

Uvod u softversko inženjerstvo

Verzioniranje
koda uz Git

Nikola Luburić
nikola.luburic@uns.ac.rs



Verzioniranje
koda uz Git

Osnovne komande

Repozitorijumi i grane

GitHub: PRs, Akcije

Proces rada na projektu

Definiši organizaciju posla (na nivou dana) za scenario gde 4 meseca praviš aplikaciju u tročlanom timu, tako da:

- ❖ Za izradu aplikacije treba 12 inženjer meseci
- ❖ Na 3 nedelje možete da prikažete sve nove funkcije cele aplikacije
- ❖ Ne koristite Git, SVN, Mercurial...
- ❖ Gde skladištite kod? Kako integrišete promene? Šta kada se pojavi bug? Šta kada menjate isti fajl?

Za prethodno definisan proces rada, šta je potrebno da se reše sledeći problemi:

- ❖ 2 nedelje pred kraj se otkriva složeni *bug* u delu aplikacije koji *ne bi trebalo* da je menjan 2 meseca
- ❖ Asistentu treba dokaz da svaki član tima doprinosi
- ❖ Inženjer mora dnevno integrisati sve promene
- ❖ 3 inženjera menjaju isti fajl da reše svoje zadatke
- ❖ Inženjer ima 2 ideje za rešavanje problema i želi obe da implementira pre nego što se odluči
- ❖ Razvoj aplikacije se širi na 2 godine i 50 ljudi

- ❖ 2 nedelje pred kraj se otkriva složeni *bug* u delu aplikacije koji *ne bi trebalo* da je menjan 2 meseca
- ❖ Asistentu treba dokaz da svaki član tima doprinosi
- ❖ Inženjer mora dnevno integrisati sve promene
- ❖ 3 inženjera menjaju isti fajl da reše svoje zadatke
- ❖ Inženjer ima 2 ideje za rešavanje problema i želi obe da implementira pre nego što se odluči
- ❖ Razvoj aplikacije se širi na 2 godine i 50 ljudi

Istorija promena

Autorstvo
promene koda

Laka integracija
promena

Razrešavanje
konflikta

Grananje
razvoja

Skalabilnost

Istorija promena

Autorstvo
promene koda

Sistem za
verzioniranje
koda

Laka integracija
promena

Razrešavanje
konflikta



Granje
razvoja

Skalabilnost

`git init
za početak`

Istorija promena

Radno stablo
(*work tree*)



Rectangle.cs

Pripremna zona
(*index*)

`git add`

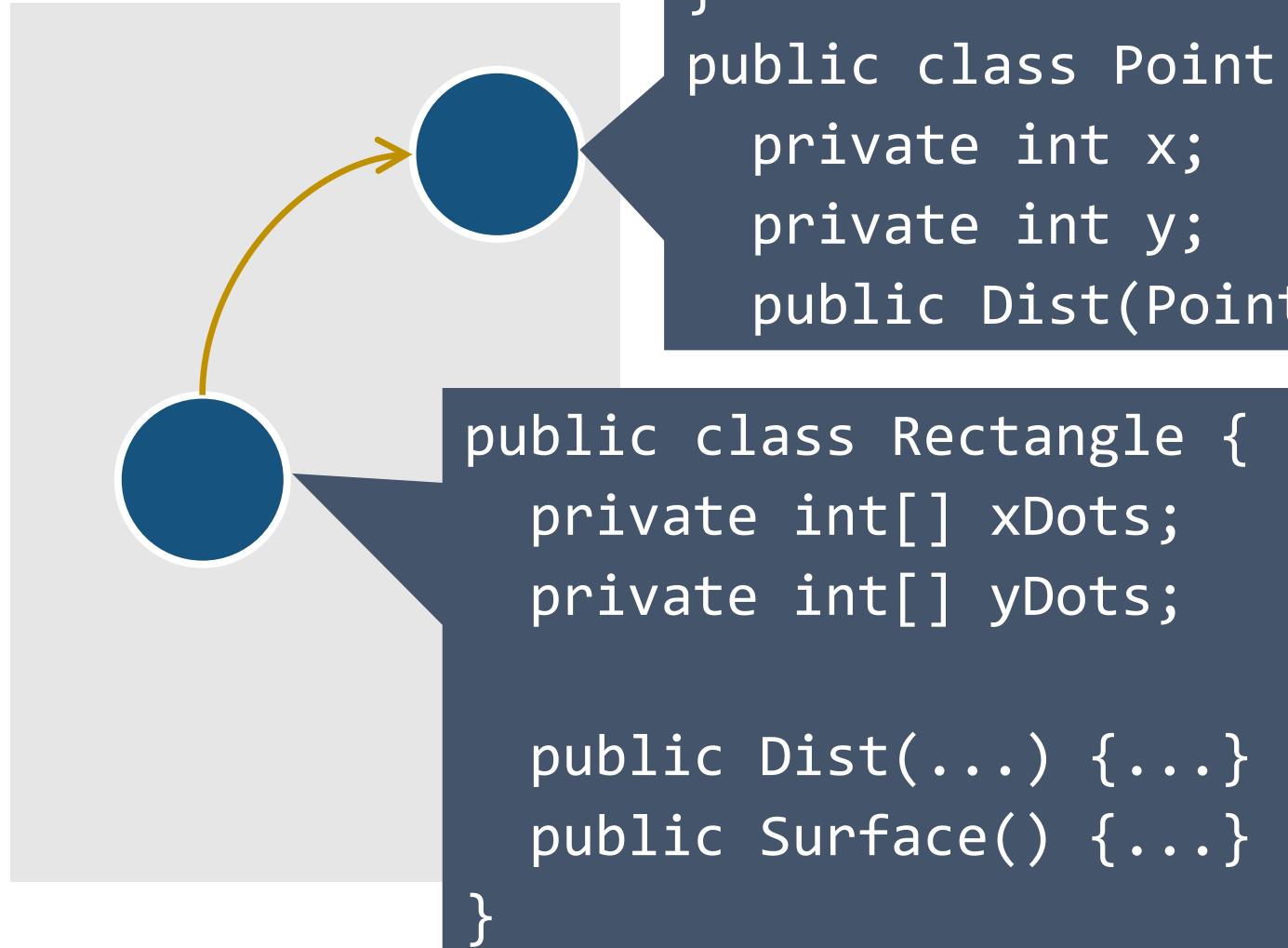
Skup
planiranih
izmena za
commit

Lokalni *git*
repozitorijum

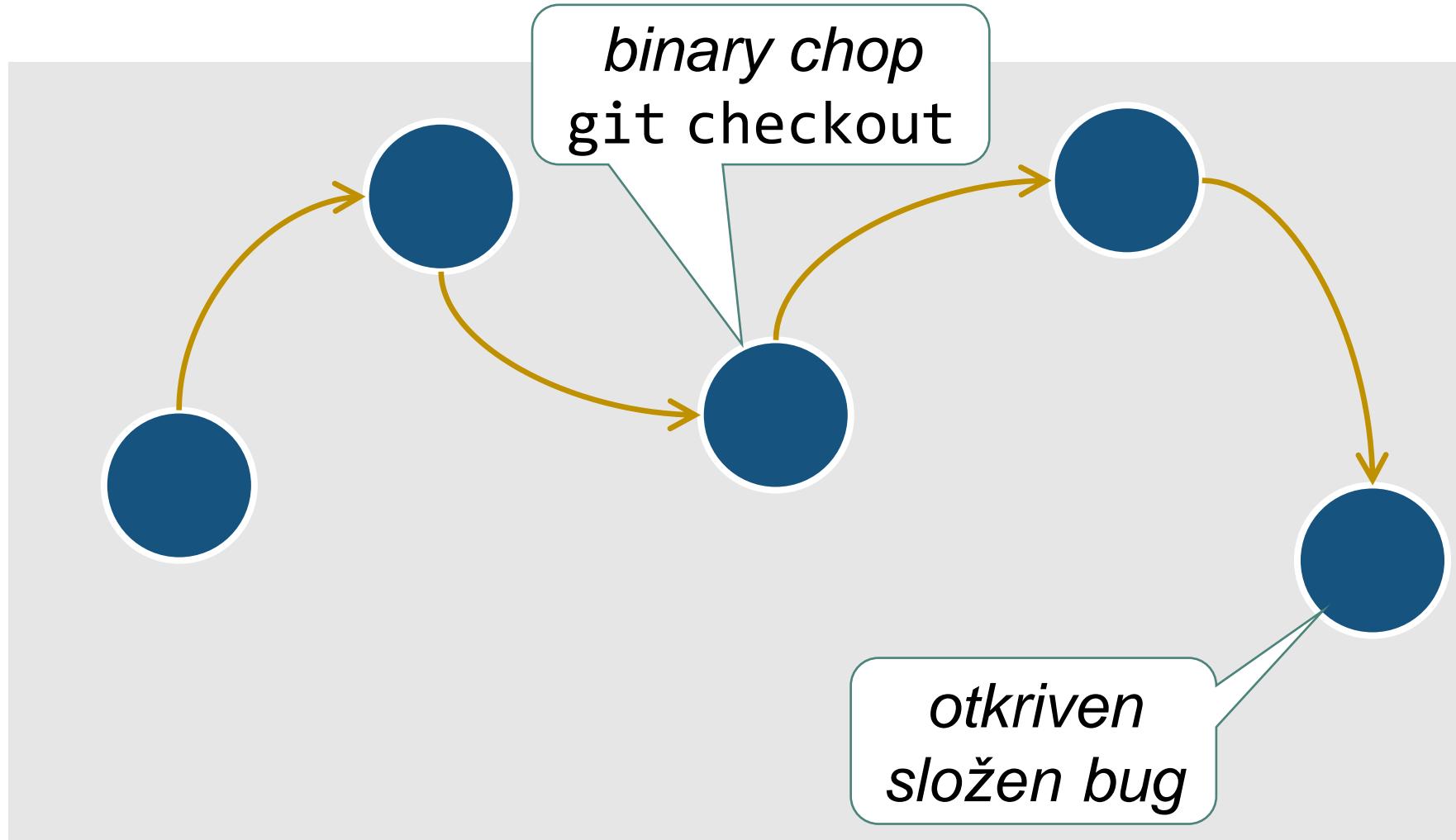
**Trajna
evidencija
izmena**

`git commit`

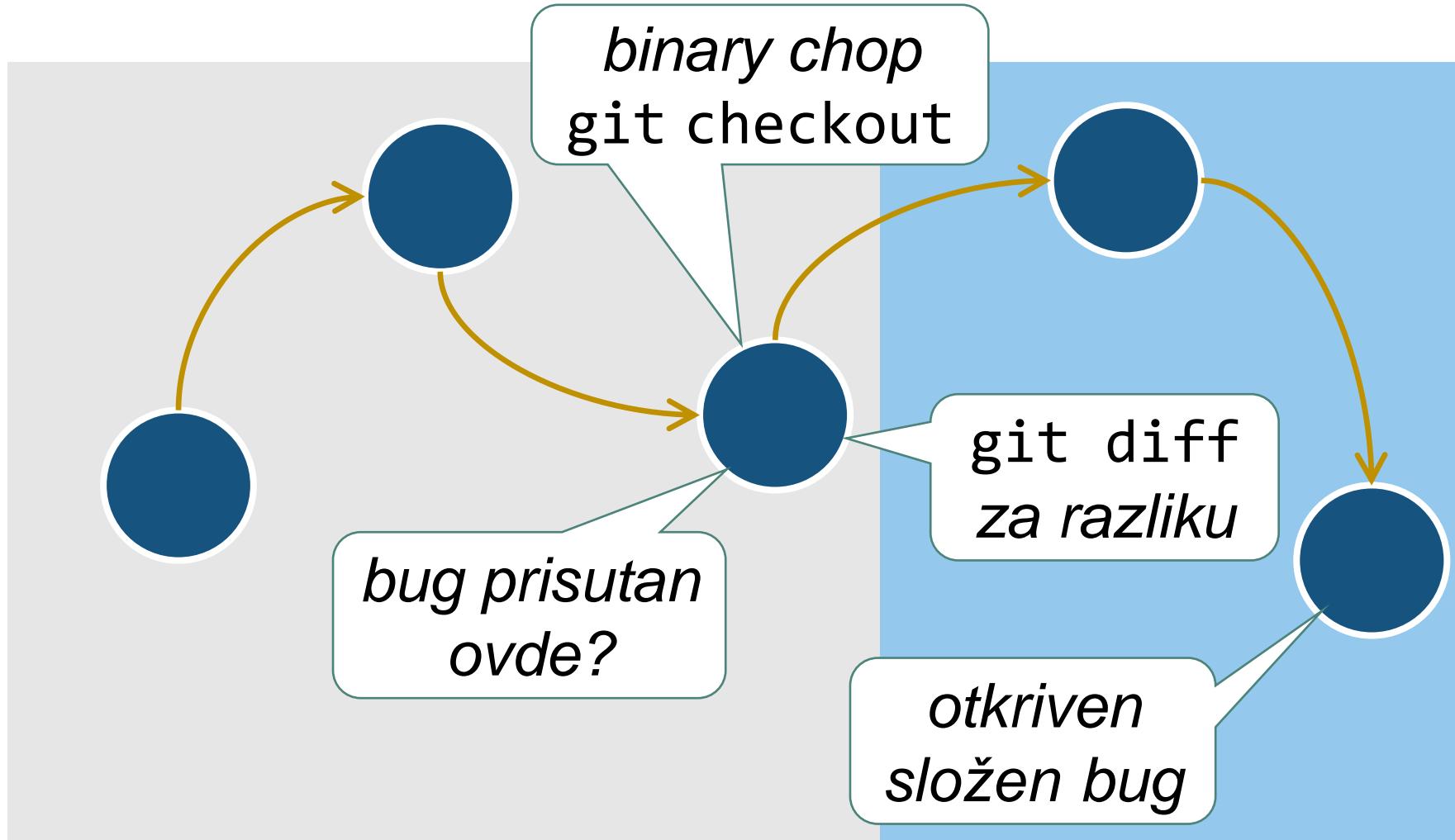
Istorija promena



Istorija promena

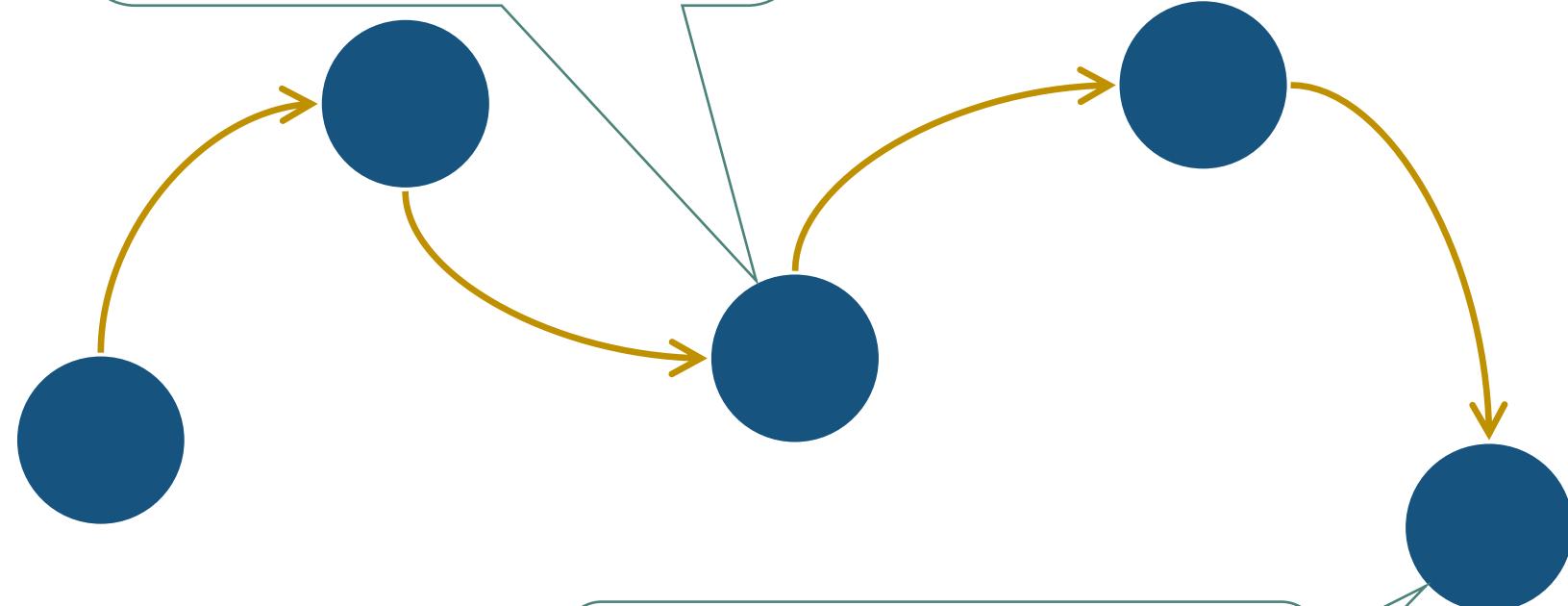


Istorija promena



Autorstvo
promene koda

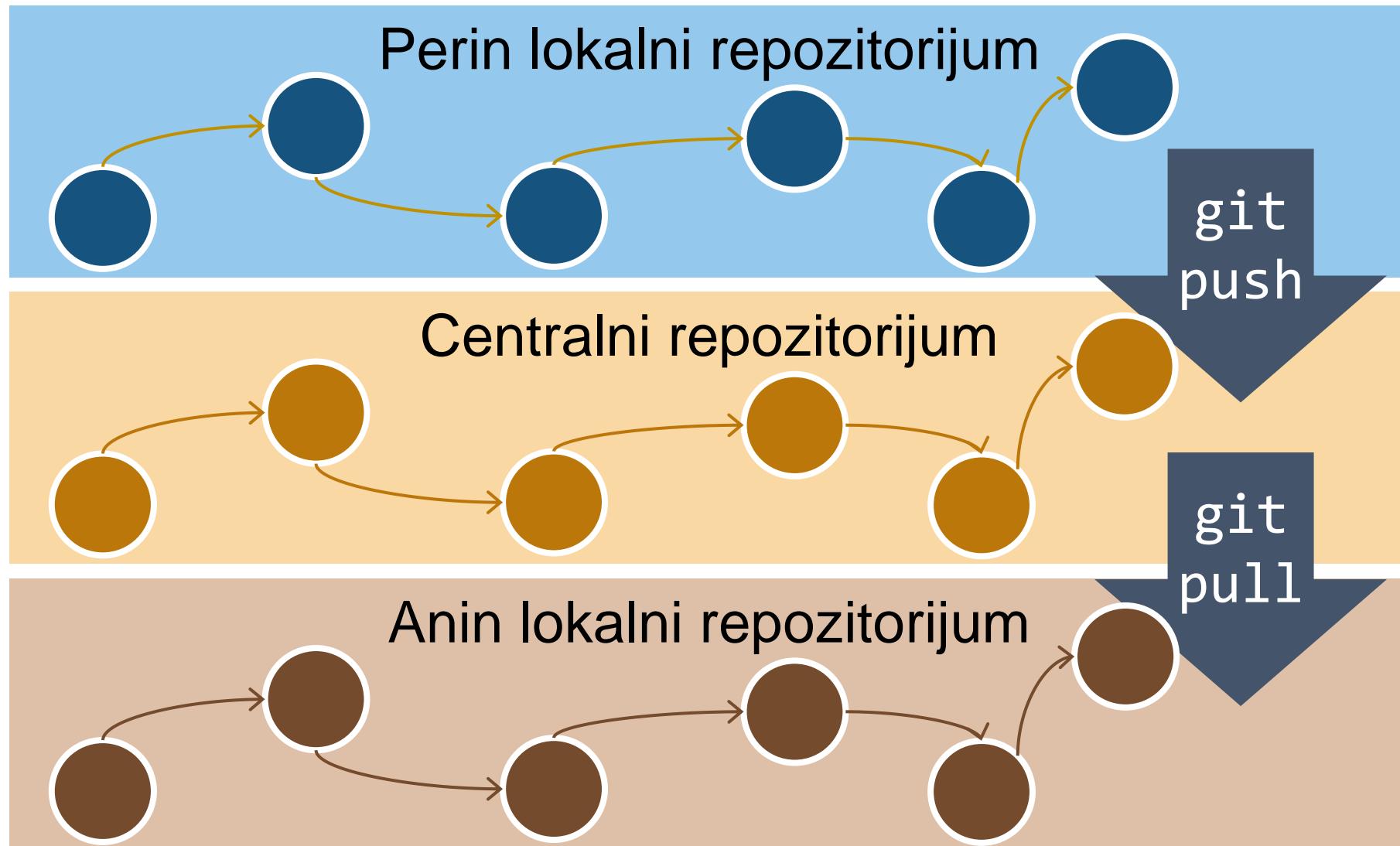
git log
commit 3fd...3sf
Author: Pera Perić
Date: 15:19 29.1.20.



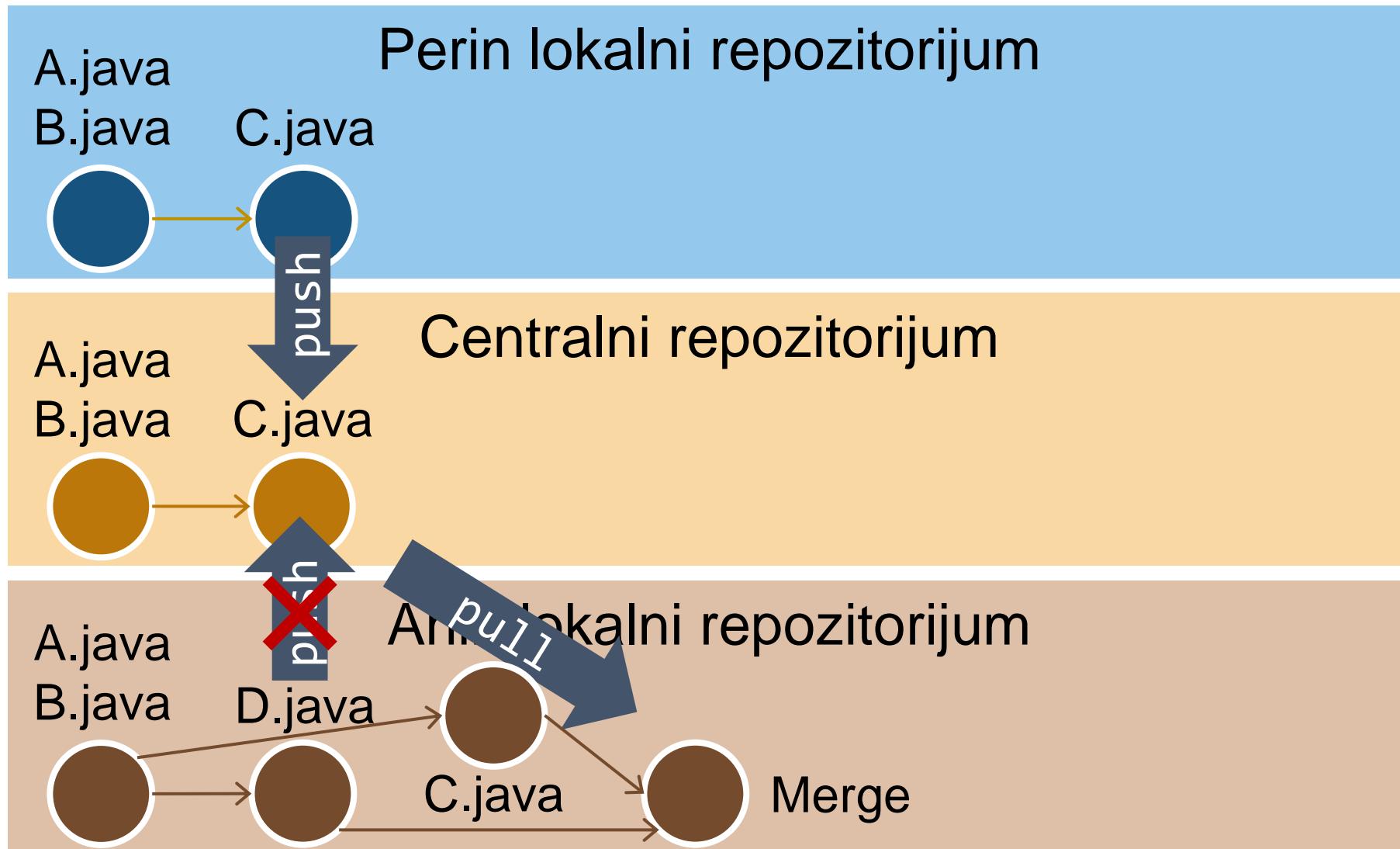
gitk
vizualizacija
istorije promena

git blame <file>
ko i kad je menjao šta

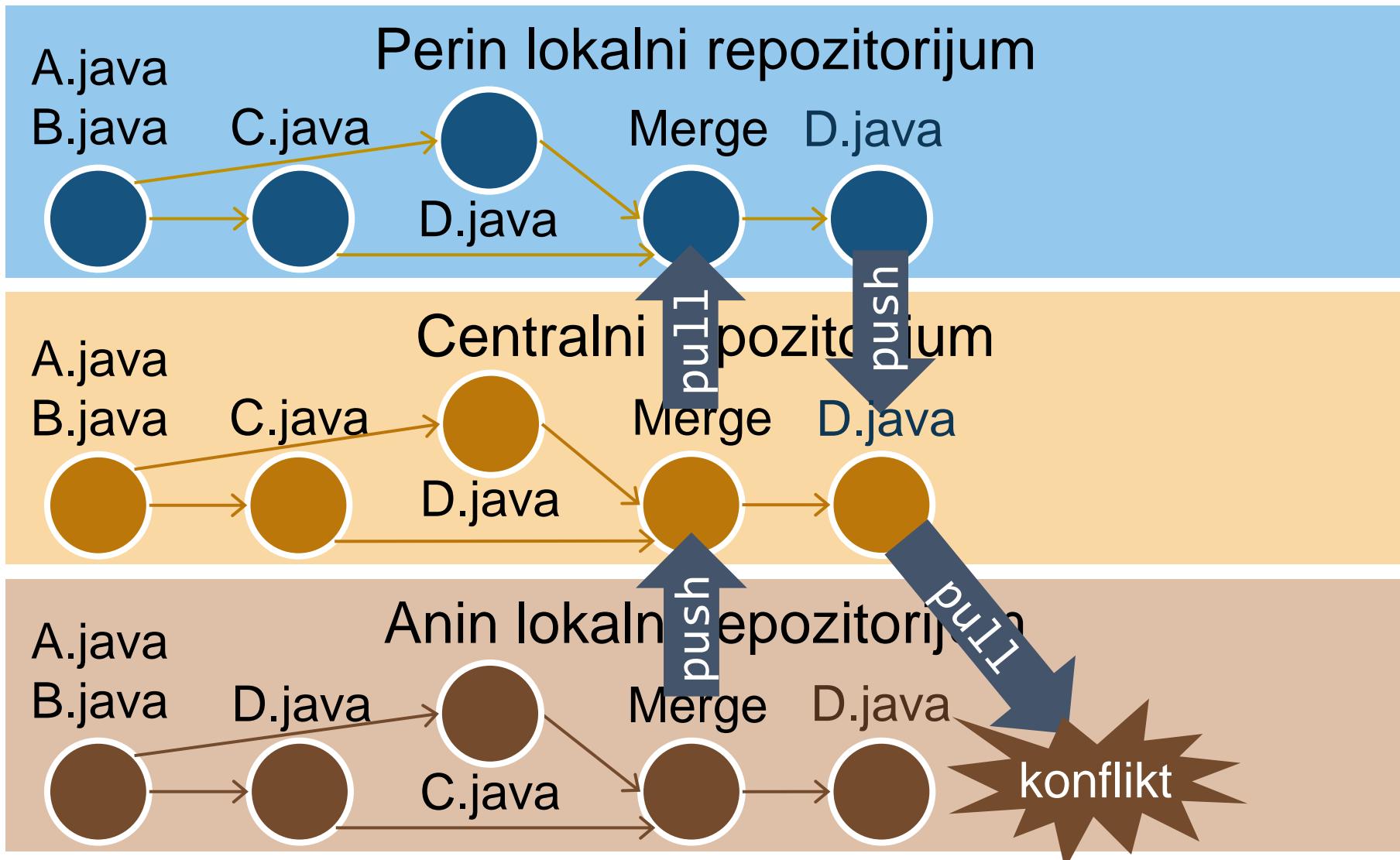
Laka integracija promena



Razrešavanje konflikta



Razrešavanje konflikta



Razrešavanje konflikta

„naše“ linije

fajl gde je konflikt

```
<<< <<< HEAD:D.txt
```

```
Hello world
```

```
=====
```

```
Goodbye
```

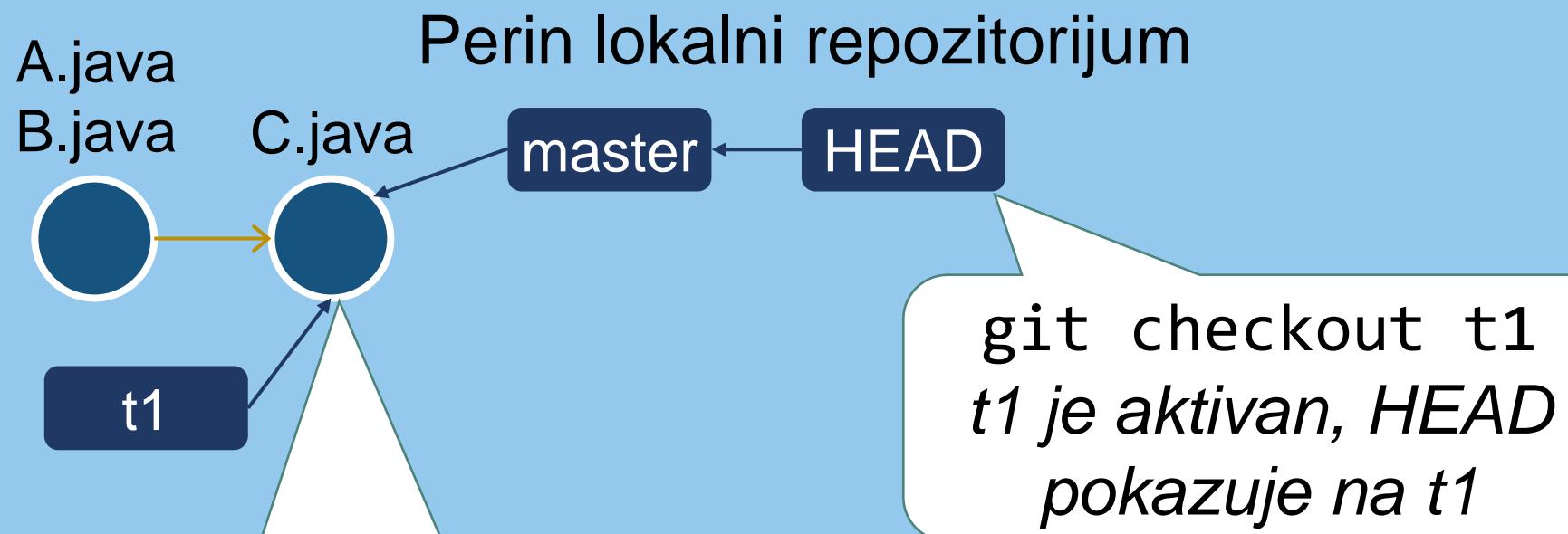
```
>>> >>> 7d3a...8s:D.txt
```

„tuđe“ linije

brši << == >>, odluči konačan kod i commit

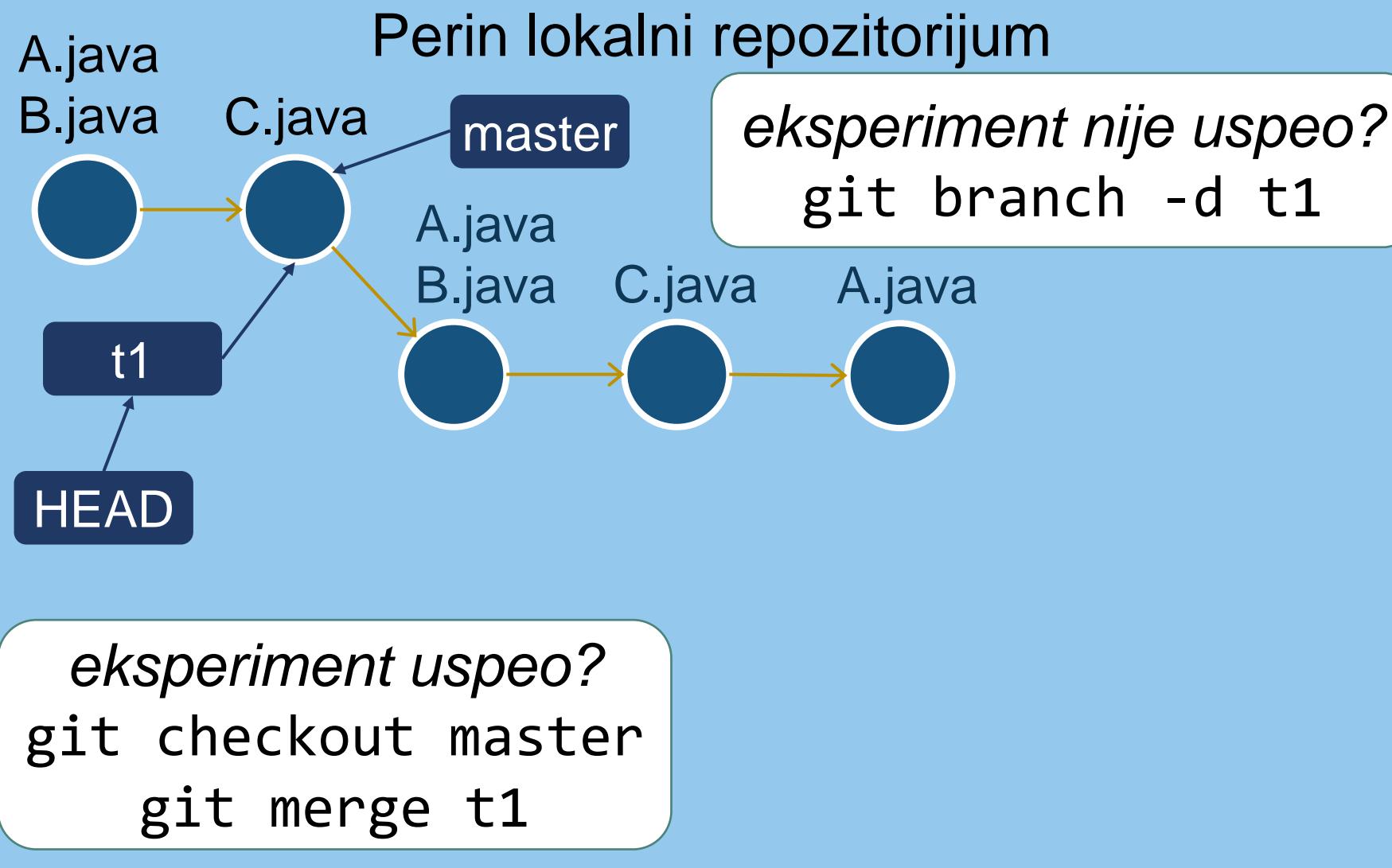
konflikt

Granje razvoja

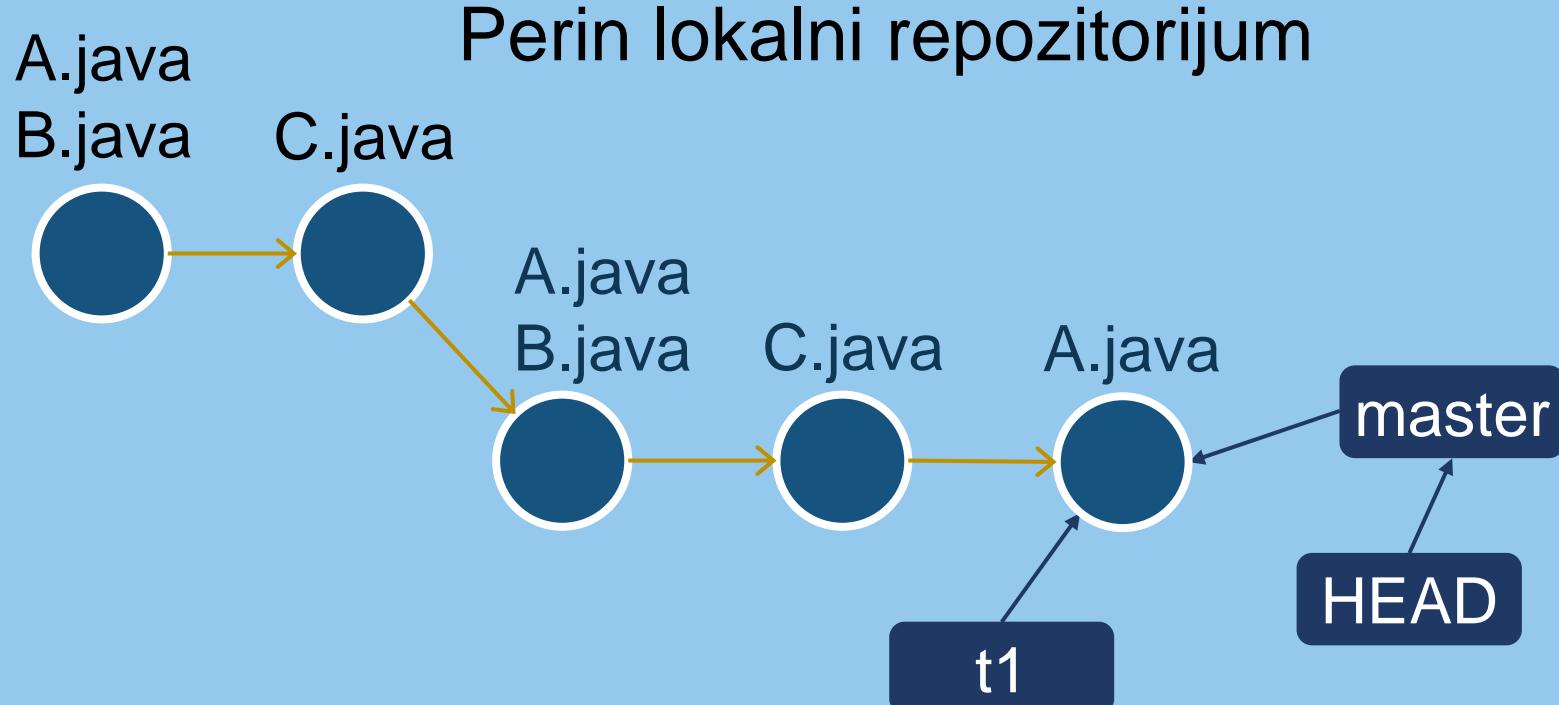


git branch t1
nova grana t1 počinje od HEAD commit-a

Granje razvoja



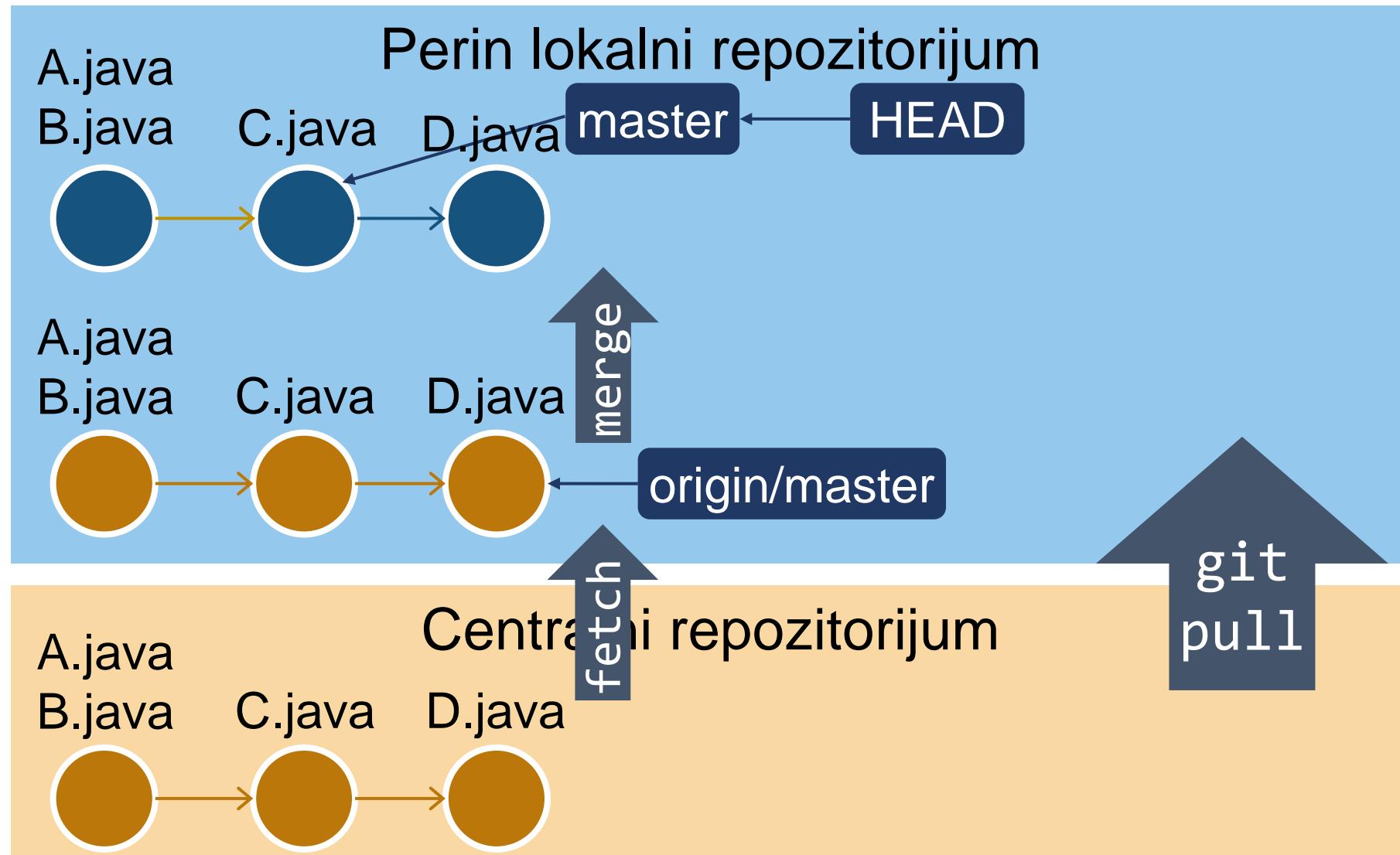
Grananje razvoja



eksperiment uspeo?
git checkout master
git merge t1

*šta da je bilo
commit-ova
na master-u?*

Grananje razvoja



Granje
razvoja

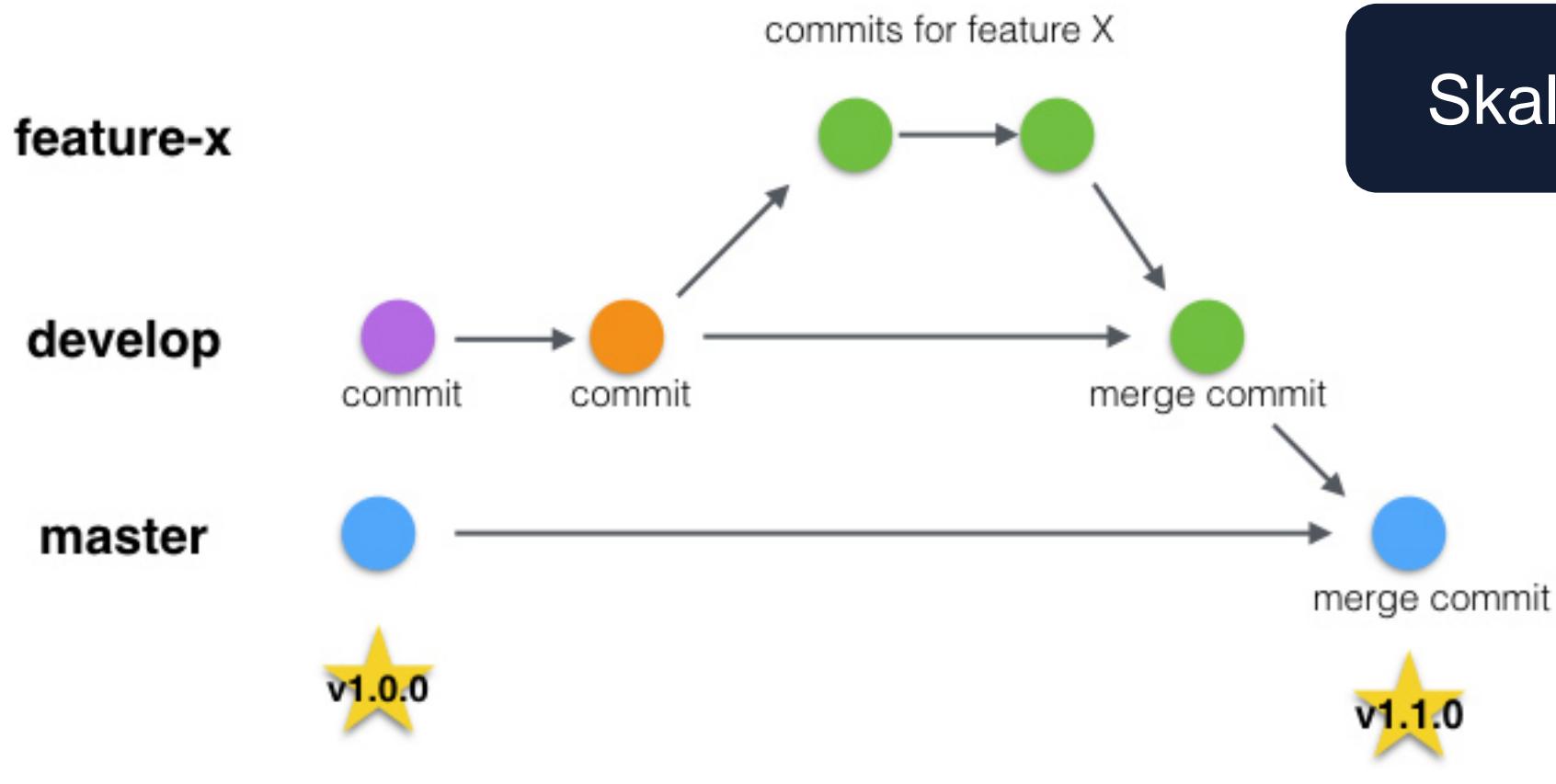
Istorija promena

Autorstvo
promene koda

.git

- **hooks/** – Skripte koje se izvršavaju u različitim delovima Git workflowa.
- **info/exclude** – Lokalni globalni .gitignore
- **objects/** – Sadržaj izmena (datoteke, direktorijumi), metapodaci commita
- **refs/** – Pokazivači na commitove (lokalne i remote grane, tagovi)
- **HEAD** – Pokazivač na ref ili commit koji ističe na šta HEAD pokazuje.
- **config** – Konfiguracija udaljenog repozitorijuma, grana...
- **description** - Opis repozitorijuma, uglavnom se ne koristi.

Skalabilnost



šta je
prikazano?

kako ide tok
razvoja?

kako
skaliramo?

[Git Workflow](#): A recipe for using Git to accomplish work in a consistent and productive manner.

Skalabilnost

The screenshot shows a pull request interface with several annotated elements:

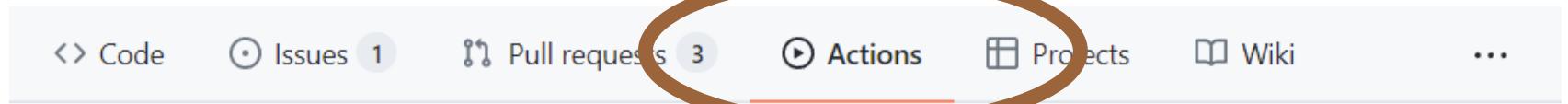
- A brown oval highlights the "Pull requests 3" button in the top navigation bar.
- A green button labeled "New pull request" is highlighted with a brown oval.
- A brown oval highlights the comment counts (5, 20, 13) next to the comments icon on the right side of the pull request list.
- A large brown box encloses the main content area, which includes:
 - A code diff view for "src/Tutor.Core/DomainModel/KnowledgeComponents/KnowledgeComponentMastery.cs".

```
src/Tutor.Core/DomainModel/KnowledgeComponents/KnowledgeComponentMastery.cs
97 +     private void When(SoughtHelp @event)
98 +     {
99 +         /*
100 +          * Possibly record how many times help was sought in AeMastery?
101 +         */
102 +     }
```
 - A comment from "milana-todorovic" 6 days ago:

I feel like keeping a count of how many times help was requested for an AE could be useful, it seems like it would be useful for AE selection in cases where multiple AEs have the same mastery.
 - A reply from "Luburic" 6 days ago:

It makes sense, but I would simplify the design for now. The plugin could be enhanced to show one hint at a time, but this is not something we can currently support (no time to write more detailed hints).

Pull request
Alat za reviziju
koda i kolaboraciju



Skalabilnost

Select workflow

Showing runs from all workflows

Filter workflow runs

98 workflow runs

Event ▾ Status ▾ Branch ▾ Actor ▾

KcMastery pass, completion, and satisfaction

Build #110: Pull request #24 synchronize by milana-todorovic

Suggested for this repository

Docker image

By GitHub Actions



Build a Docker image to deploy, run, or push to a registry.

Configure

Dockerfile ●

.NET

By GitHub Actions



Build and test a .NET or ASP.NET Core project.

Configure

C# ●

.NET Desktop

By GitHub Actions



Build, test, sign and publish a desktop application built on .NET.

Configure

C# ●

AE selection tracking, and more data added ...

Build #109: Pull request #25 synchronize by milana-todorovic

feat/ae-selection-tracking

5 days ago

5m 9s

...

GitHub Actions

Alat za automatizaciju razvoja softvera

Dobro formiran *commit*...

Skalabilnost

sadrži povezane izmene

- Lakše za razumevanje
- Lakše za *rollback*

se pravi često

- Frekventna integracija promena
- Manji gubitak posla prilikom škartiranja

je kompletan

- Logička celina koda, ne „šta sam stigao pre odlaska...“
- Revidiran, istestiran

prati dobra poruka

- Poštuje [conventional commit](#) strukturu
- Odgovara na „zašto je napravljena izmena“
- Objasnjava šta je suštinska promena u implementaciji

*Posebno bitno
u timskom radu*

Istorija promena

Autorstvo
promene koda

Sistem za
verzioniranje
koda

Laka integracija
promena

Razrešavanje
konflikta

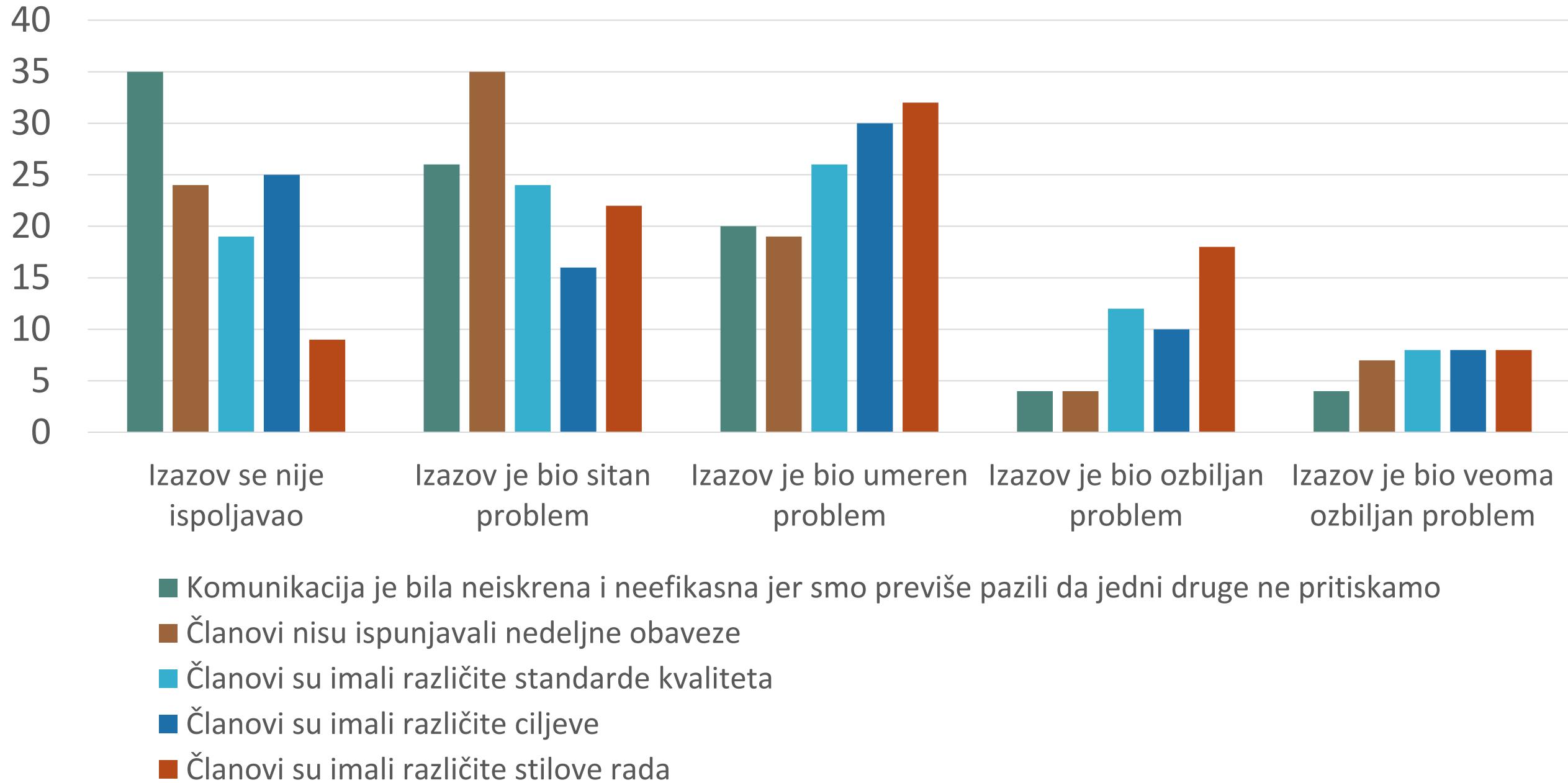


Grananje
razvoja

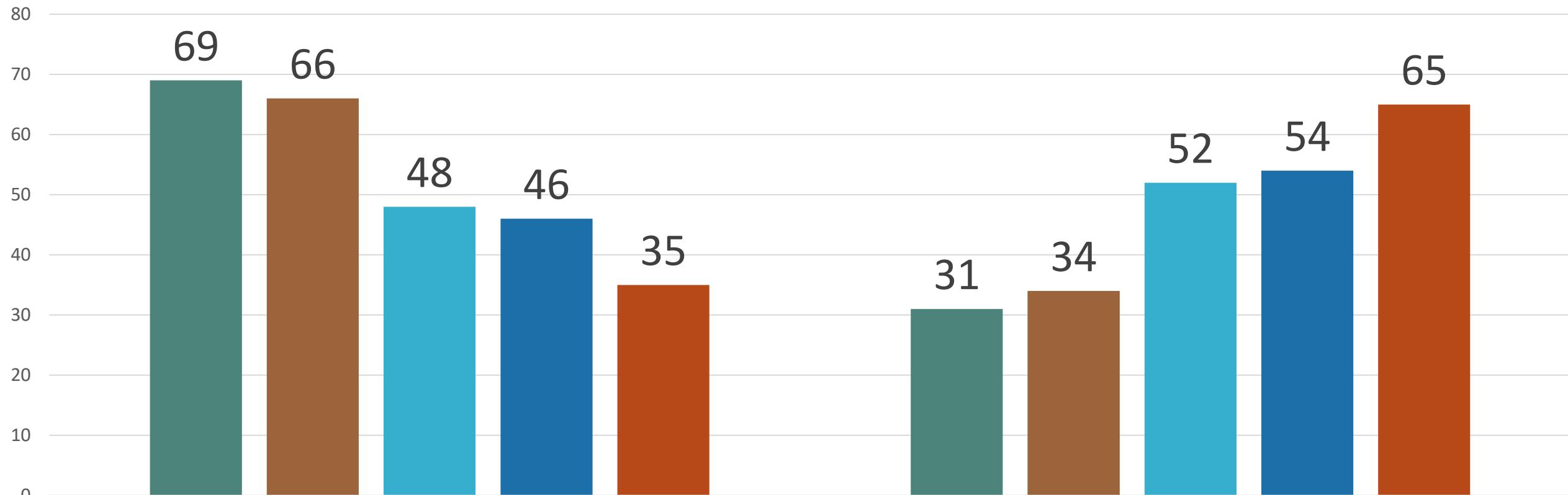
*da li koristi
pojedincu?*

Skalabilnost

Izazovi koji su redukovali rezultate timskog rada



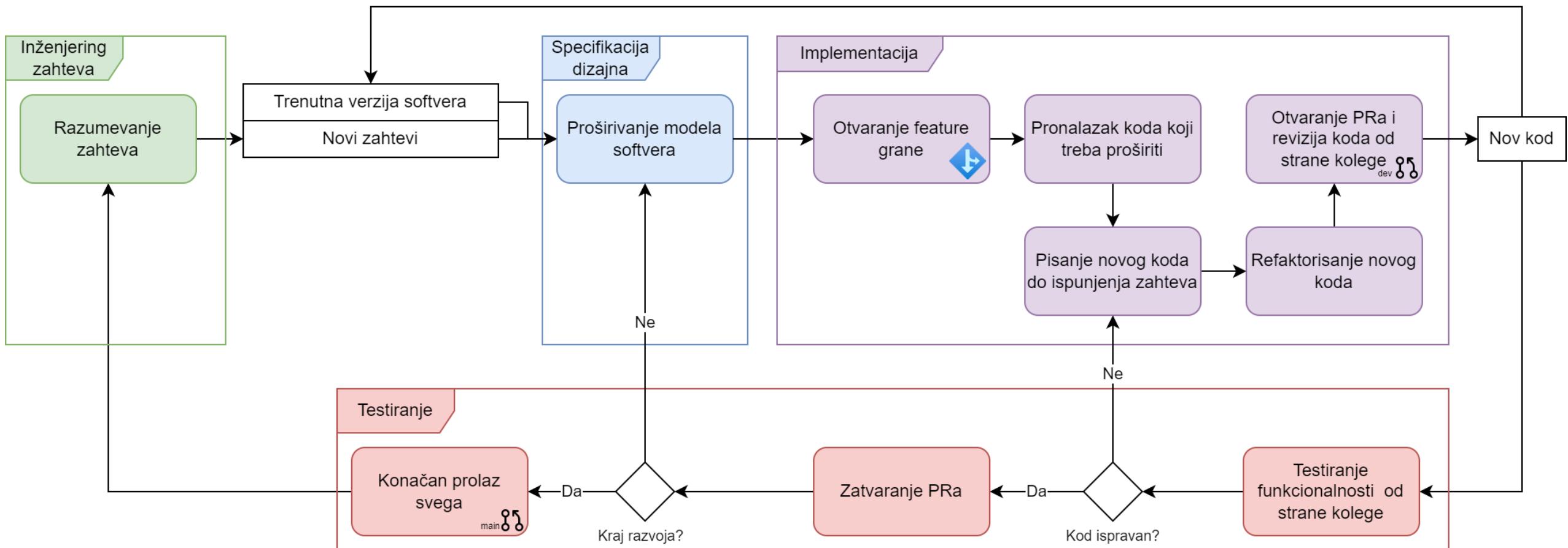
Izazovi koji su redukovali rezultate timskog rada

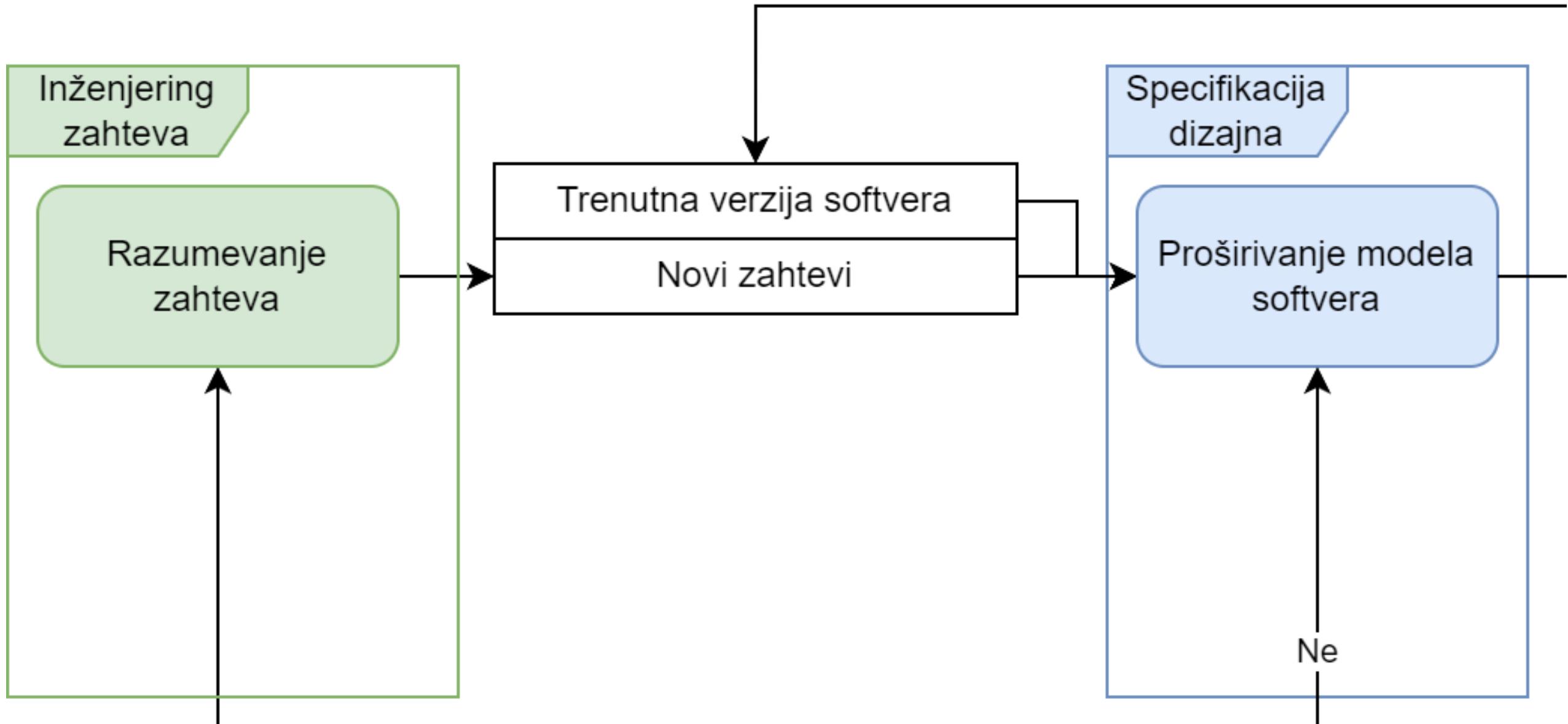


Izazov se nije ispoljavao ili je bio sitan problem

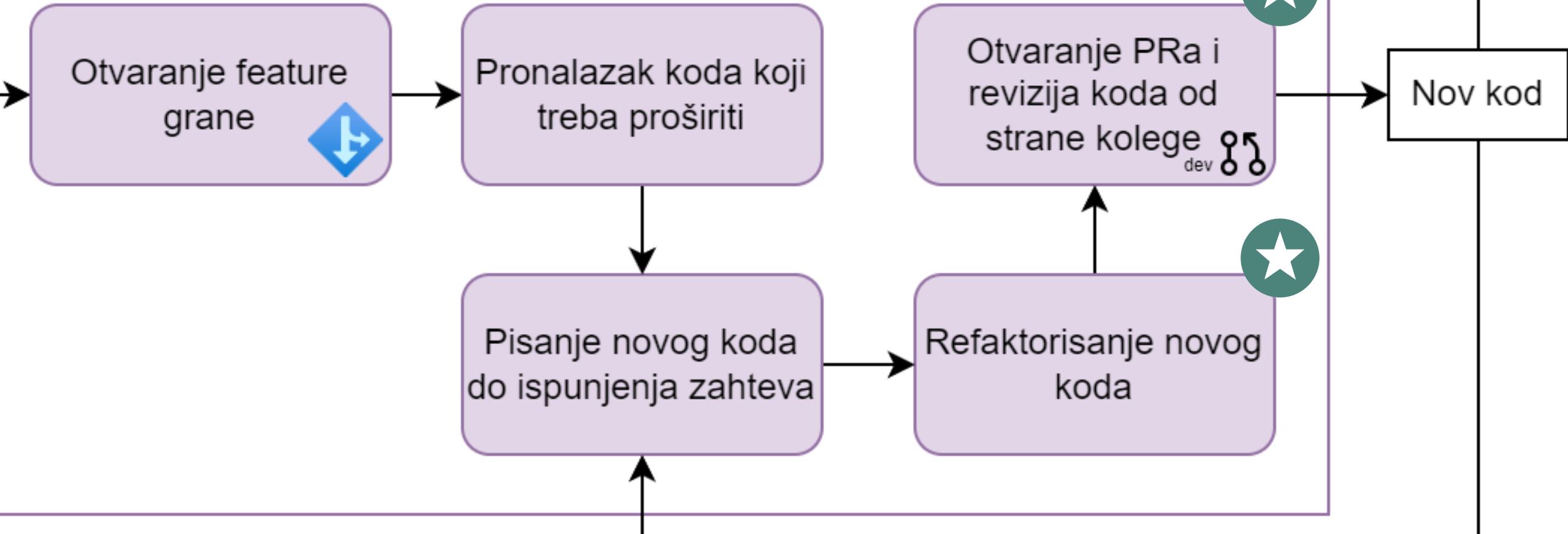
Izazov je bio umeren ili ozbiljan problem

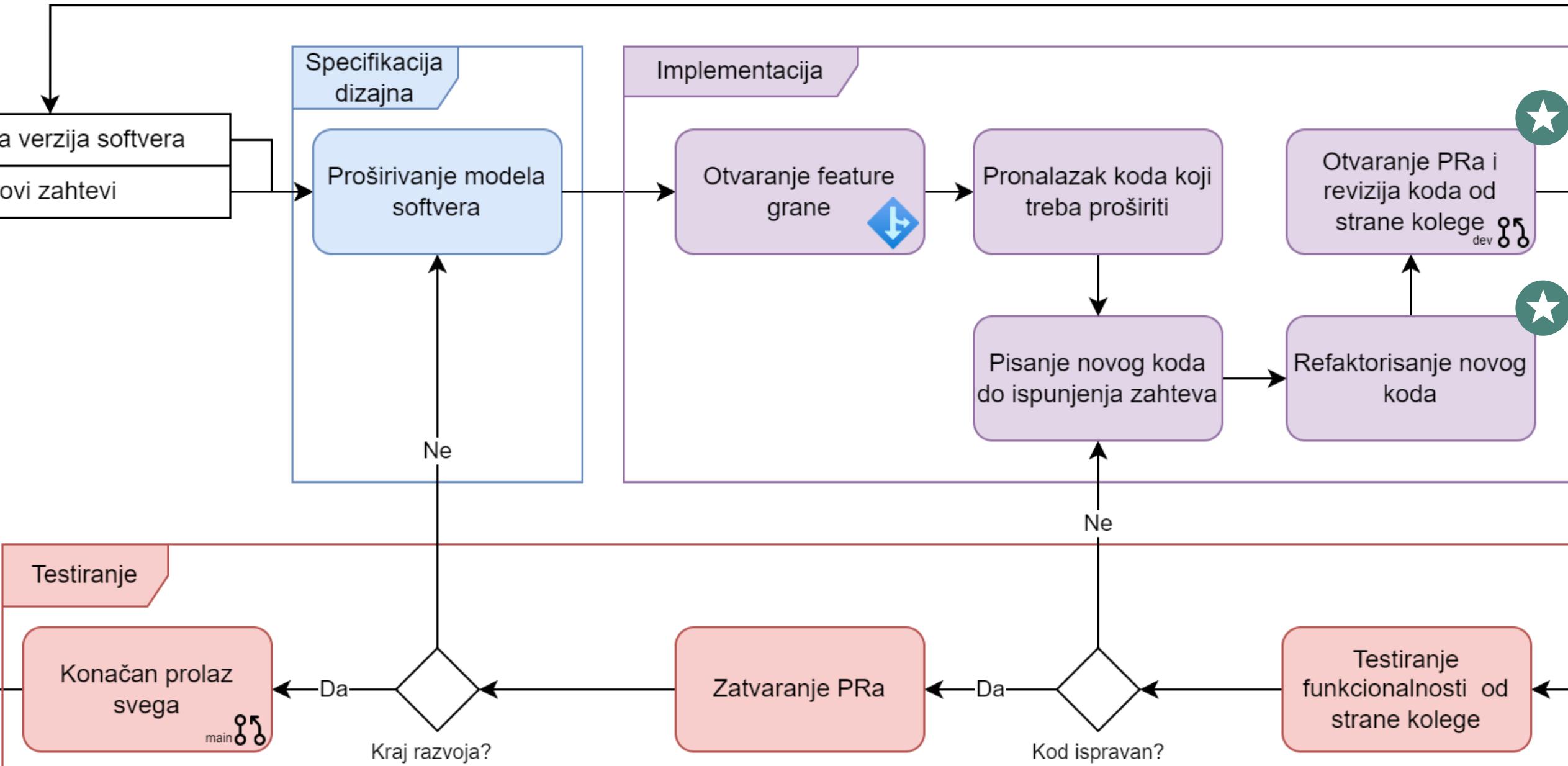
- Komunikacija je bila neiskrena i neefikasna jer smo previše pazili da jedni druge ne pritiskamo
- Članovi nisu ispunjavali nedeljne obaveze
- Članovi su imali različite standarde kvaliteta
- Članovi su imali različite ciljeve
- Članovi su imali različite stilove rada

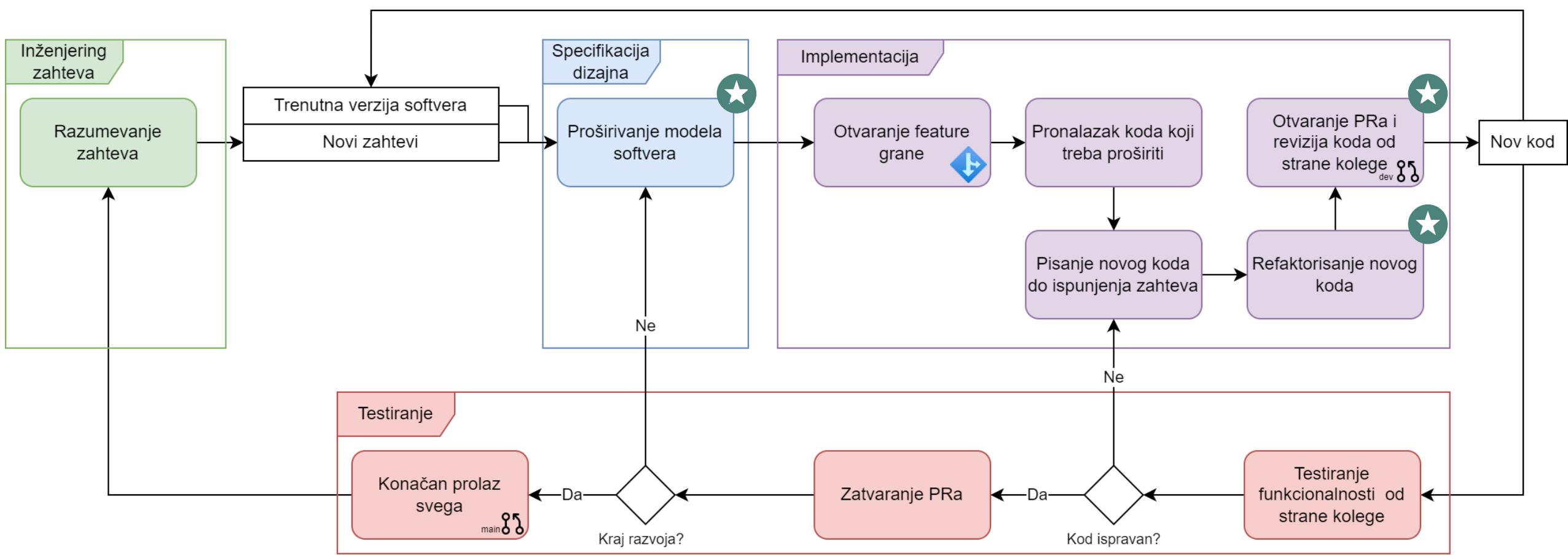




Implementacija



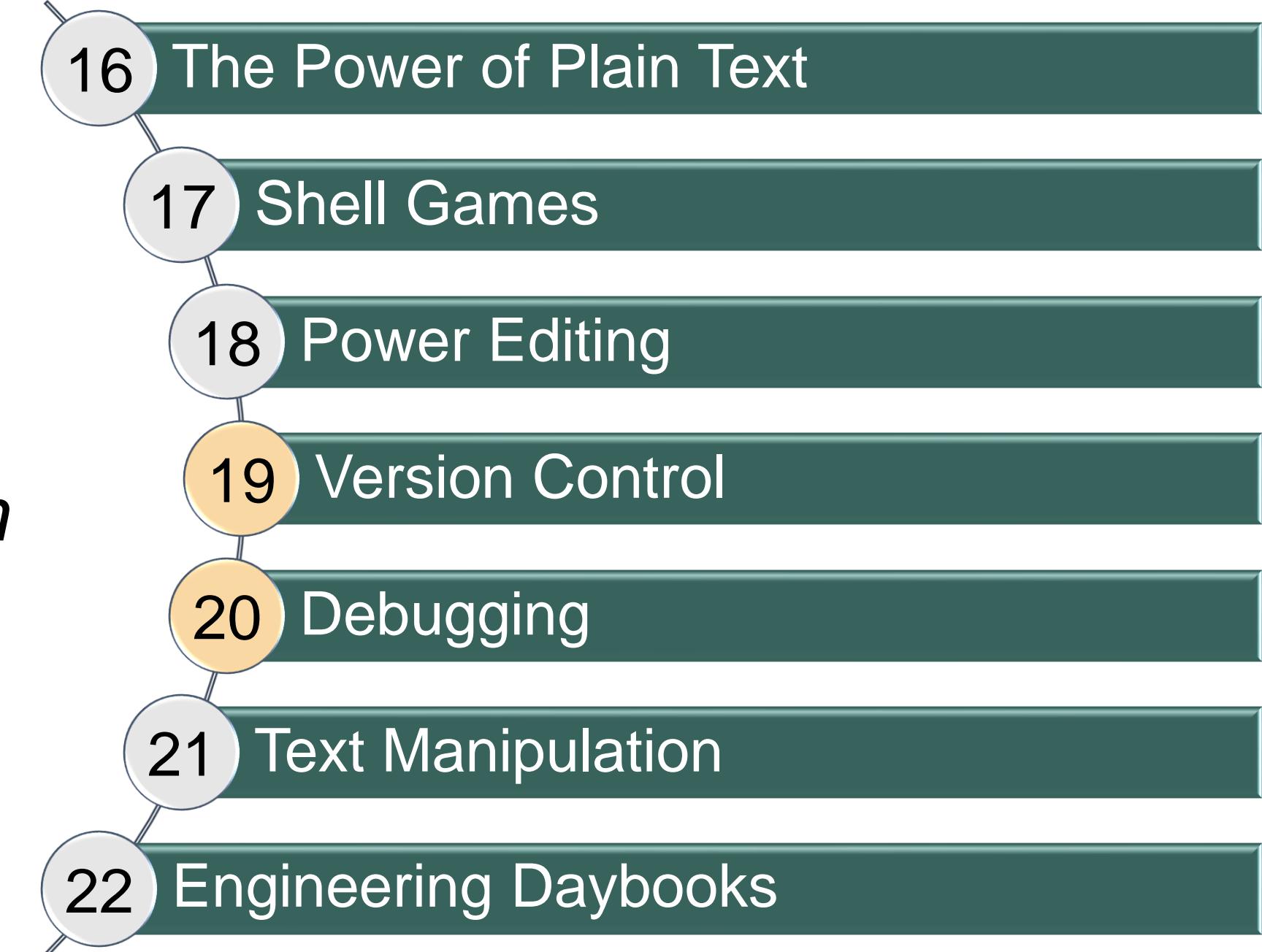




- Komunikacija je bila neiskrena i neefikasna jer smo previše pazili da jedni druge ne pritiskamo
- Članovi nisu ispunjavali nedeljne obaveze
- Članovi su imali različite standarde kvaliteta
- Članovi su imali različite ciljeve
- Članovi su imali različite stilove rada

Pro Git
Poglavlja 1, 2, 3

*Pragmatičan
Programer*
The Basic Tools



Za sledeći čas...

*Pragmatičan
Programer*
Before the Project

45

The Requirements Pit

46

Solving Impossible Puzzles

47

Working Together

48

The Essence of Agility