.
* Bendrasis reikalavimas turi apibūdinti bendrus tikslinio naudotojo poreikius, ne konkrečias funkcijas.
* Aprašyta nauda turi įvardinti, kodėl tikslinis naudotojas galėtų iškelti tokį reikalavimą.

Po to iš bendro reikalavimo išplaukia funkcijos, kurias programa turės gebėti atlikti. Šios funkcijos Agile Scrum metodikoje yra vadinamos „feature“. Jos apibūdina funkcijas, kurios bus įdiegtos programoje. Šios funkcijos turi atitikti aprašytus „user stories“, t.y. jos turi aprašyti užduotis, kurios atitiks tikslinio naudotojo bendrus reikalavimus.

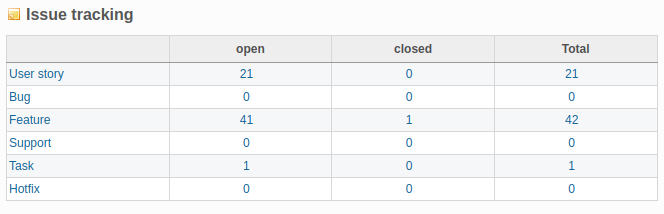


Pasitelkdami šią platformą, mes turime geresnį supratimą apie tai, kaip mūsų komandos nariams pavyksta arba nepavyksta atlikti jam paskirtas užduotis ir suteikti jam pagalbą jas atliekant. Remdamiesi šia informacija, galime esamo „sprinto“ metu aprašyti kitą „sprintą“, kuris geriau atitiks mūsų darbo tempą bei pobūdį.

Tarp projektų valdymo sistemų yra ir kita alternatyva, vadinama Jira. Ja naudojasi tokios didžiosios įmonės, kaip Amazon, GitLab bei Microsoft. Tačiau savo darbui mes visgi nusprendėme pasirinkti Redmine. Tam darė įtaką keli veiksniai:

* Redmine yra atviro kodo, taigi nemokamas ir lengvai modifikuojamas pagal mūsų poreikius. Tuo tarpu Jira yra uždaro kodo, komercinis produktas.
* Redmine naudotojo sąsaja yra paprastesnė naudojimui ir įsisavinimui, nei Jira.
* "Redmine" yra labai lengvai pritaikoma ir suteikia daug lankstumo, kad galėtumėme ją pritaikyti prie konkrečių savo komandos reikalavimų. "Jira" taip pat suteikia tam tikrų pritaikymo galimybių, tačiau jos gali būti ne tokios lanksčios kaip "Redmine".

Redmine pagalba mes sukūrėme 3 sprintus, 21 userstory ir 42 features.



## 2.2 Aprašyti pasirinktos IDE privalumus, naudą

### 2.2.1 Visual Studio

**Visual Studio privalumai:**

* Universalus kalbų palaikymas
* Išsamus įrankių rinkinys

A screenshot of a computer program

Description automatically generated with medium confidence

* Pritaikoma sąsaja
* Integravimas su versijų valdymo sistemomis
* Debesų palaikymas
* Išplečiamumas
* Reguliarūs naujinimai
* Bendradarbiavimo funkcijos
* Mobiliųjų įrenginių kūrimo palaikymas
* Stipri bendruomenė ir palaikymas
* Integruoti testavimo įrankiai
* Kodo generavimas ir pastoliai

A screenshot of a computer

Description automatically generated

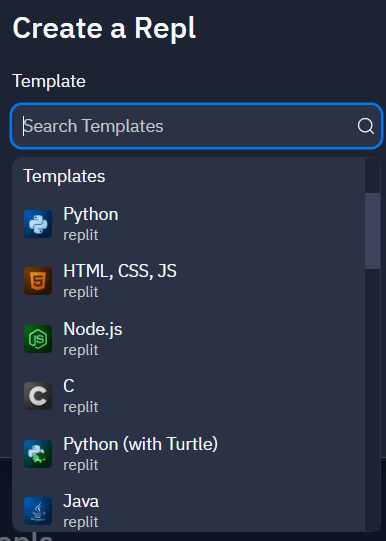
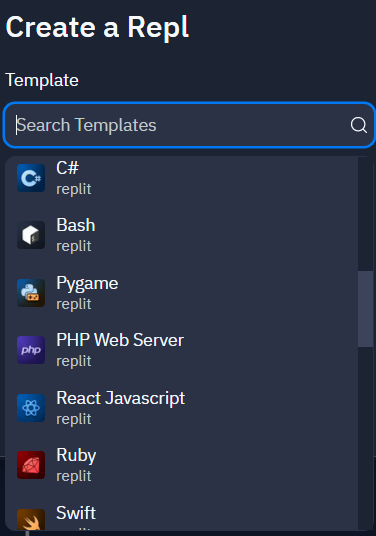
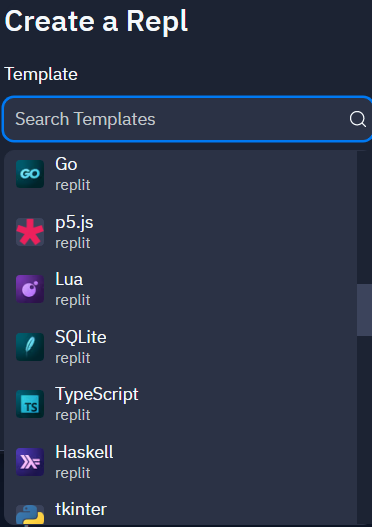
**Visual Studio trūkumai:**

* Daug išteklių reikalaujanti
* Statesnė mokymosi kreivė
* Kaina
* Platformos palaikymas
* Galimybė išsipūsti
* Ribotas ne „Microsoft“ technologijų palaikymas
* Patentuota programinė įranga
* Lėtas paleidimo laikas
* Suderinamumo problemos
* Didelis diegimo plotas
* Retkarčiais pasitaikančios našumo problemos

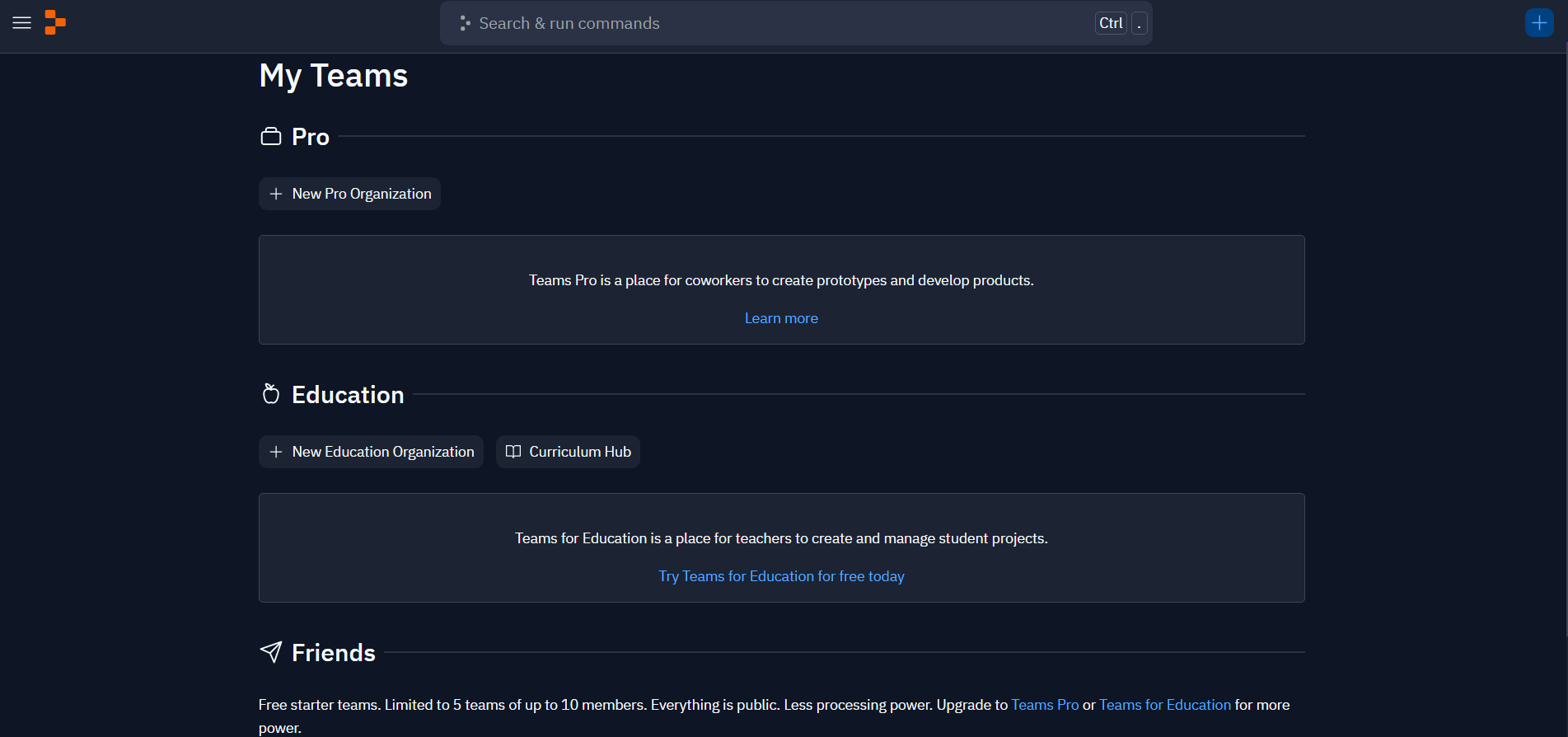
### 2.2.2 Replit

**Replit privalumai:**

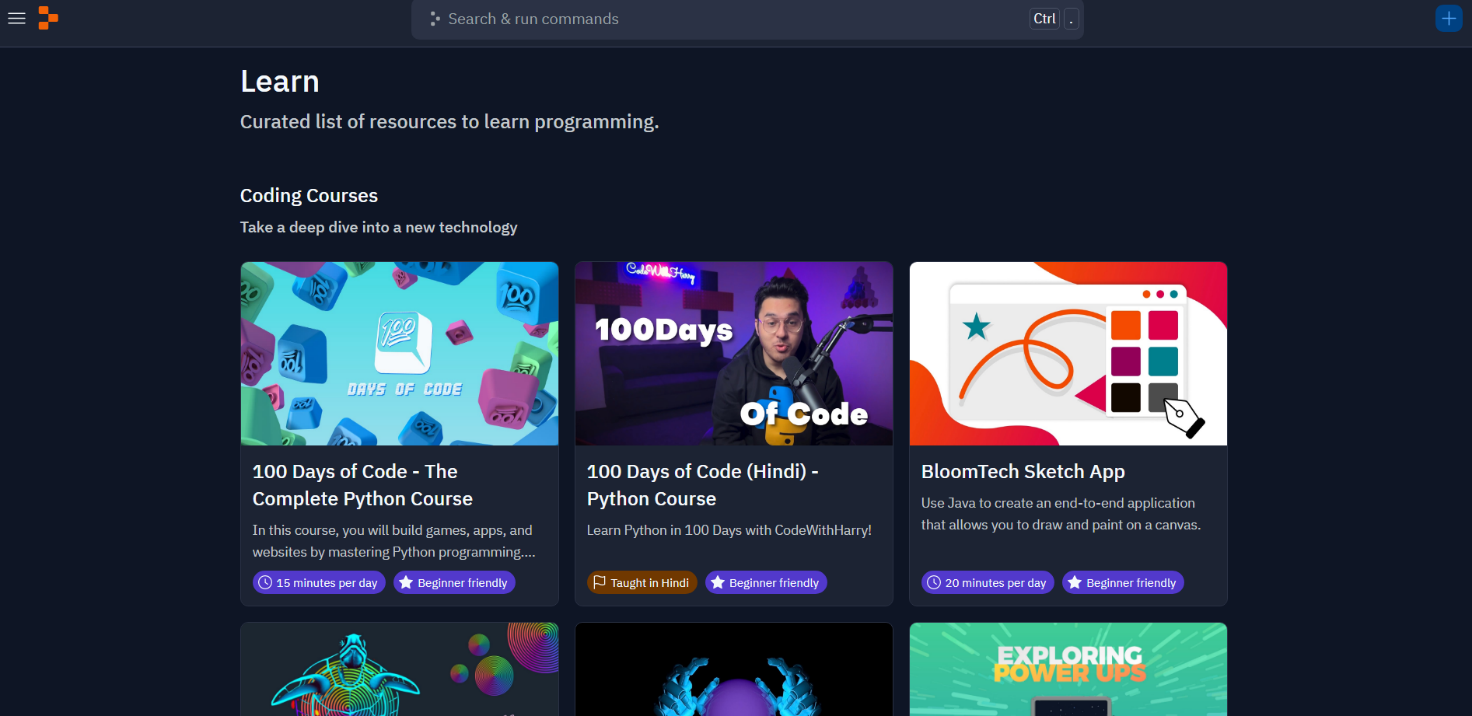
* Yra nemokama ir leidžia naudoti daugelį pagrindinių funkcijų.
* Gali būti naudojama bet kur ir bet kada, nereikalaujant jokių papildomų diegimų.
* Palaiko daugybę programavimo kalbų, įskaitant Python, JavaScript, Ruby ir kt

* Yra puikus įrankis kodo dalijimui su kitais programuotojais. Tai leidžia kitiems pamatyti jūsų kodą ir jį išbandyti.



* Suteikia daugybę projektų valdymo įrankių, įskaitant versijų kontrolę ir eilutes kodui rašyti.
* Siūlo integruotą vystymosi aplinką, kurioje galite išbandyti kodą ir matyti, kaip jis veikia realiame laike.
* Siūlo talpinti serverius internete. Tai ypač naudinga, jei kuriate internetinius projektus arba internetinius žaidimus.
* Yra puiki priemonė mokantis programuoti, nes jame galite kurti savo projektus, dalintis kodu su kitais ir išmokti iš kitų programuotojų.



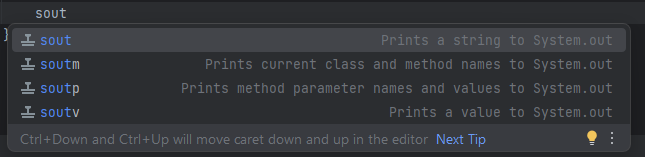
**Replit trūkumai:**

* Nors Replit siūlo daugybę programavimo kalbų, ji turi tam tikrų apribojimų, pvz., apribojimą dėl failų dydžio, kaip ir bet kokia internetinė programavimo aplinka.
* Norint naudoti Replit, reikia turėti prisijungimo paskyrą. Tai gali būti trukdymas tam tikroms grupėms, pvz., žmonėms, kurie nori naudoti Replit anonimiškai.
* Nors Replit siūlo galimybę talpinti serverius, jų talpinimo galimybių skaičius yra ribotas, o tai gali būti trukdymas tam tikriems projektams, kuriems reikia didesnių resursų.
* Nors Replit siūlo galimybę dalintis kodu su kitais, kai kurie programuotojai gali bijoti, kad kiti gali pamatyti jų kodą arba duomenis. Replit taip pat nepalaiko visų galimybių atskirti privatų kodą ir viešą kodą.
* Replit yra internetinė programavimo aplinka, todėl, jei neturite stabilios interneto ryšio, gali būti sunku naudotis platforma.

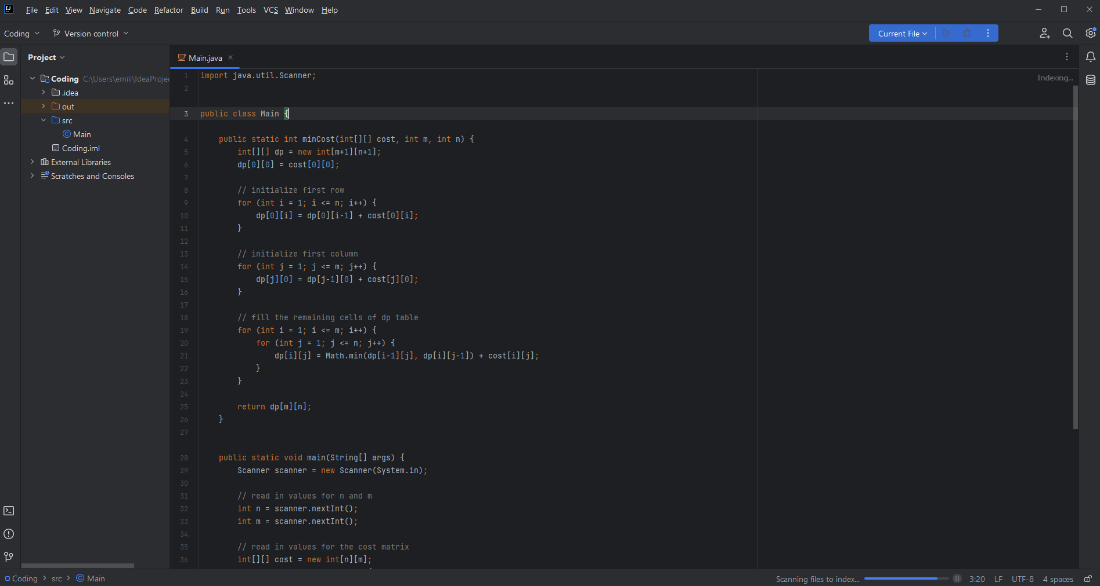
### 2.2.3 IntelliJ IDEA

**IntelliJ IDEA privalumai**

* "Markdown" dokumentų palaikymas leidžia lengvai rašyti gerai suformatuotus dokumentus
* "OpenApi" specifikacijų palaikymas palengvina API projektavimą ir priežiūrą
* PlantUML diagramų palaikymas neleidžia jūsų komandai skųstis, kad diagramos yra pernelyg nuobodžios
* Kodo užbaigimas / aiškumas



* Kodo susiejimas ir nuorodos
* Įrankių integravimas
* Sintaksės paryškinimas
* Klaidų taisymas
* Geras derinimo palaikymas.



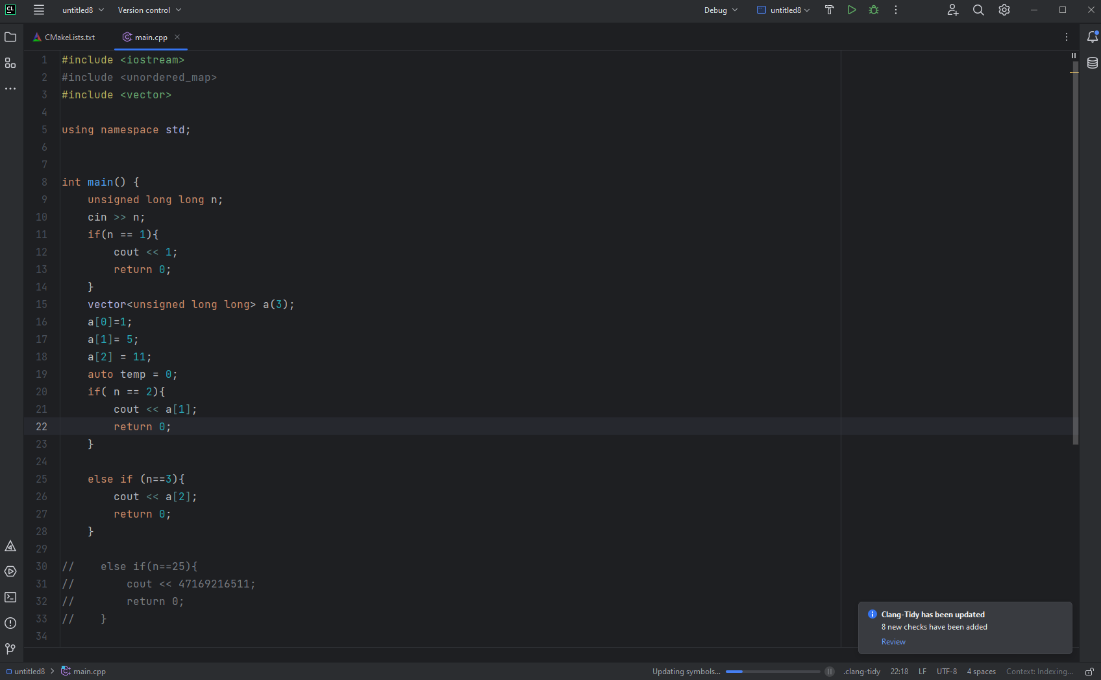
* Gera vartotojo sąsaja.
* Lengva navigacija.

**IntelliJ IDEA Trūkumai**

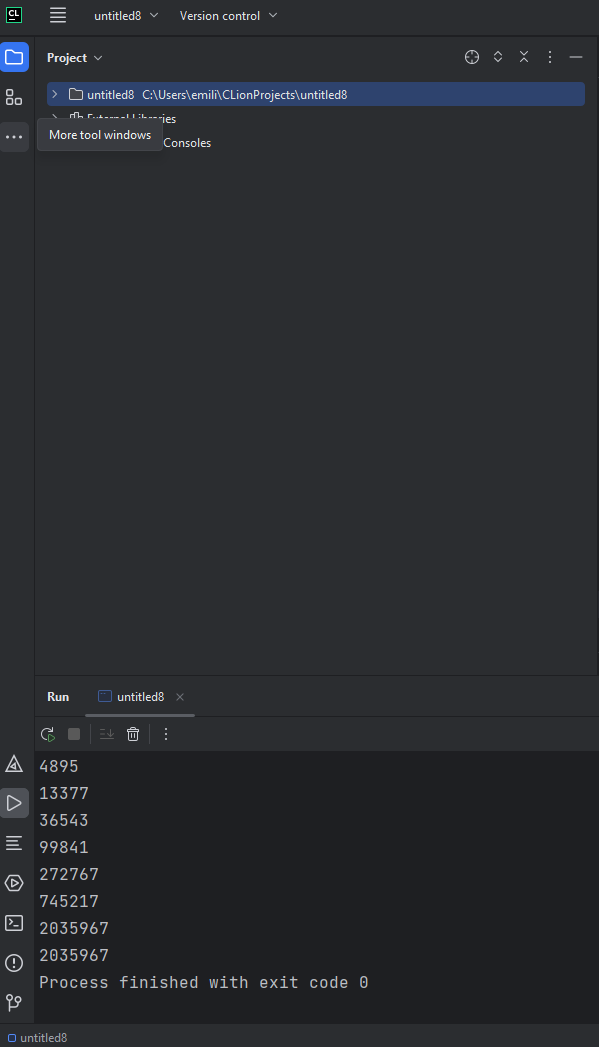
* Trūksta funkcijos "Rasti ir pakeisti visuose failuose vienu metu
* Nepalaikomi kai kurie "Eclipse" spartieji klavišai
* Kartais "Git" nesinchronizuojama
* Atminties valdymas
* Skirtingiems projektams skirti specialūs naudotojo profiliai
* Galimybė lanksčiai pridėti naujų aspektų prie esamų projektų

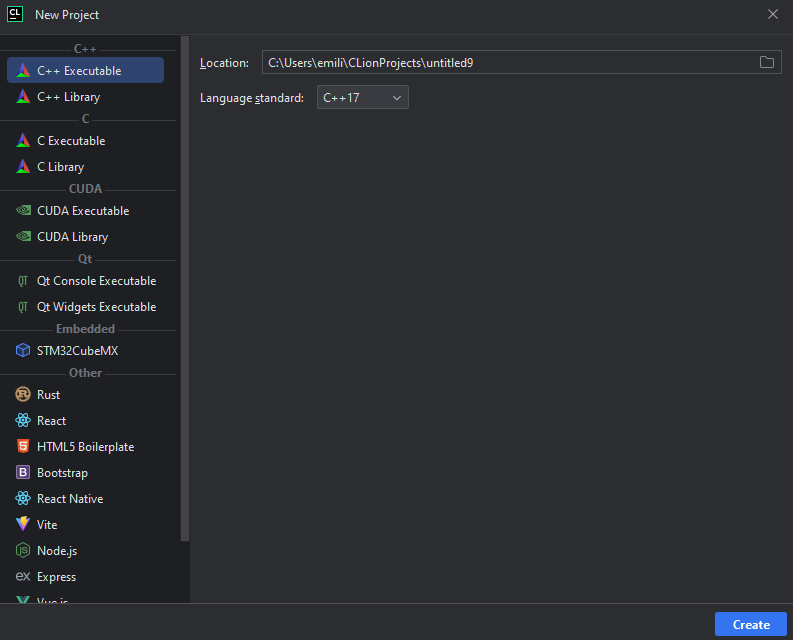
### 2.2.4 CLion

**CLion Privalumai**



* Patogi vartotojo sąsaja
* Palaiko VIM greitam kodavimui
* Išmanusis kodo užbaigimas
* Tarpplatforminis palaikymas
* Padalytų langų funkcija
* Palaiko C ir C++
* Suteikiama nemokama studentų licencija





**CLion Trūkumai**

* Neturi integruoto kompiliatoriaus
* Kompilatoriaus diegimas gali būti sudėtingas pradedantiesiems
* Blogai išplėtota Makefiles palaikymo sistema

## 2.3 Atlikti automatizuotą testavimo scenarijų ir pritaikyti jūsų sistemai

Testuosime 5 Click Up funkcijas (Board Funkcijos testavimas, Sukurti Space be template, „DOC“ funkcijos funkcianalumas, Sukuri Space su template, Mind Map funkcijų testavimas) Remiantis redmine įkeltomis užduotimis pradiniai testai atrodytų taip:

• Agile Scrum Board funkcijos testavimas:

1. Prieš testą buvo sukurtas ir pasirinktas naujas Agile Scrum tipo space, testavimas:
2. Paspausti ant Board skilties.
3. Open skiltyje paspausti Create Task mygtuką.
4. Įrašyti Task pavadinimą.
5. Pasirinkti Start Date - 2023-04-26.
6. Pasirinkti Due Date - 2023-05-06.
7. Pasirinkti Set priority mygtuką ir nustatyti High priority.
8. Išsaugoti Task, spaudžiant SAVE.
9. Sukurti subtask naudojant Create subtask mygtuką.
10. Parašyti subtask pavadinimą.
11. Išsaugoti pakeitimus.
12. Nutempti sukurtą Task į In Progress skiltį.

• Sukurti space be template

1. Paspausti kairiajame šone esantį mygtuką „New space“.
2. Į lauką „space name“ įvesti space pavadinimą.
3. Pasirinkti space ženklelio spalvą.
4. Pasirinkti privatumo nustatymus „space Workspace“.
5. Spausti „Next“, kadangi template nesirenkame.
6. Pasirinkti numatytuosius „ClickApps“
7. Pasirinktu numatytuosius views nustatymus
8. Patvirtinti pasirinkimą
9. Spausti „create Space“

• ClickUp "DOC" funkcijos funkcianalumas

1. Sukurti naują Doc Failą.
2. Ištrinti seną Doc Failą.
3. Prirašyti teksto į failą.
4. Pakeisti pavadinimą failo.
5. Pridėti failo komentarą.
6. Pažymėti komentarą kaip "Solved"
7. Priskirti failą prie Task.
8. Pridėti template į docs skiltį.
9. Pridėti "Backgroud" į doc failą.
10. Eksportuoti failą PDF formatu

• Sukurti space su template

1. Atidaromas ClickUp puslapis
2. Paspaudžiamas mygtukas "New Space"
3. Naujame atidaryte lange paspaudžiamas mygtukas "Template"
4. Testuojama ar vienas iš pasirinktų šablonų turi tolimesnį perėjimą
5. Paspaudžiama ant "Class Assignment" šablono
6. Paspaužiama ant atnaujinto ekrano mygtuką "Use Template"
7. Testuojamas mygtukas "Customize import items", kad būtų sužinota kur jis nuveda
8. Paspaudžiamas mygtukas "Use Template", kad sukurti šabloną
9. Testuojama ar "Fall 2021" kategorija egzsituoja šitame puslapyje
10. Paspaudžiama ant kategorijos mygtuko "Spring 2022"
11. Testuojamas "History 102" mygtukas dėl veikimo
12. Paspaudžiama ant vieno iš duotu failų "History 102" skiltyje
13. Paspaudžiama ant mygtuko "To Do" kuris pasikeičia į "In Progress"
14. Paspaudžiama ant mygtuko "In Progress" kuris pasikeičia į "Complete"
15. Paspaudžiama šalia mygtuko ant kurio paspausta per 12 ir 13 žingsnius
16. Testuojamas iššokes laukelis po 14 žingnsio ir perduodama informacija, kad užfiksuotas tekstas yra "In Progress"
17. Uždaromas laukelis atidarytas per 14 žingsnį
18. Uždaromas atidarytas failas iš "History 102" skilties
19. Uždaroma "History 102" skiltis

• Mind Map Funkciju testavimas

1. prisijungiame prie ClickUp puslapio
2. užeiname i "Borad" Skilty
3. sukuriam nauja Task
4. sukuriam jam subtask
5. sukuriam papildoma subtask
6. sukuriam nauja task
7. sukuriam jam subtask
8. sukuriam papildoma subtask
9. sukuriam nauja task
10. atsidarome "Mind Map" skilty
11. sukuriame mind mape nauja subtask
12. naujai sukurta sub task perkeliame i task
13. tada ji perkeliame i subtask kitame task
14. sukuriame nauja subtask
15. perkeliame subtask i task

“Ghost Inspector” tai automazituota testu sistema, kur karta įrašius testa ji galima automazituoti pakeitus duomenis ghost inspectoriaus nustatymuose ir testus vykdyti automatiskai.

A screenshot of a computer

Description automatically generated with medium confidence

Testo įrašymo pradėjimui naudojamas plėtinys “Google Chrome” naršyklėje

A screenshot of a login form

Description automatically generated with medium confidence

A screenshot of a web page

Description automatically generated with medium confidence

Pasinaudojus “Ghost Inspector” siūlomais irankiais, kaip pvz.: “Record operations”, “Make assertions” ir t.t. įrašome testa, įrašius testa “Ghost Inspector” veikla sustabdoma ir laukiama testo ataskaitos

A screenshot of a computer

Description automatically generated with low confidence

Ataskaitoje matome ar testas buvo ivikdytas iki galo, ar buvo atlikti visi testo zingsniai, kiek laiko uztruko, ar viskas buvo ivykdyta be klaidu ir tesingai.

A screenshot of a computer

Description automatically generated with medium confidence

Testui pavykus atsiranda galimybe jivykdyti automatiskai keiciant ivestus duomenis

“Selenium” yra “Ghost Inspector” alternatyva kuri veikia tuo panašiu principu, bet yra nemokama, už ja moketi reikia tik atliekant testus, kita vertus už “Ghost Inspector” reikia mokėti menesini arba metinį mokesti, priklausomai nuo pasirinkto plano.

“Selenium” testuoja naršyklės puslapius, bei naršyklės tipo programas, kaip “FaceBook”, “Twitter” ir t.t.

A screenshot of a computer

Description automatically generated with medium confidence

“Selenium” kaip ir “Ghost Inspector” turi naršyklės plėtinius kurie leidžia lengviau įrašyti testus, kaip jie turėtų būti atlikti, skirtingai nuo “Ghost Inspector”, “Selenium” gali veikti ir “JavaScript” parašytu kodu.

Selenium leidžia pasirinkti 3 skirtingus komandų tipus: pridėti duomenis, pasirinkti duomenis iš kelių variantų, paspausti mygtuką.

Todėl automatinių testavimo įrankių paklausa padeda supaprastinti testavimo procesą. Be šių įrankių, rankinis vykdymas tinkama seka užtruktų ilgiau ir būtų pridaryta daugiau klaidu. „Ghost Inspector“ siūlo patogesnę vartotojo sąsają, tačiau tai kainuoja, o „Selenium“ reikalauja staigesnės mokymosi kreivės, sunkiau perprasti ją ir tai reikalauja papildomo laiko, tačiau ji yra visiškai nemokama naudojant debesies prieigą.

## 2.4 Veikiančios sistemos klaidų paieška (klaidų suradimas pasinaudojus IDE – privalumai)

(Allain, 2022)

(Microsoft, First look at the Visual Studio Debugger, 2022)

Kūrimo procesą galima stebėti „Visual Studio“ vartotojo sąsajos apačioje esančiame išvesties lange. Čia rodomos klaidos, įspėjimai ir kūrimo operacijos.

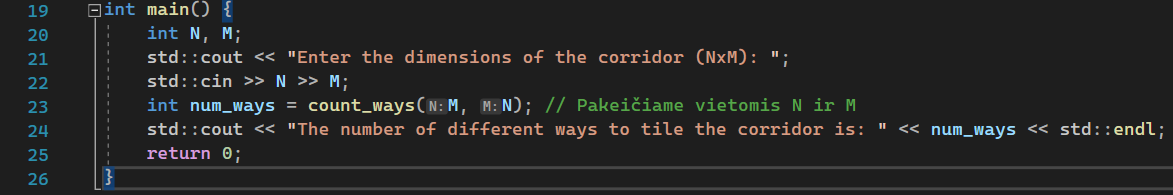
Rezultatų lange po redaktoriumi yra du langai su skirtukais: išvesties langas, kuriame yra neapdorota kompiliatoriaus išvestis (įskaitant klaidų pranešimus); ir klaidų sąrašo langas, kuriame pateikiamas rūšiuojamas ir filtruojamas visų klaidų ir įspėjimų sąrašas.

Klaidų sąrašo lange galima atlikti automatinę Bing paiešką, kad gauti daugiau informacijos apie klaidą arba įspėjimą.

“[Quick Actions](https://docs.microsoft.com/en-us/visualstudio/ide/quick-actions?view=vs-2022)” gali būti naudojami visur, kur kodo analizatoriai nustato, kad yra galimybė pataisyti, pertvarkyti arba patobulinti kodą.

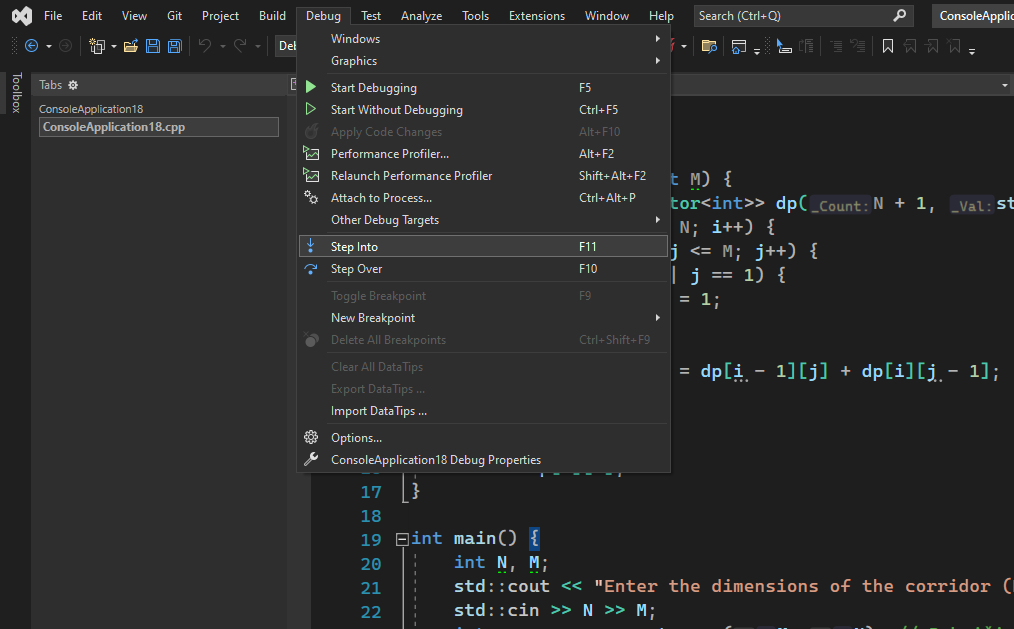
Be failo formatavimo tarpams, įtraukoms ir kt., “Code Cleanup” taip pat taiko apibrėžtų kodo stiliaus susitarimų rinkinį.

“[Breakpoints](https://docs.microsoft.com/en-us/visualstudio/debugger/using-breakpoints?view=vs-2022)” nurodo, kur „Visual Studio“ turėtų sustabdyti vykdomą kodą, kad galima pažvelgti kintamųjų reikšmes, atminties elgseną arba tai, ar paleidžiama kodo šaka, ar ne.

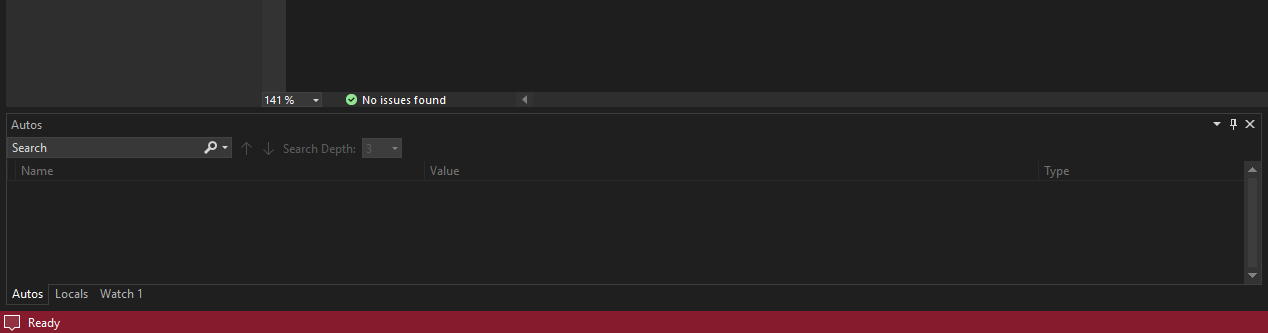


„Visual Studio“ įdiegia „Microsoft“ vienetų testavimo sistemas tiek valdomam, tiek vietiniam kodui. Naudokite vieneto testavimo sistemą, kad sukurtumėte vienetų testus, vykdytumėte juos ir praneštumėte apie šių testų rezultatus.

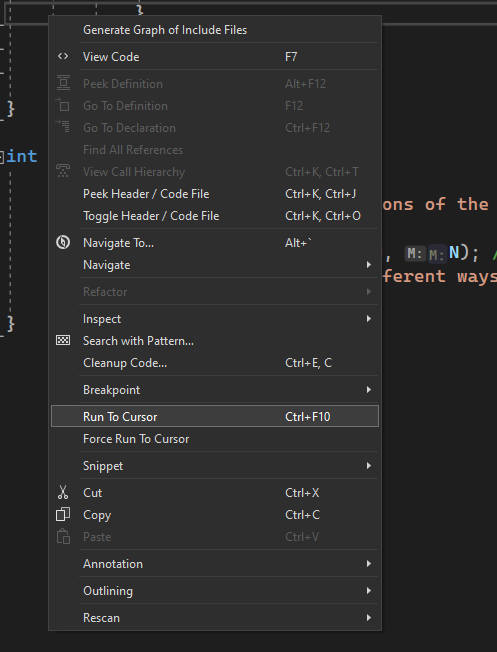
Naudojant „step into“ galite greitai patekti į funkciją, nereikia surasti funkcijos ir nustatyti laikiną pertraukos tašką (arba naudoti „run to cursor“).



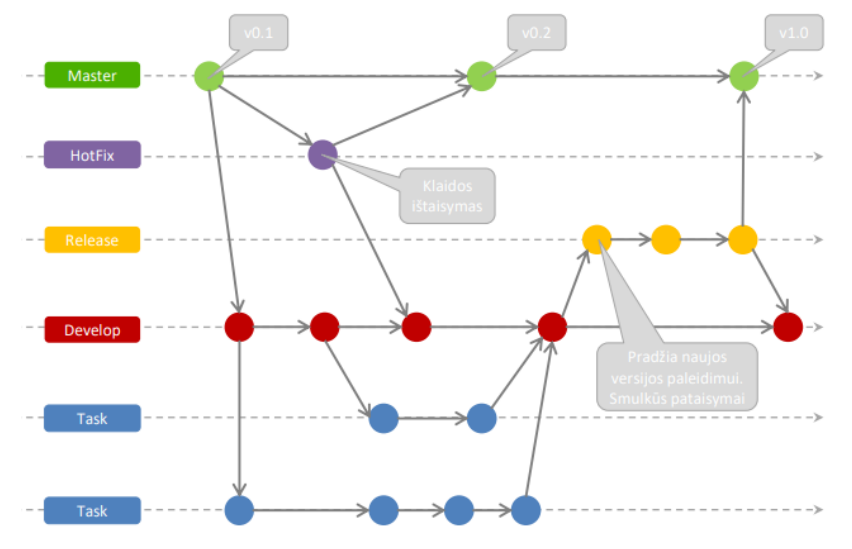
„Autos“ lange gali būti rodomas funkcijos įvertinimo rezultatas



Kai reikia išmesti pertraukos tašką arba norint išvengti ėjimo po vieną žingsnį per daug kodo yra galimybė naudoti “Run To Cursor”.

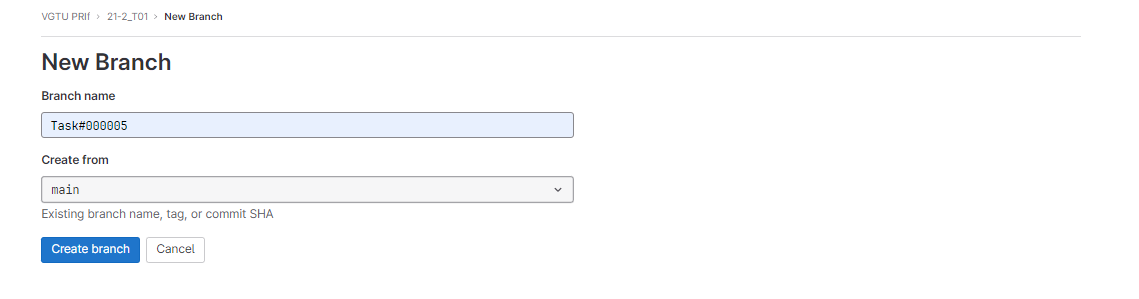


## 2.5 Projekto išteklių susiejimas su kodo saugykla bei pakeitimų versijavimas

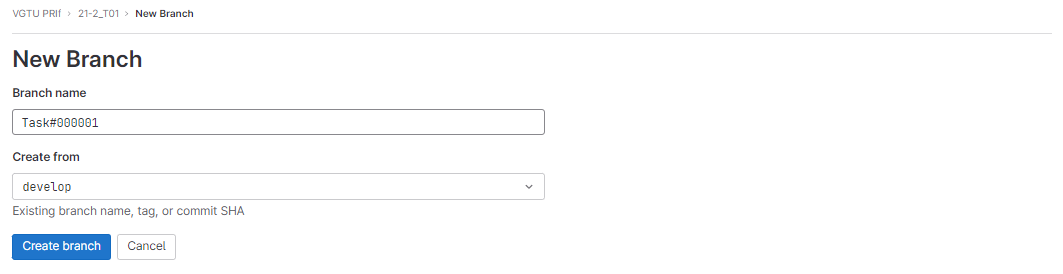


Versijų valdymo sistema - tai programinė įranga, kuri išsaugo failo ar projekto pakeitimus ir saugo šių pakeitimų istoriją, kad būtų galima juos atkurti ar analizuoti ateityje. Atliekant programos ar projekto pakeitimus, versijos numeris paprastai padidėja.

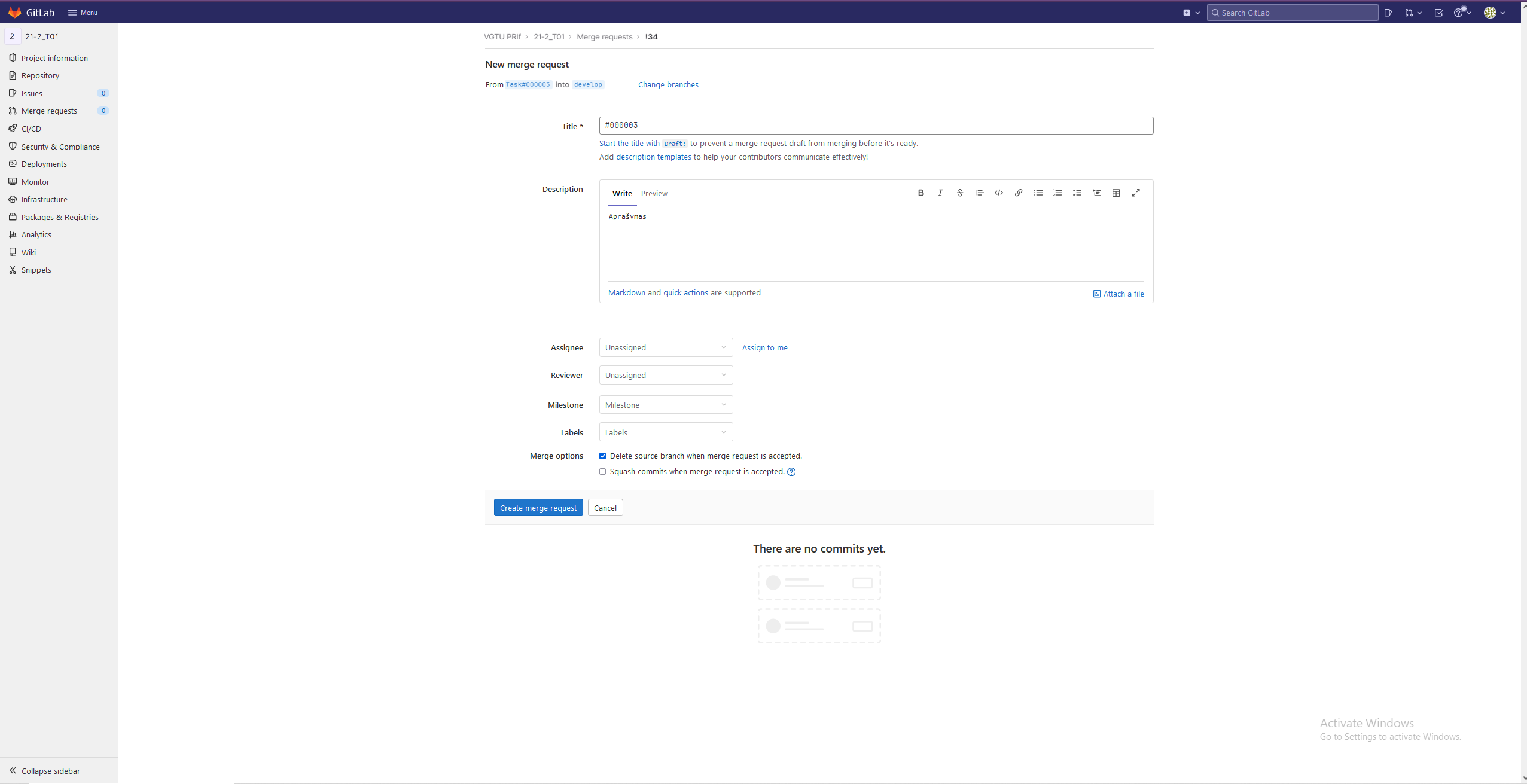
Savo projekte versijų valdymui naudojome GitLab sistemos Gitflow Workflow modelį. GitLab leidžia naudoti kelias šakas, kuriose gali dirbti skirtingi žmonės, o paskui visas jas galima sujungti į vieną pagrindinę projekto / programos šaką. "Main" (master) šaka jau buvo sukurta, todėl vienas iš komandos narių sukūrė "Develop" šaką, kilusią iš "Main" šakos.



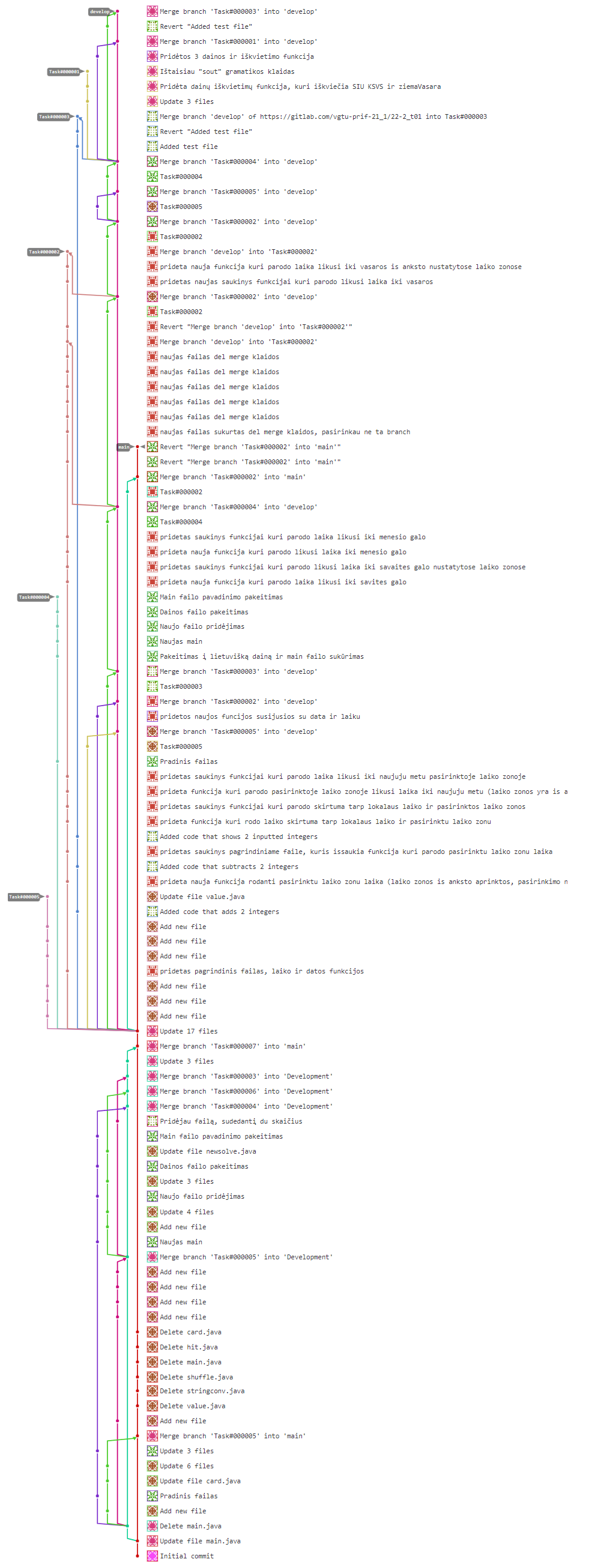
Po to kiekvienas komandos narys sukūrė savo šaką, pavadintą "Task#00000x (užduoties numeris)". Šios šakos buvo sukurtos iš "GitLab" šakos "Develop". Ir atlikinėjo pasirinktas funkcijas.



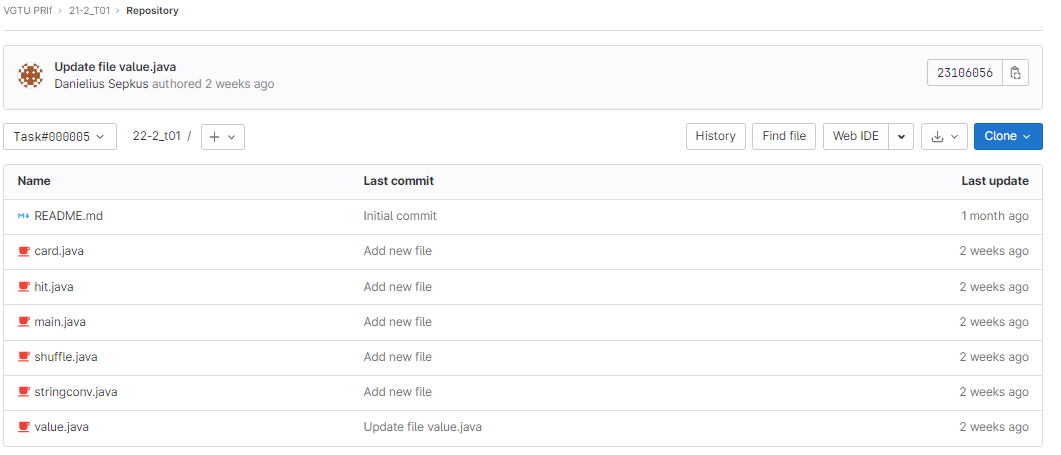
Atlikę visus reikiamus pakeitimus, pavyzdžiui, pakeitę kodą arba į "GitLab" saugyklą įtraukę naujų failų, ir ištaisę visas klaidas, komandos nariai sujungė savo individualią (-as) užduoties šaką (-as) su pagrindine "GitLab" atšaka "Develop". Taip buvo užtikrinta, kad visi komandos narių atlikti pakeitimai būtų sujungti į pagrindinę projekto šaką.



Galiausiai, kai visi pakeitimai buvo sujungti su "Develop" šaka ir buvo nuspręsta, kad viskas veikia tinkamai ir yra paruošta, "Develop" šaka buvo sujungta su "Main" (master) "GitLab" šaka. Vykstant šiam procesui buvo sudarytas grafikas, kuriame vizualiai parodytos šakos ir jų ryšys, rodantis, kad užduočių šakų pakeitimai buvo sujungti į "Develop" šaką, o po to - į "Main" šaką. Dėl kilusių problemų ir įvairių nesusipratimų (pvz.: vienas komandos narys netyčia ištrynė visas šakas), visus žingsnius teko pakartoti, todėl grafikas, kuriame pavaizduotos šakos ir jų ryšiai yra labai didelis.



O taip atrodė mūsų projekto saugyklos puslapis:



GitLab yra puikus bendradarbiavimo įrankis, leidžiantis dalintis savo kodu su kitais, vadovauti projektams ir redaguoti dokumentus. Ši sistema leido lengvai ir efektyviai dirbti su keliais uždaviniais vienu metu. Naudojantis šiuo įrankiu - lengva atlikti kuriamos sistemos versijavimą.

# Išvados

Po išsamaus tyrimo ir analizės galime padaryti išvadą, kad Redmine yra geriausia projektų valdymo sistema kainos ir kokybės santykiu, nes ji yra patogi, nemokama ir atviro kodo programa. GitLab yra geriausia versijavimo sistema, nes ji yra debesyje padaryta, saugi ir atviro kodo sistema, leidžianti pilnai išmokti jos veikimo principų. Dėl testavimo įrankių, tai nors Ghost Inspector yra patogesnis, bet mokamas įrankis, todėl tinkamesnis didesnėms įmonėms. Mažoms įmonėms ar laisvai samdomiems darbuotojams geriau naudoti nemokamą, bet sunkiau įvaldomą Selenium įrankį. Palyginus abi integruoto programavimo aplinkas galima teigti, kad „Visual Studio“ yra tinkamesnis pasirinkimas mūsų projektui dėl patogios grafinės sąsajos, automatinio teksto užbaigimo, taip pat tai yra programa, kuri labiau pritaikyta komandiniam darbui dėl kodo versijavimo. Nors „Xcode“ turi naudingą “Assistant Editor” funkciją bei, kaip ir „Visual Studio“, turi patogų grafinę sąsają, „Xcode“ yra pritaikytas tik “Apple”operacinėms sistemoms, todėl ir pasirinkome naudoti „Visual Studio“ integruotą programavimo aplinką.

# Naudoti dokumentai ir literatūra

Allain, A. (2022 m. april 24 d.). *5 Awesome Visual Studio Debugger Features*. Nuskaityta iš cprogramming: https://www.cprogramming.com/tutorial/visual\_studio\_tips.html

Gillbert, N. (2022 m. april 24 d.). *Compare Redmine vs ProjectManager*. Nuskaityta iš financesonline: https://comparisons.financesonline.com/redmine-vs-projectmanager

Karla, E. (2022 m. april 24 d.). *What are the advantages of Visual Studio IDE?* Nuskaityta iš minstick: https://www.mindstick.com/forum/156477/what-are-the-advantages-of-visual-studio-ide

Microsoft. (2022 m. april 24 d.). *Assisted IntelliSense – C#*. Nuskaityta iš visualstudio.microsoft: https://visualstudio.microsoft.com/services/intellicode/

Microsoft. (2022 m. april 24 d.). *First look at the Visual Studio Debugger*. Nuskaityta iš microsoft: https://docs.microsoft.com/en-us/visualstudio/debugger/debugger-feature-tour?view=vs-2022

Pedamkar, P. (2022 m. april 24 d.). *What is Visual Studio Code?* Nuskaityta iš educba: https://www.educba.com/what-is-visual-studio-code/

suggest, s. (2022 m. april 24 d.). *Compare Selenium Automation Tools vs Ghost Inspector*. Nuskaityta iš softwere suggest: https://www.softwaresuggest.com/compare/selenium-vs-ghost-inspector#:~:text=The%20quality%20and%20timeliness%20of,while%20Ghost%20Inspector%20provides%20email.

T, G. (2022 m. april 24 d.). *Introducing Collection of New Visual Studio Themes!* Nuskaityta iš microsoft: https://devblogs.microsoft.com/visualstudio/custom-themes/

TrustRadius. (2022 m. april 24 d.). *Visual Studio IDE*. Nuskaityta iš TrustRadius: https://www.trustradius.com/products/visual-studio-ide/reviews?qs=pros-and-cons#overview

KnowledgeHut. (2019 m. 11 11 d.). *Xcode vs Swift*. Nuskaityta iš KnowledgeHut: https://www.knowledgehut.com/blog/programming/swift-vs-xcode