

Informatikos fakultetas

**T120B162 Programų sistemų testavimas**

**4 laboratorinio darbo ataskaita**

|  |  |  |
| --- | --- | --- |
|  | Studentas: | Monika Kaltenytė, IFF-5/3 |
| Dėstytojas: | doc. Šarūnas Packevičius |

Kaunas 2018

# Įvadas

4 laboratorinio darbo tikslas – panaudoti projektų valdymo įrankį savo programinės įrangos kūrimo bei testavimo procese.

Darbo uždaviniai:

1. Pasirinkti ir išmokti naudotis projekto valdymo įrankiu.
2. Teisingai panaudoti įrankį kuriant bent testuojant sistemą.
3. Pateikti įrankio naudojimo rezultatus ataskaitoje

# Projektų valdymo įrankiai

Skyriuje pateikiami projektų valdymo įrankių pavyzdžiai, bei vienas iš jų pasirenkamas.

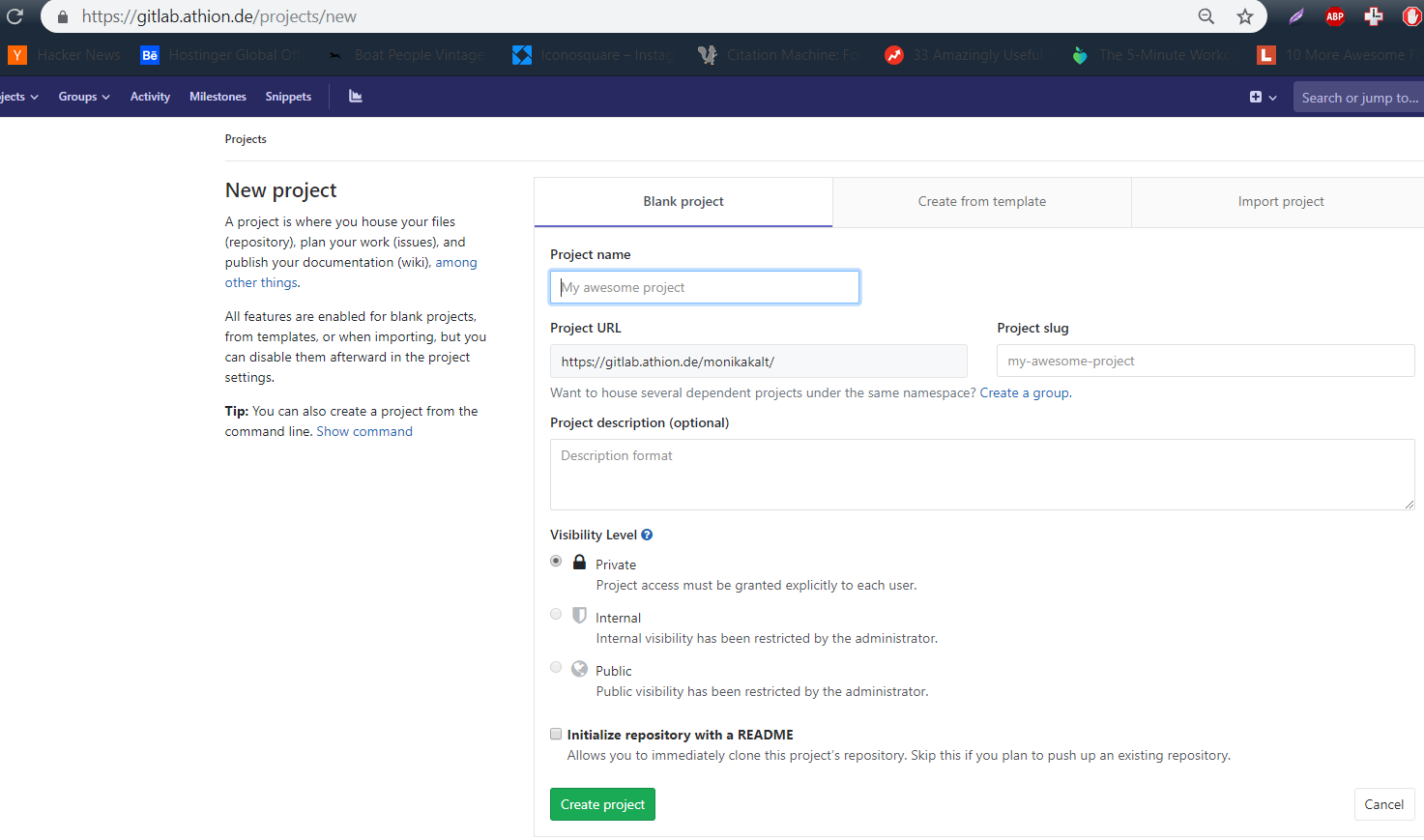
Populiarūs projektų valdymo įrankiai:

1. BitBucket
2. GitLab
3. Redmine
4. GitHub

Projekto valdymui pasirinktas įrankis GitLab. Tai nemokamas įrankis, kurį lengvą išmokti naudoti, turi didelį naudotojų skaičių, bei jį galima integruoti į programavimo aplinkas.

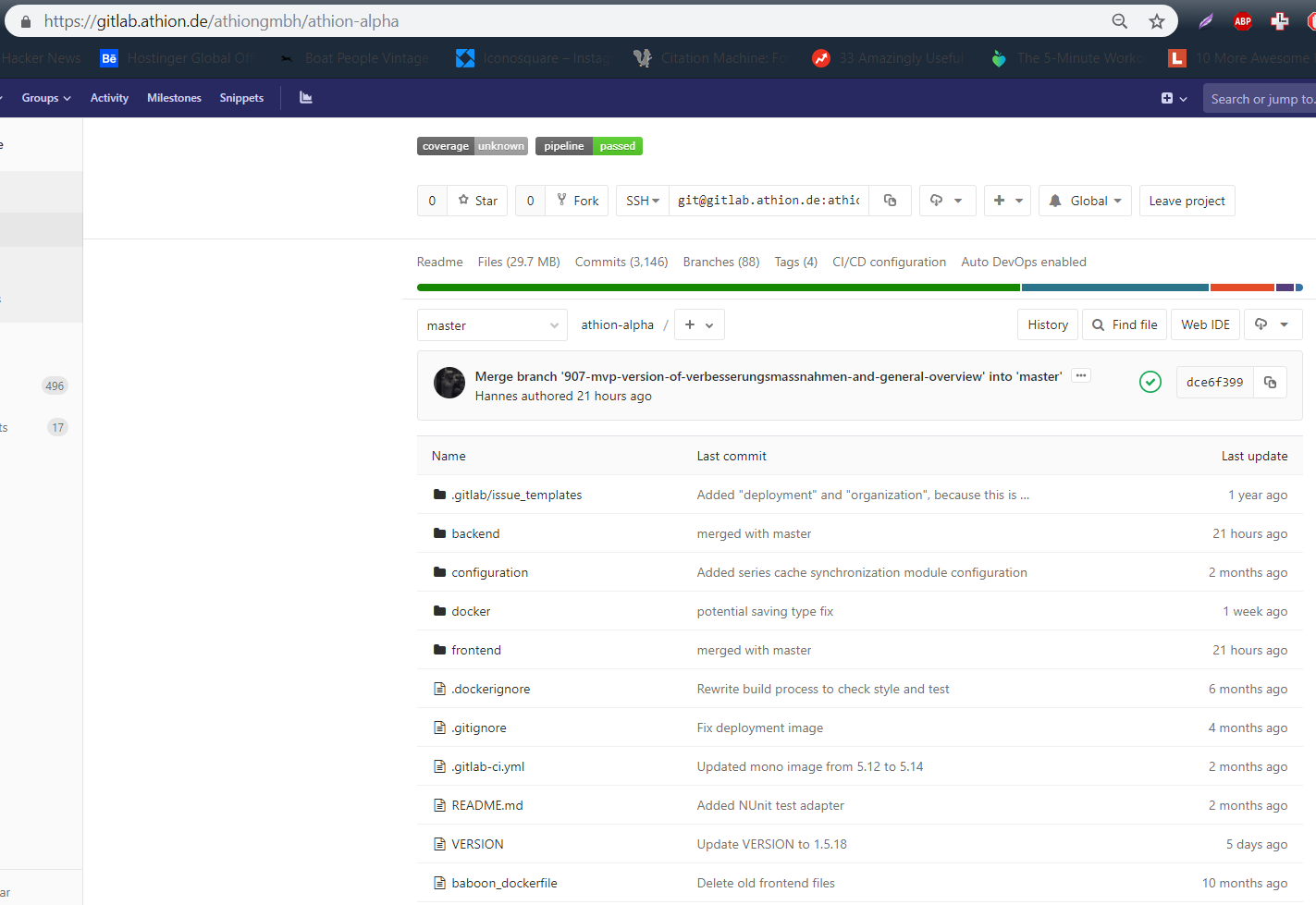
# Projekto aplinkos paruošimas GitLab įrankyje

GitLab įrankyje projektai skirstomi į projektus, taigi pirmiausia jį reikia susikurti:



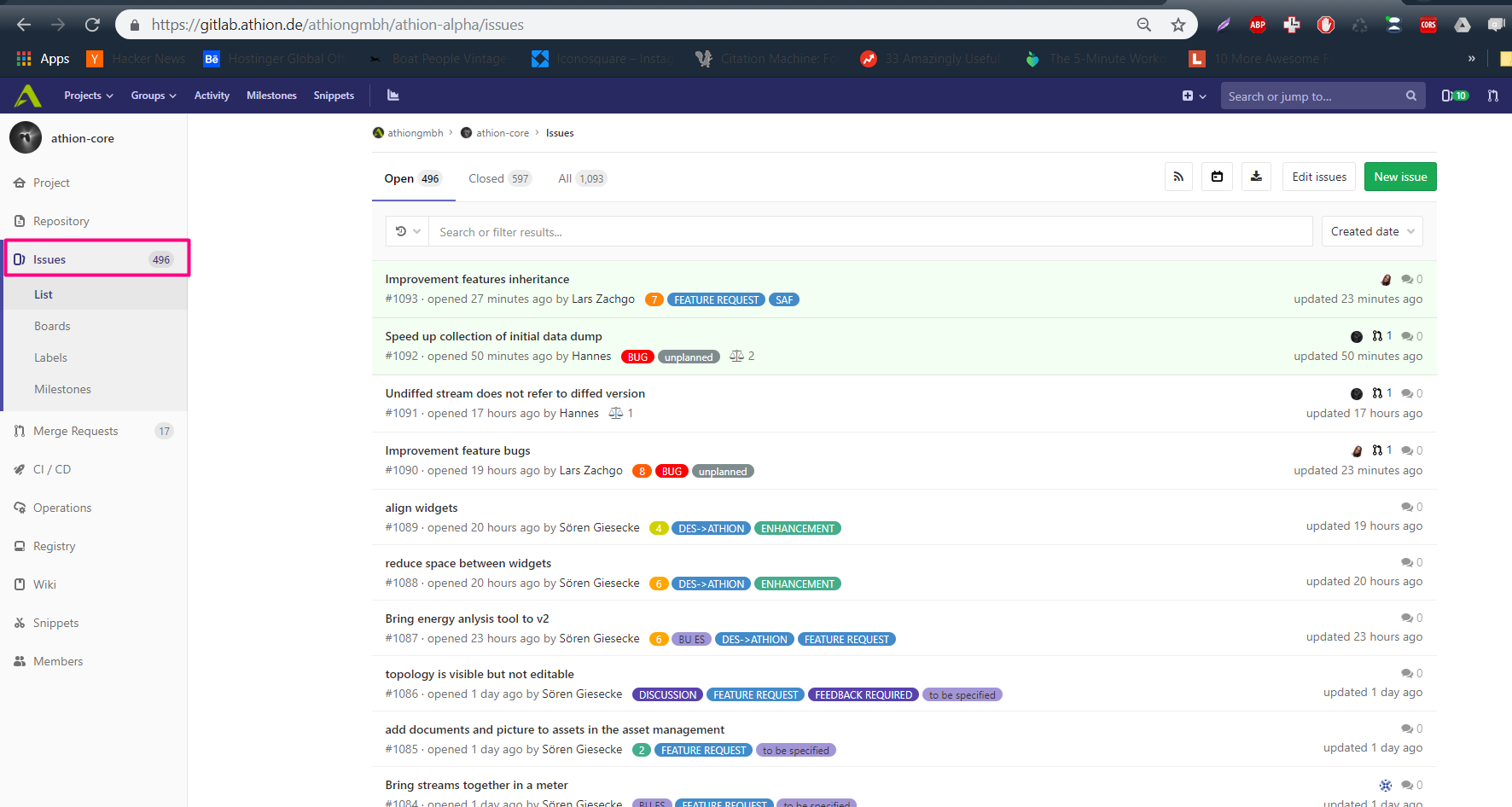
pav. Projekto sukūrimas

Sukūrus repozitoriją į ją patalpinami projekto failai:



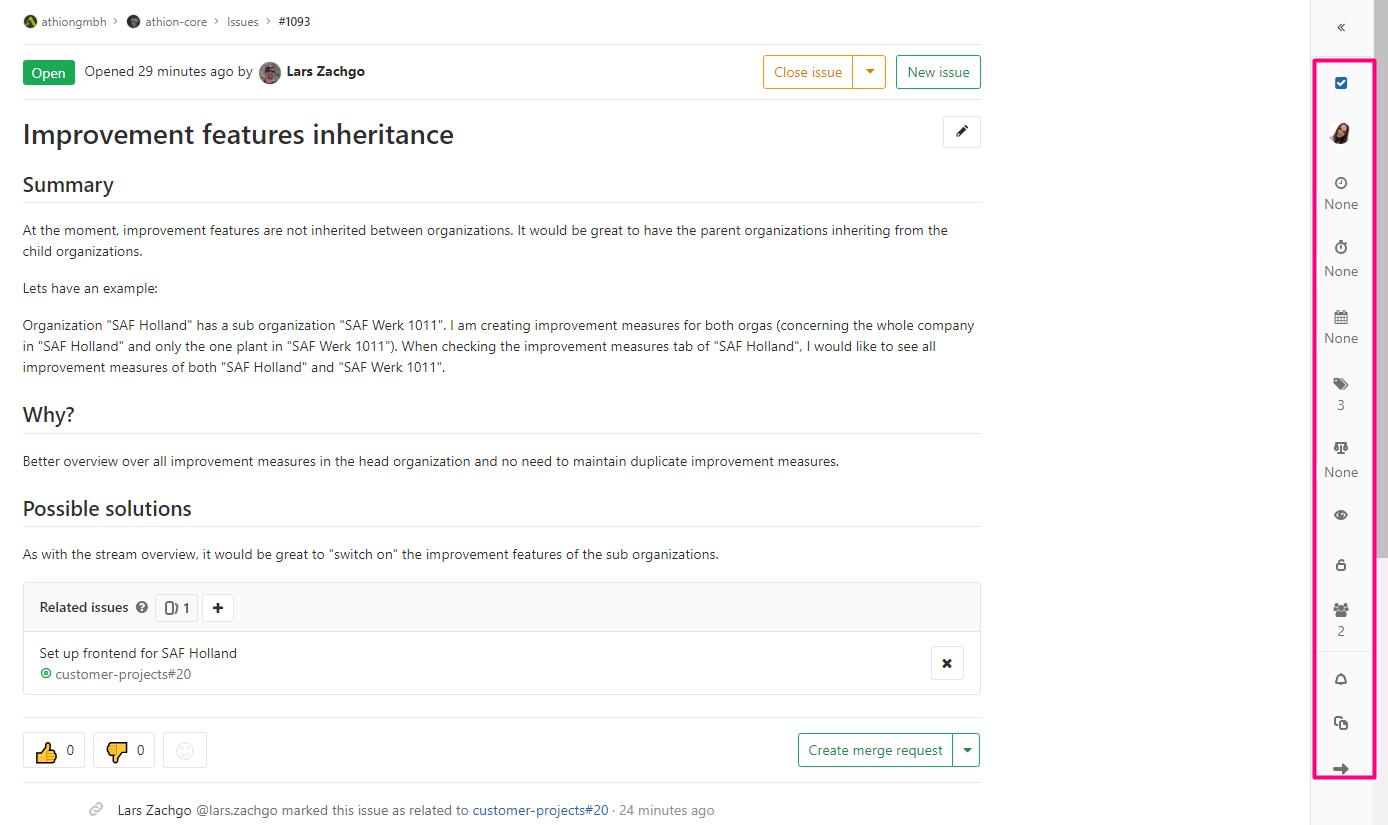
pav. Repositorijos vaizdas

Toliau norint peržiūrėti užduotis, reikia nueiti :



pav. Užduočių sekyklė

Kiekvienai užduočiai galima priskirti skirtingas kategorijas, svorius, žmones, kuriems ji skirta ir panašią informaciją:



Pav. Atskiros užduoties vaizdas

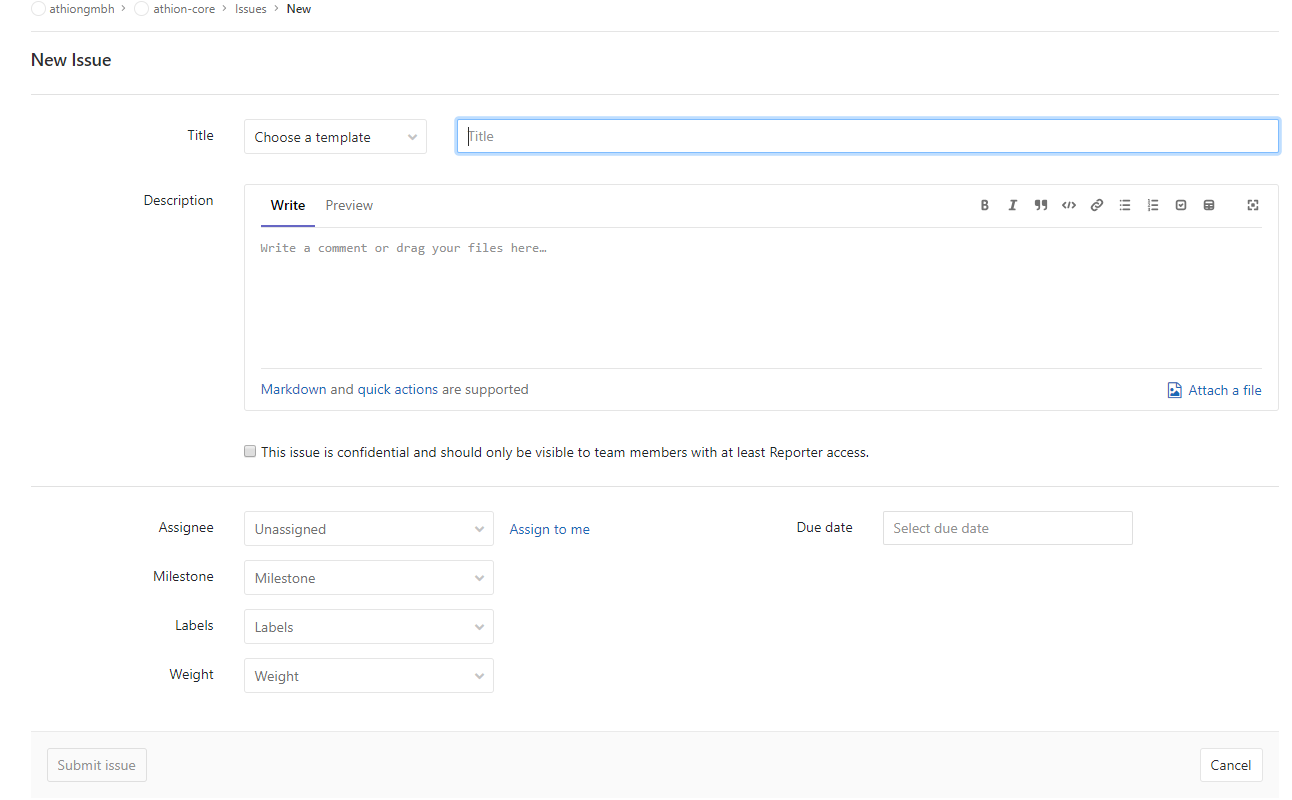
# 4. Klaidų užduočių registravimas

Skyriuje pateikiami projekto testavimo atvejų registravimo rezultatai BitBucket įrankyje.

Pirmo laboratorinio metu aprašyti testavimo atvejai užregistruojami į repozitorijos skiltį Issues:

pav. Bendras testavimo atvejų vaizdas

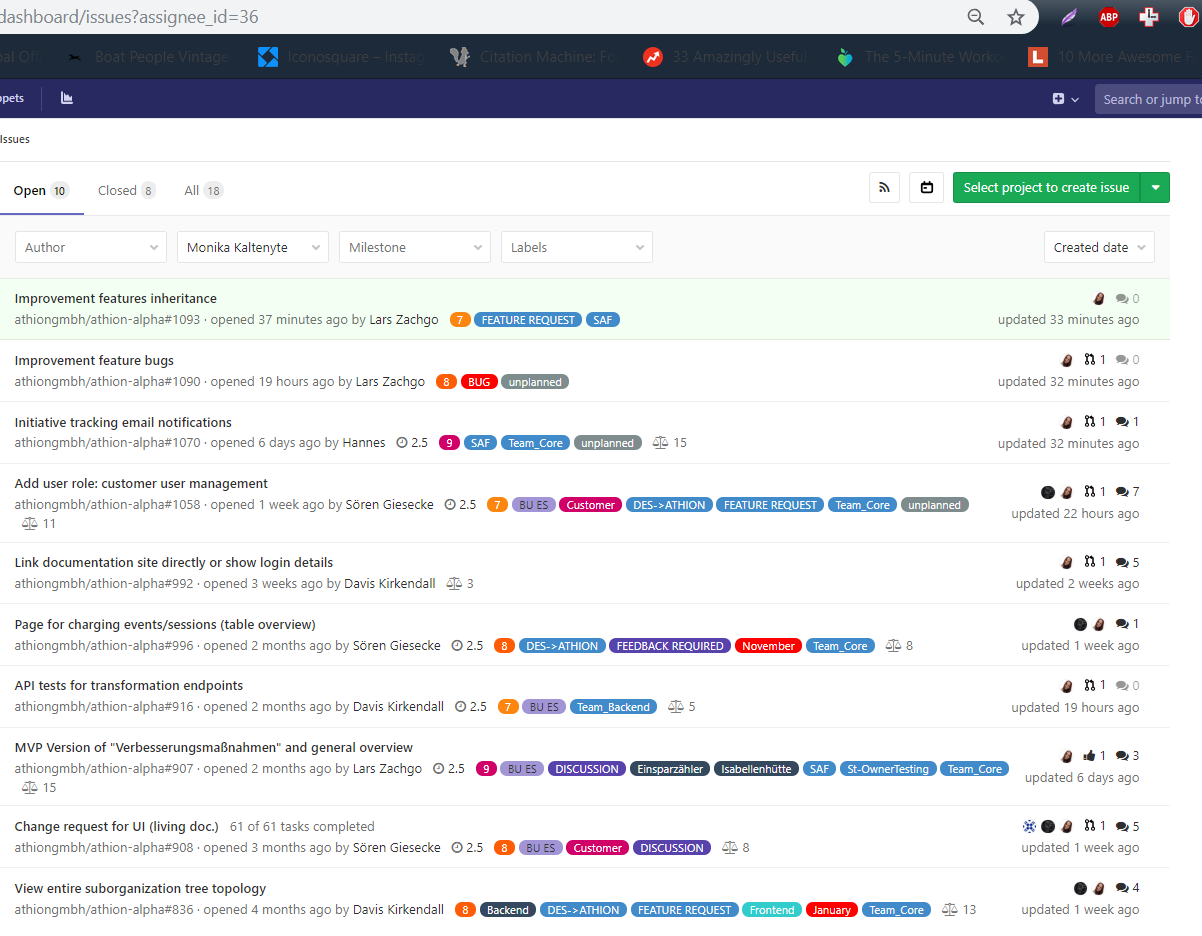
Atskiro užduoties scenarijaus kūrimas atrodo taip:



pav. Užduoties kūrimo bendras bendras vaizdas

# Testavimo užduočių atlikimas

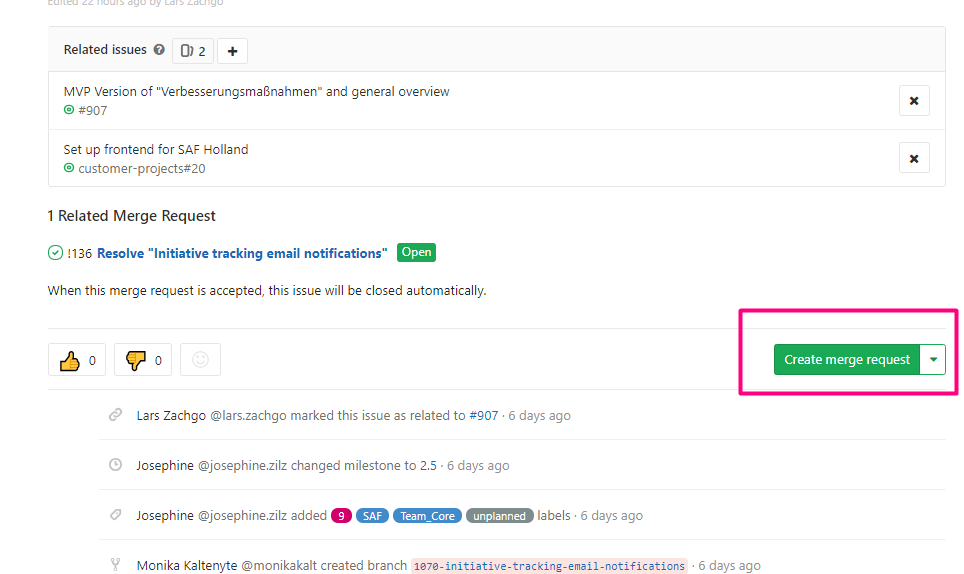
Skyriuje pateikiamas paaiškinimas, kaip turi būti išsprendžiamos užregistruotos užduotys GitLab sistemoje.



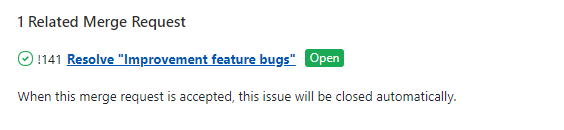
pav. 7 Matomos naujos užduotys

Jas galima rūšiuoti pagal įvairius atributus.

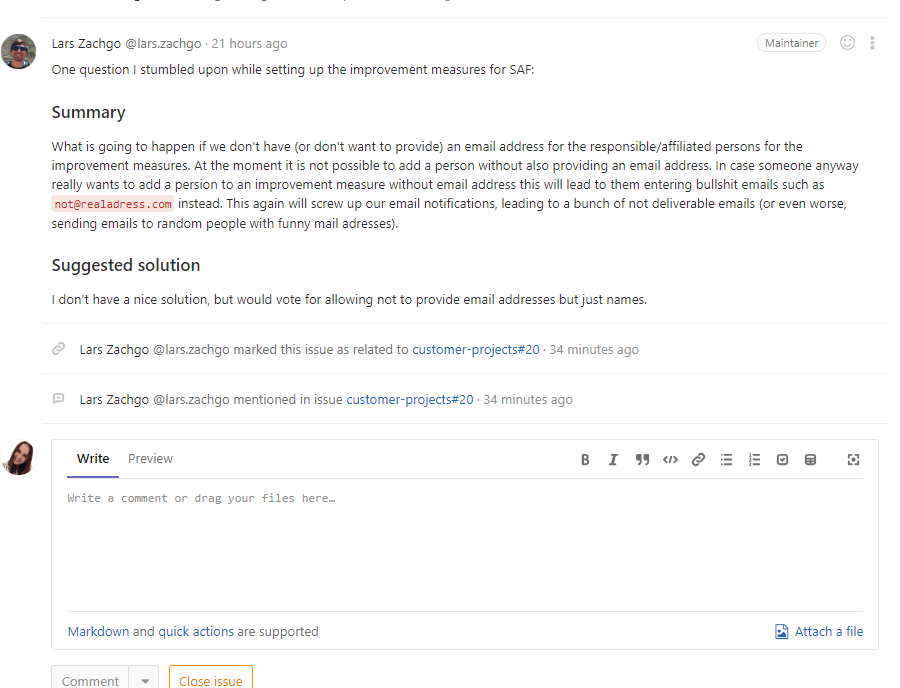
1. Norint pradėti vykdyti užduotis reikia sukurti Merge request:



Prie užduoties matoma, kad buvo sukurtas Merge request:

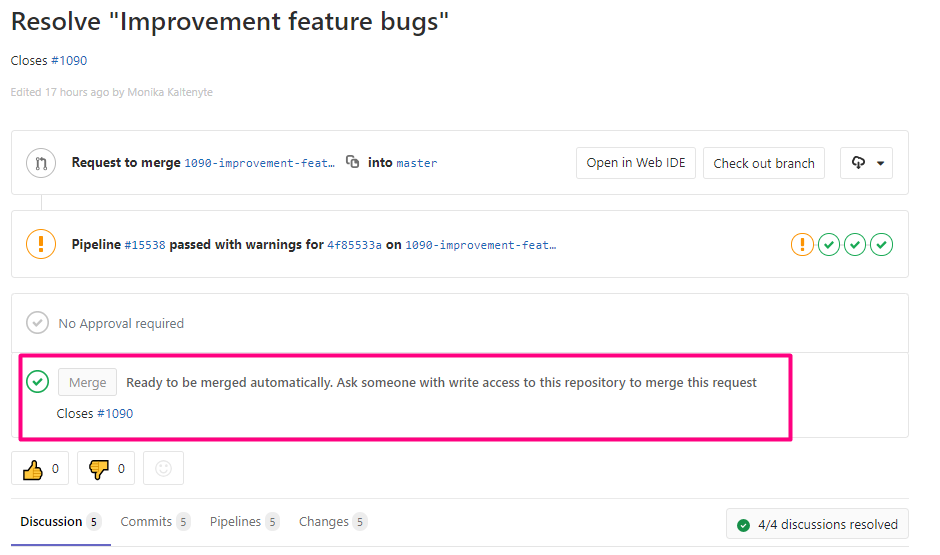


1. Testuotojui iškilus klausimų, jis gali prie užduoties palikti komentarą / klausimą:



pav. Pakomentuota užduotis

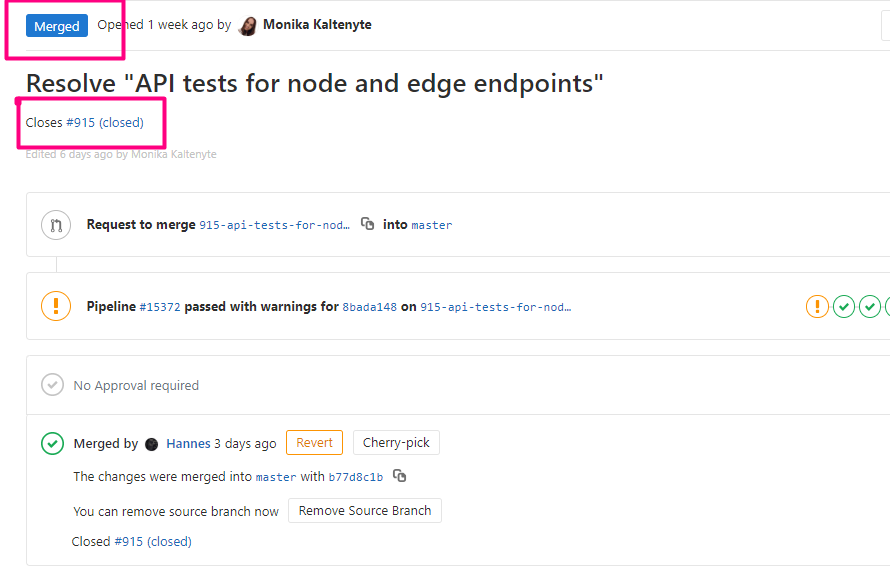
1. Kai užduotis atlikta, reikia panaikinti WIP (Work In Progress) statusą ir perskirti užduotį, kad kažkas ją peržiūrėtų:



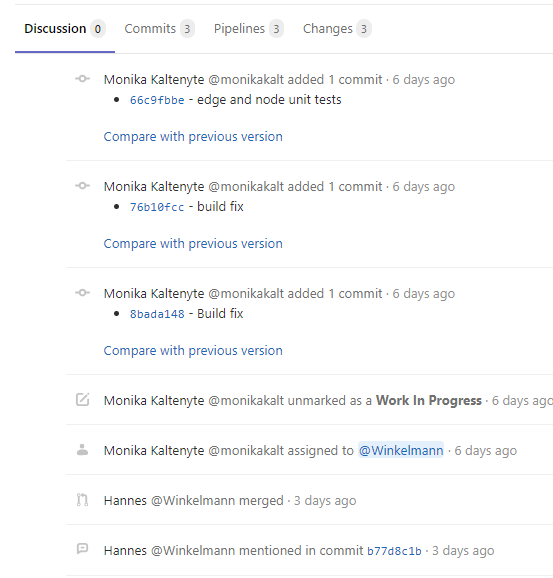
Matoma, kad užduoties pakeitimai gali būti sukelti į gamybinę versiją.

1. Užduotis yra uždaroma:

Kai užduotis baigta, ją galima uždaryti. Tokiu atveju galima panaikinti prieš tai sukurtą šaką.



GitLab sistemoje atsidarius užduotį galima matyti visą užduoties gyvavimo istorija, įskaitant statuso pasikeitimus, bei komentarus:



pav. Pilna užduoties istorija

# Testavimo užduočių registravimas

GitLab sistemo patogu registruoti sistemoje pastebėtas klaidas. Tai atliekama tokiu pačiu principu, kaip klaidų užduočių registravimas.

1. Testuotojas sukuria naują užduotį (issue) ir uždeda kategoriją ,,test‘‘
2. Jei reikia, užduočiai priskiriamos kitos kategorijos.
3. Įvykdžius užduotį, panaikinamas Work In Progress statusas ir užduotis uždaroma.

# Išvados

* 1. Programinės įrangos kūrimo proceso valdymui buvo surastas šiai užduočiai atlikti tinkamas įrankis GitLab;
  2. GitLab buvo naudojamas programinės įrangos kūrimo bei testavimo procesuose, taip procesus padarant lengviau valdomais, detalesniais bei aiškesniais. Taip pat klaidų registravimui.
  3. Ataskaitoje pateikti naudojimo įrankiu rezultatai, bei komentarai.