

Uvod u softversko inženjerstvo

Verzioniranje
koda

Nikola Luburić
nikola.luburic@uns.ac.rs



Verzioniranje koda

Razumevanje problema

Git: Osnovne komande

Git: Repozitorijumi i grane

GitHub: Kolaboracija, Akcije

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 2 nedelje možete da prikažete sve nove funkcije kompletne aplikacije
- ❖ Ne koristite Git, SVN, Mercurial...
- ❖ Gde skladištite kod? Kako integrišete promene? Šta kada se pojavi bug? Šta kada menjate isti fajl?

Skiciraj aktivnosti kao blok dijagram algoritma

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

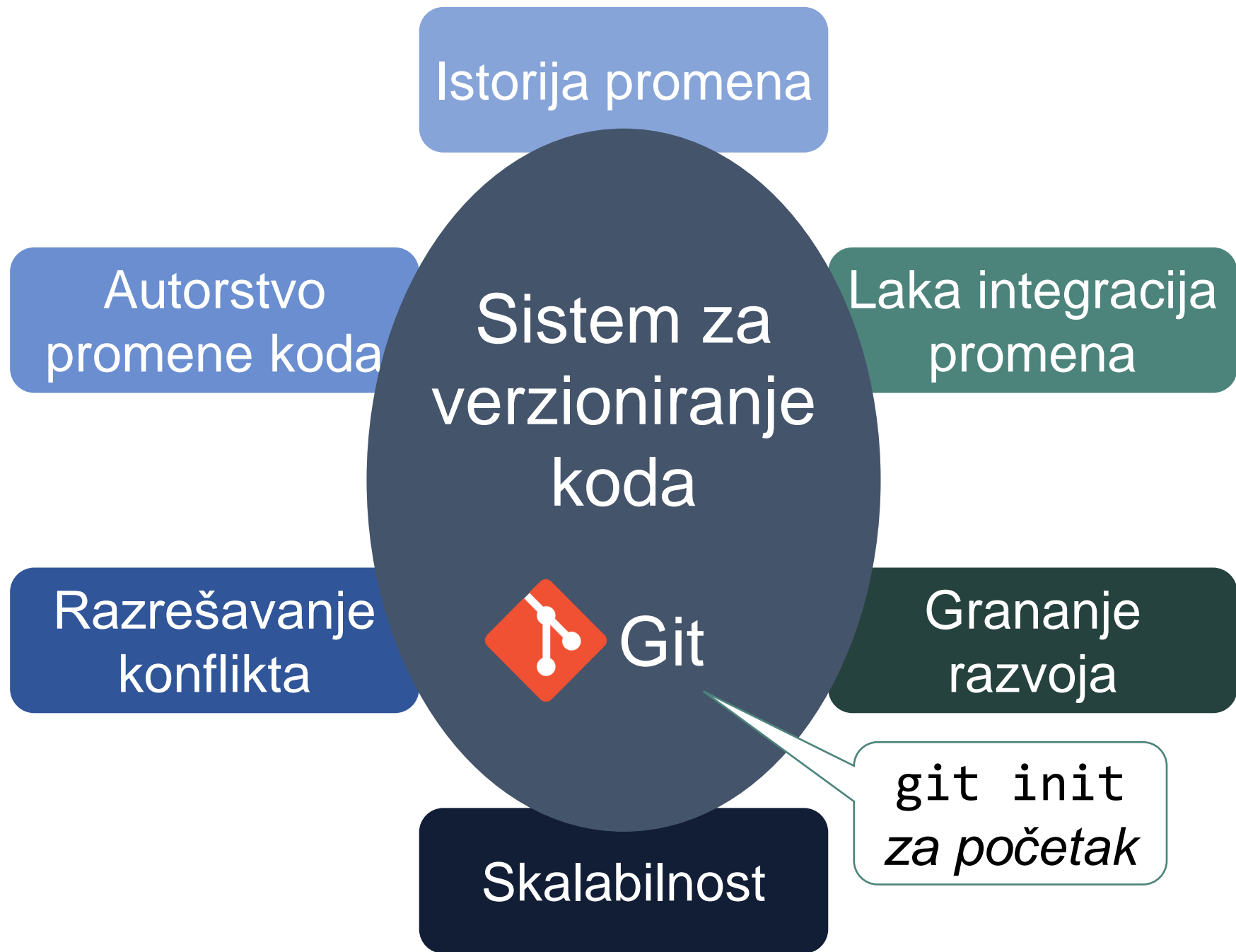
Autorstvo
promene koda

Laka integracija
promena

Razrešavanje
konflikta

Grananje
razvoja

Skalabilnost



Istorija promena

Radno stablo
(*work tree*)



Rectangle.java

git add

Pripremna zona
(*index*)

Skup
planiranih
izmena za
commit

git commit

Lokalni *git*
repozitorijum

**Trajna
evidencija
izmena**

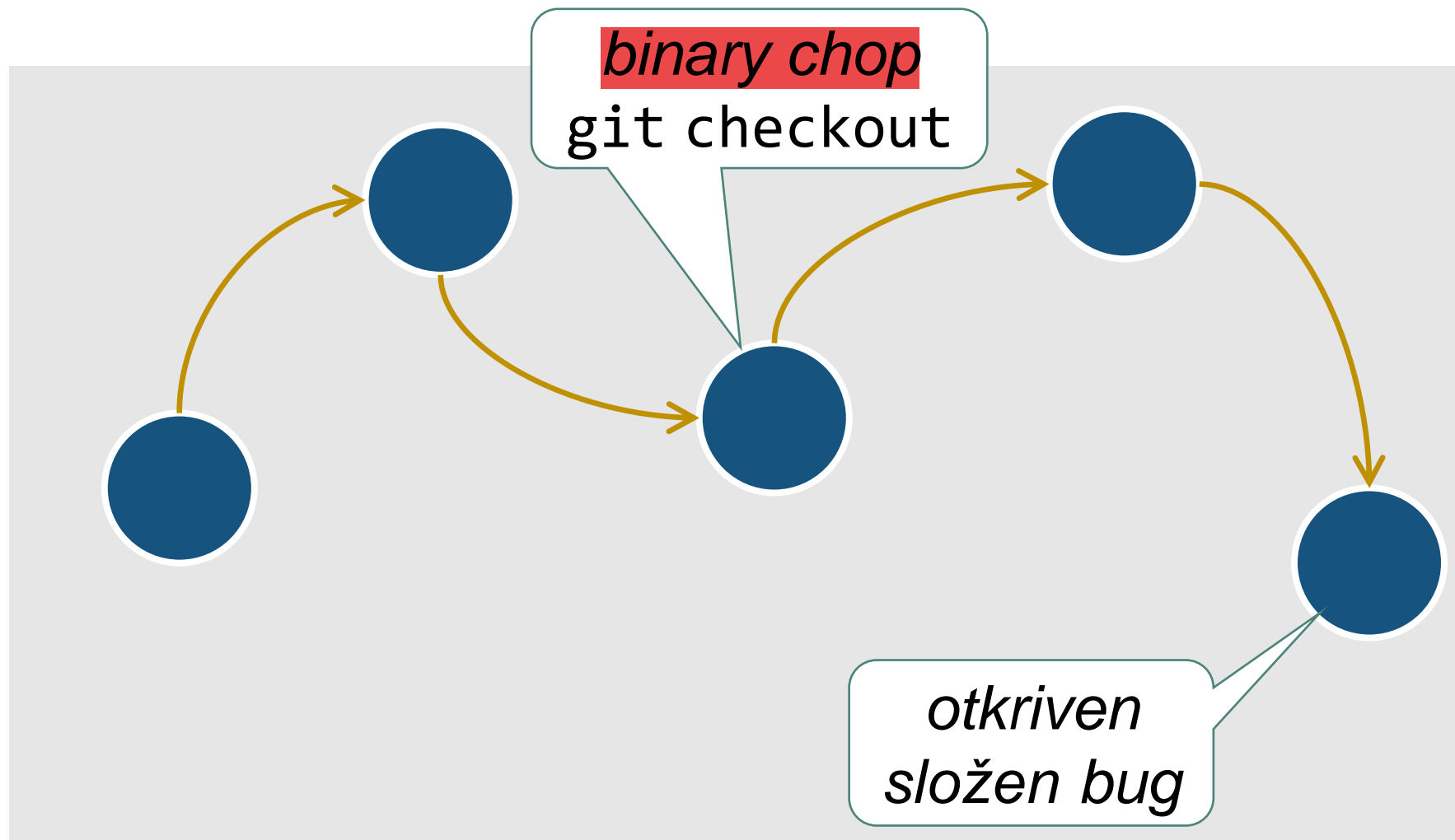


Istorija promena

```
public class Rectangle {  
    private Point[] points;  
    public surface() {...}  
}  
  
public class Point {  
    private int x;  
    private int y;  
    public dist(Point p) {...}
```

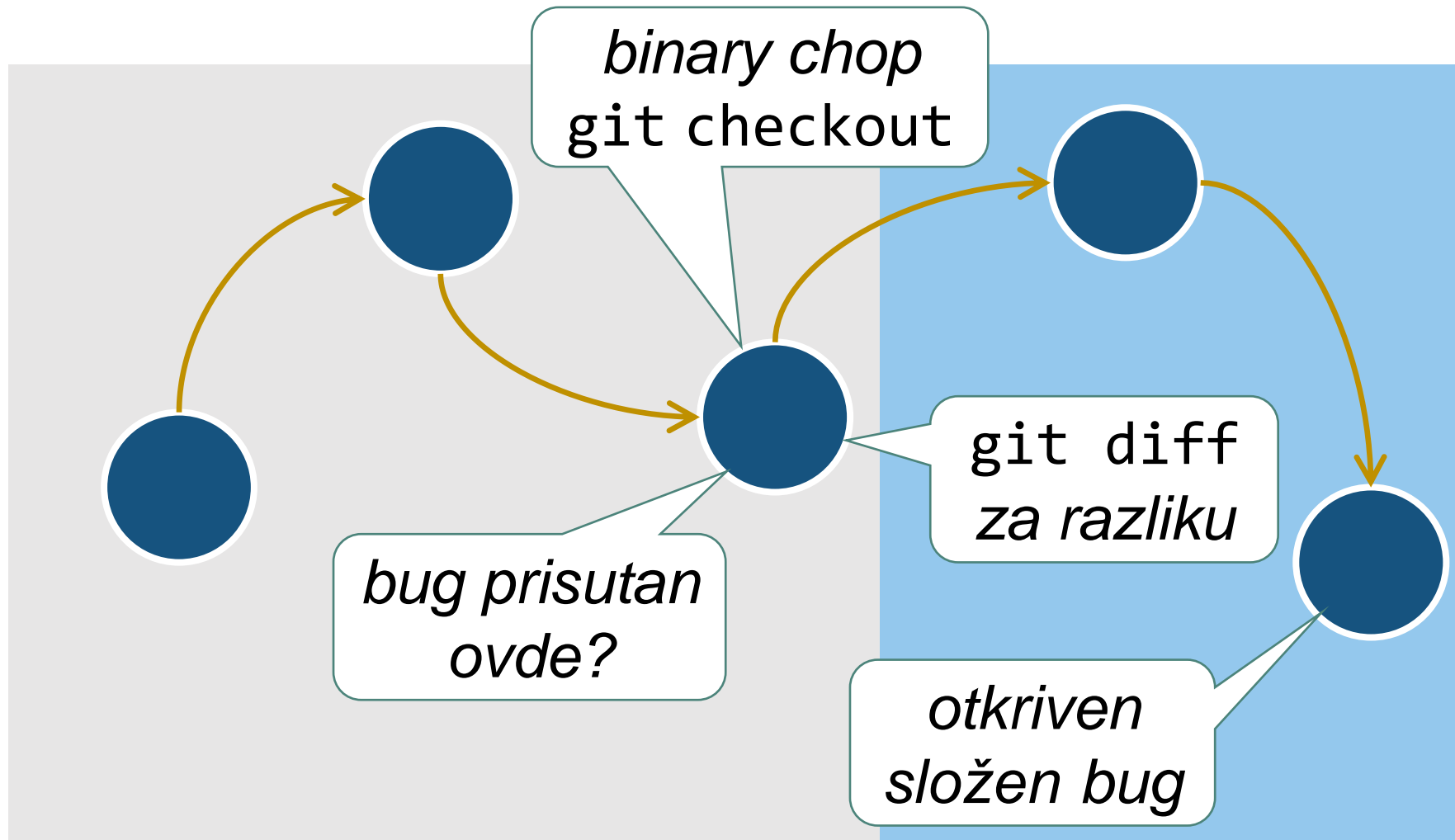
```
public class Rectangle {  
    private int[] xDots;  
    private int[] yDots;  
  
    public dist(...) {...}  
    public surface() {...}  
}
```


Istorija promena



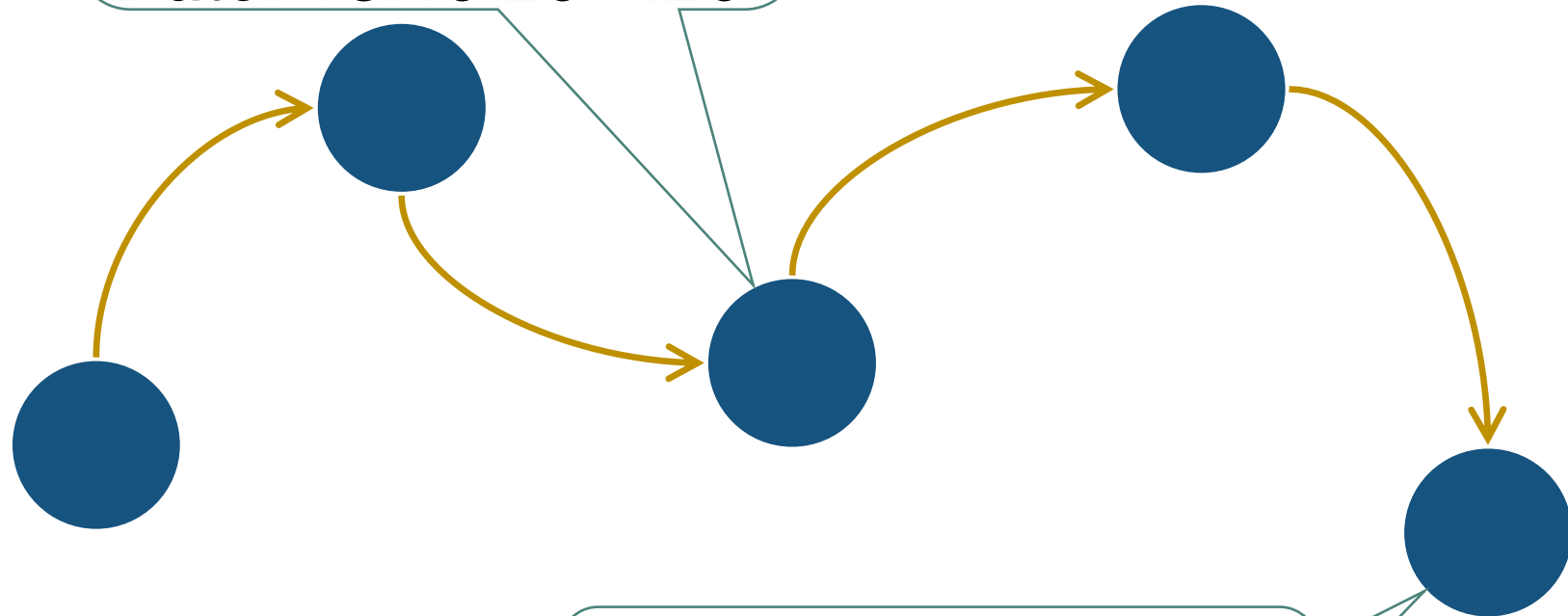
Istorija promena

Z: istraži git bisect komandu



Autorstvo
promene koda

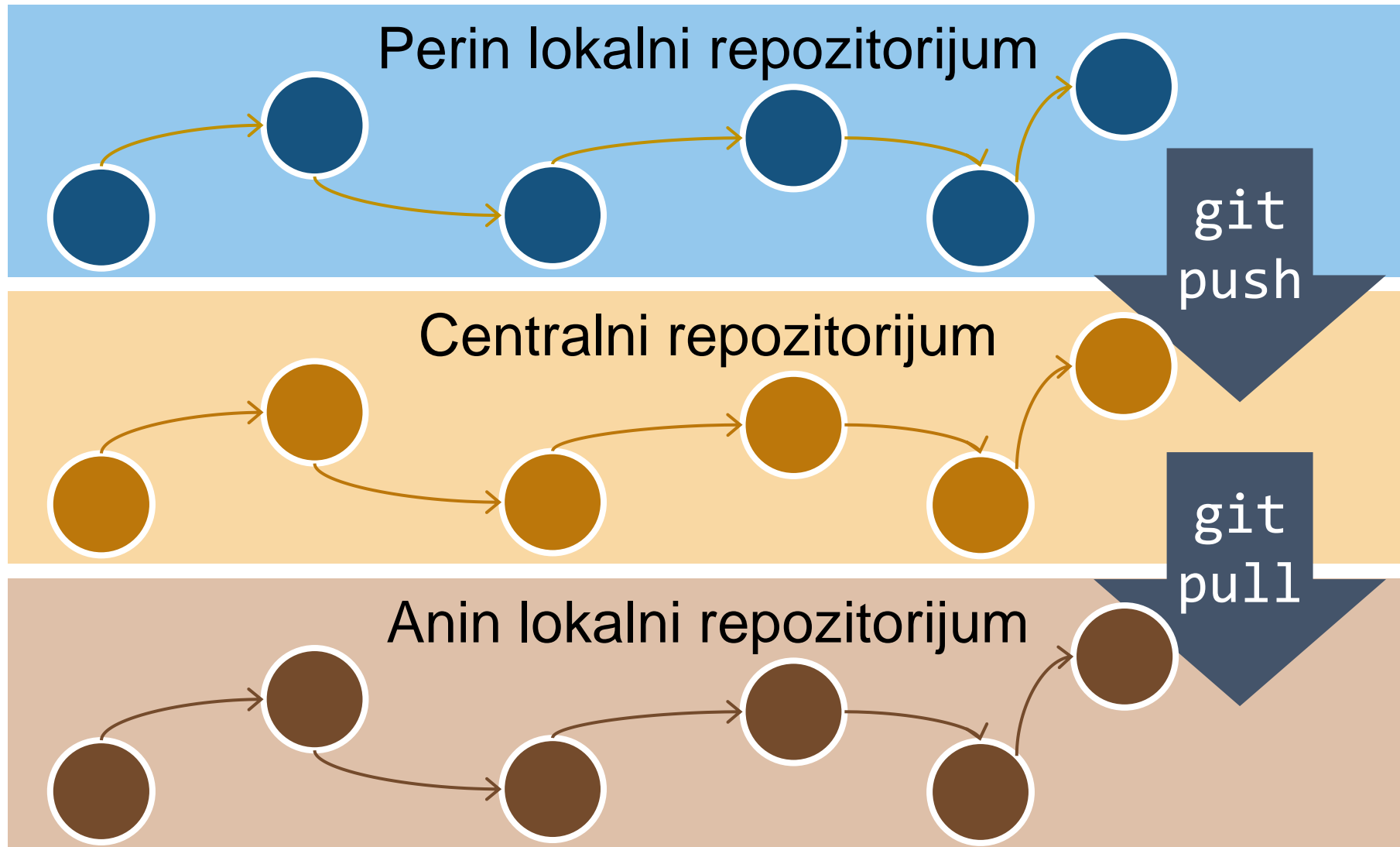
`git log`
commit 3fd...3sf
Author: Pera Perić
Date: 15:19 29.1.20.



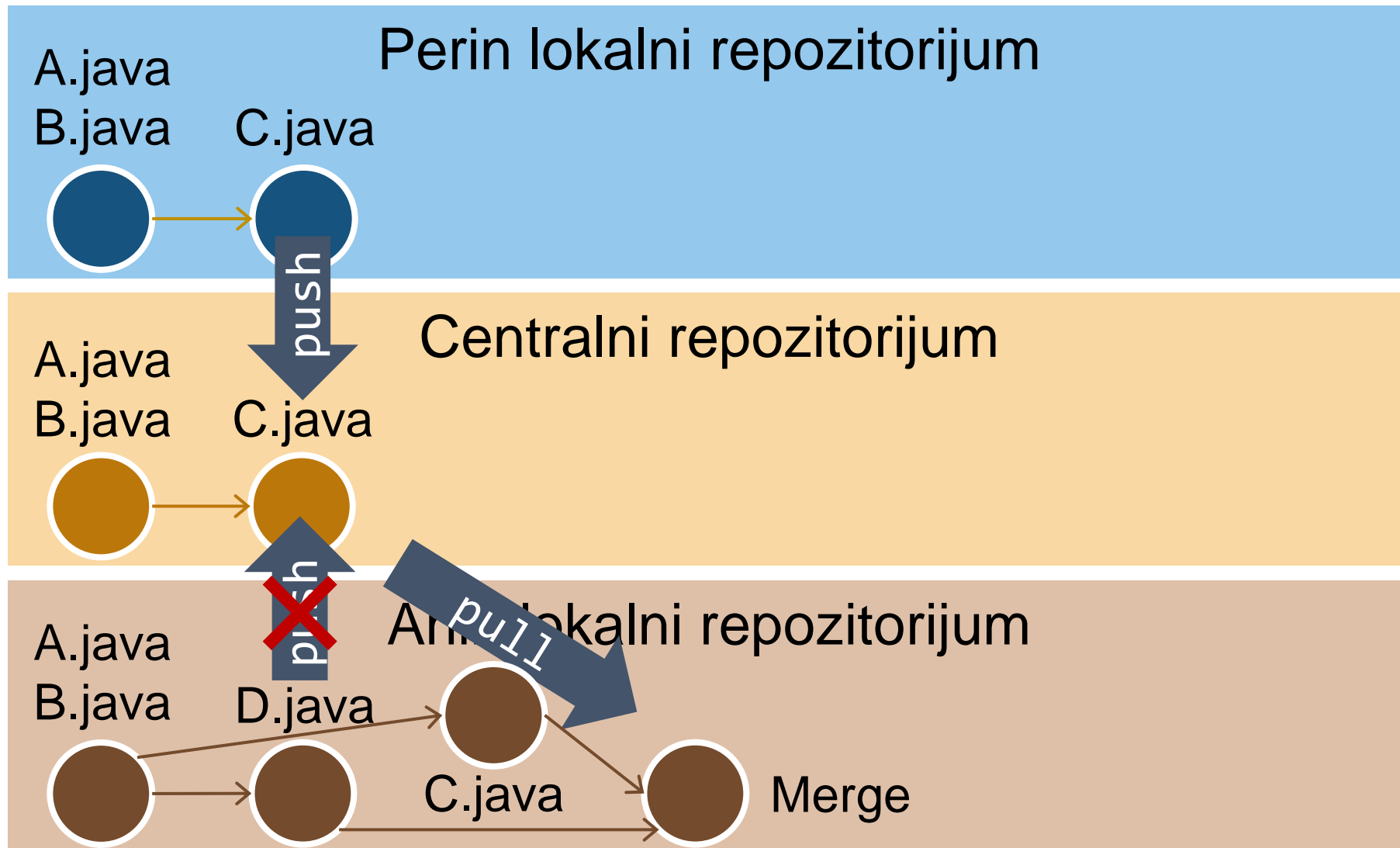
`gitk`
vizualizacija
istorije promena

`git blame <file>`
ko i kad je menjao šta

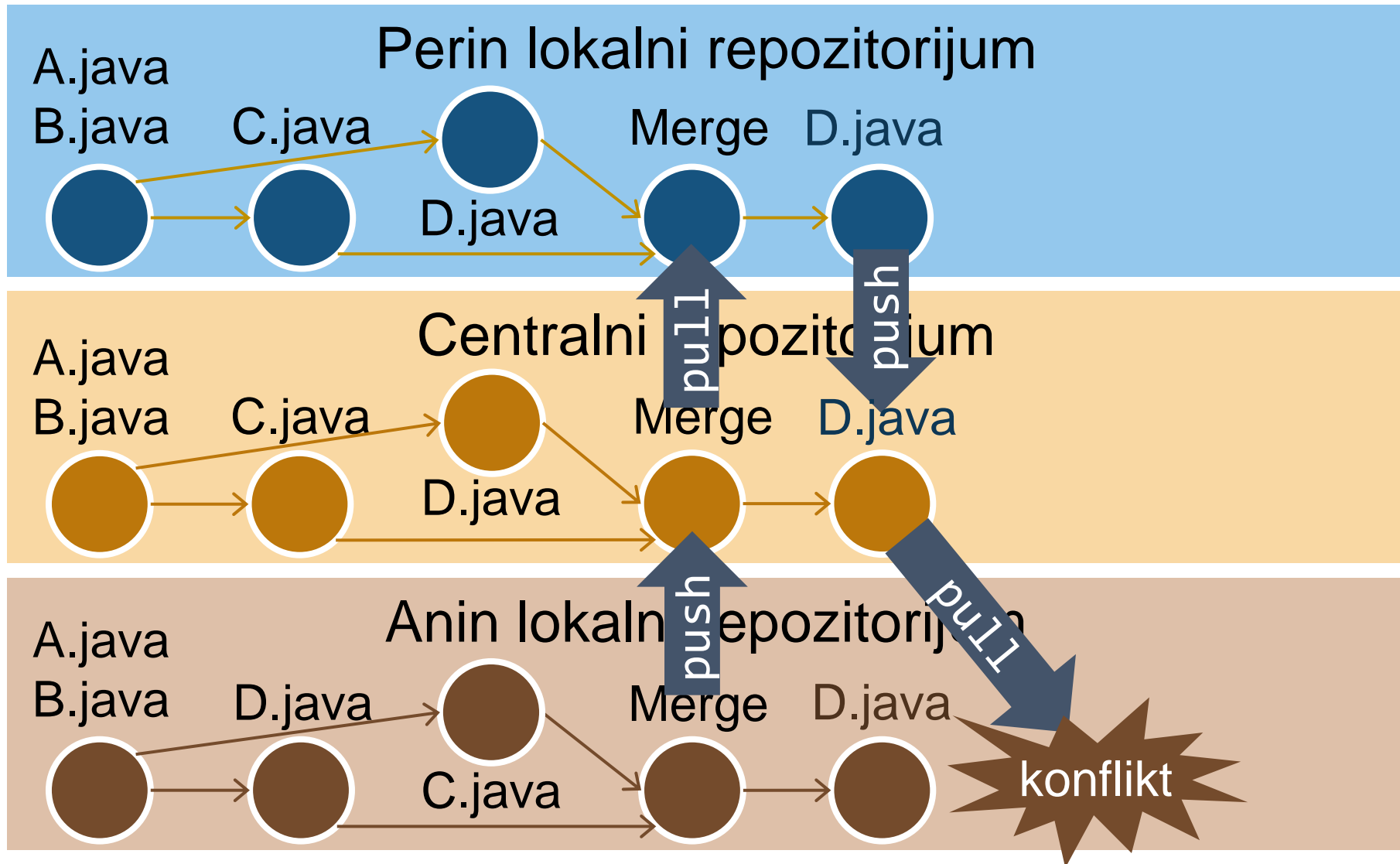
Laka integracija promena



Razrešavanje konflikta



Razrešavanje konflikta



Razrešavanje konflikta

„naše“ linije

*fajl gde je
konflikt*

<<<<<< HEAD:D.txt

Hello world

=====

Goodbye

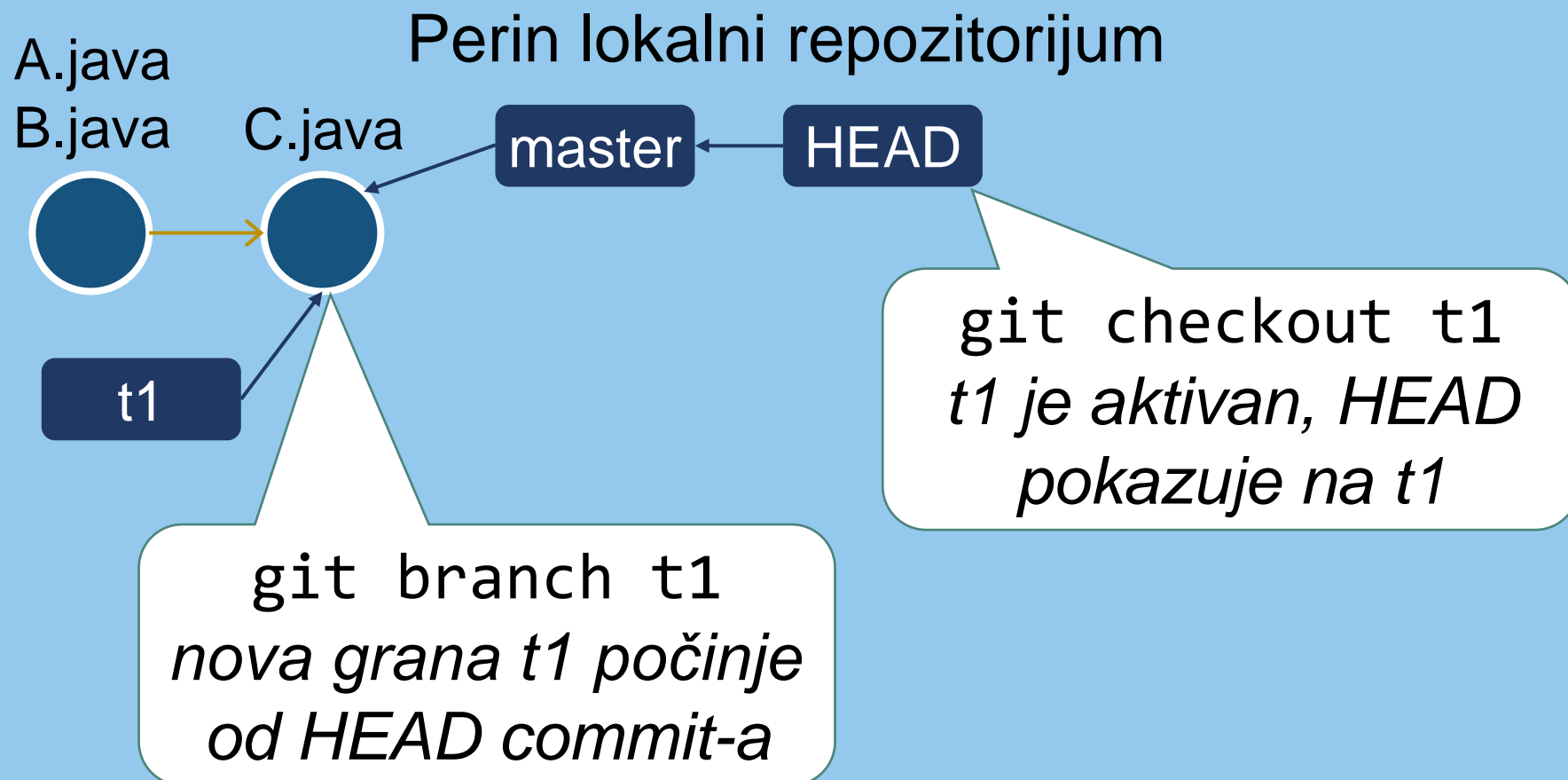
>>>>>> 7d3a...8s:D.txt

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

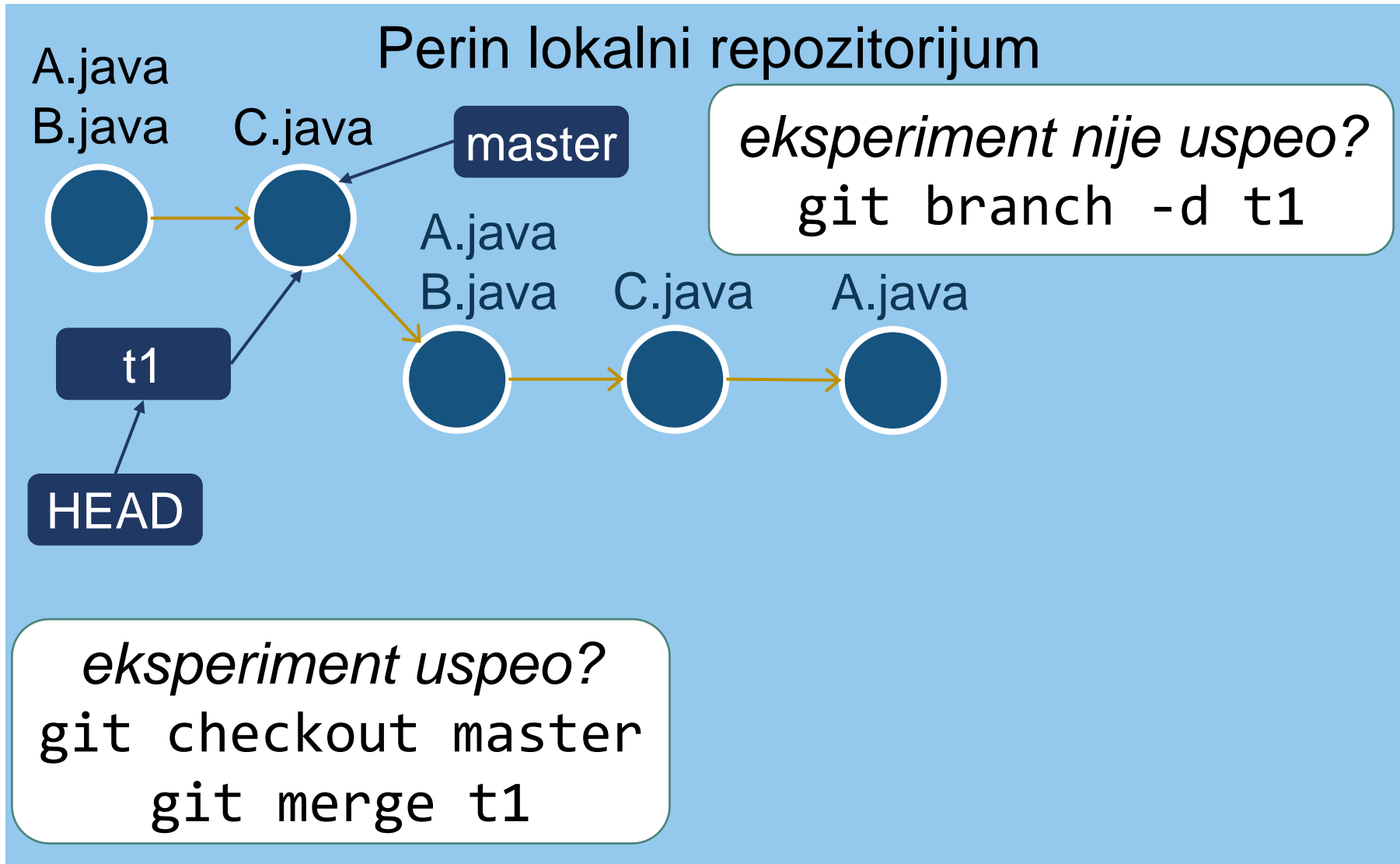
„tuđe“ linije

konflikt

Grananje razvoja

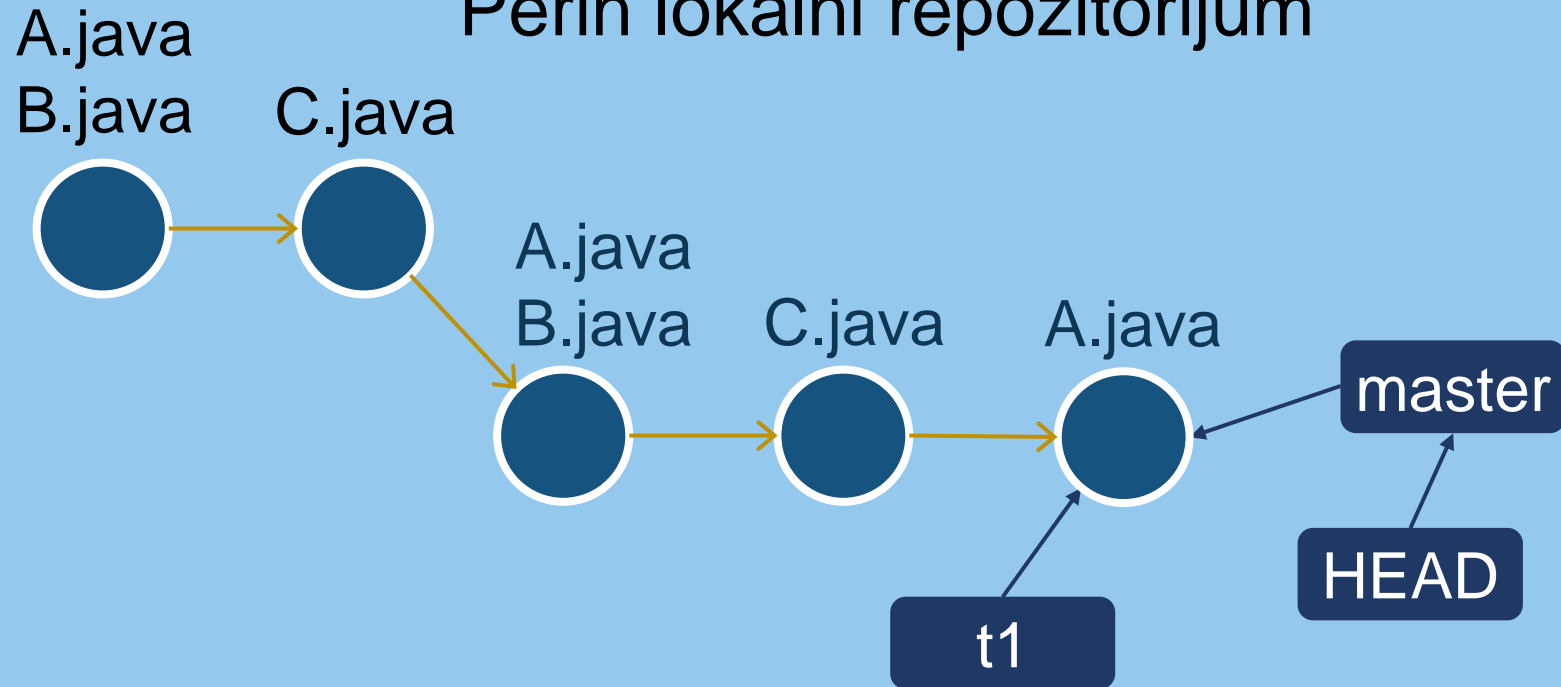


Grananje razvoja



Grananje razvoja

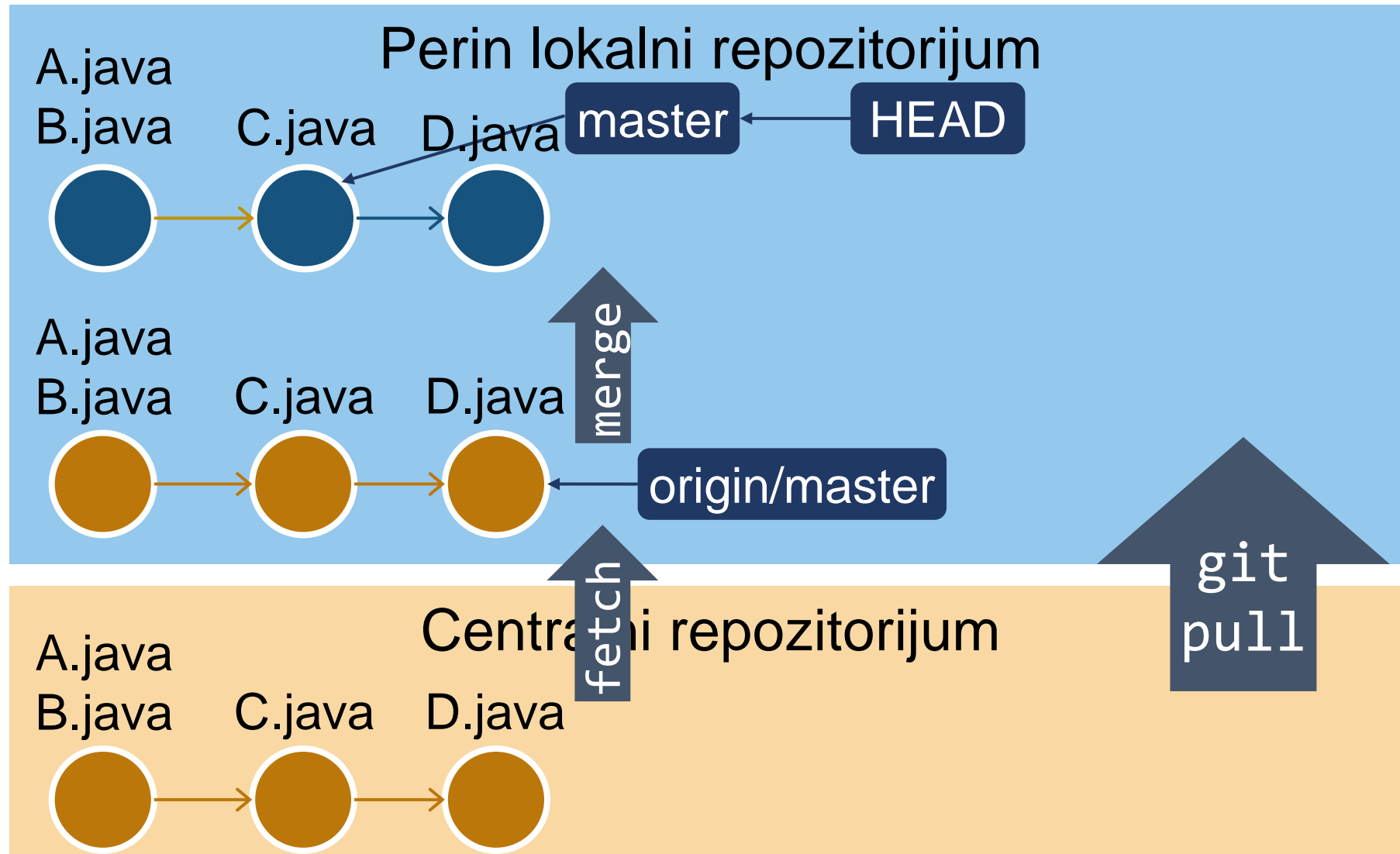
Perin lokalni repozitorijum



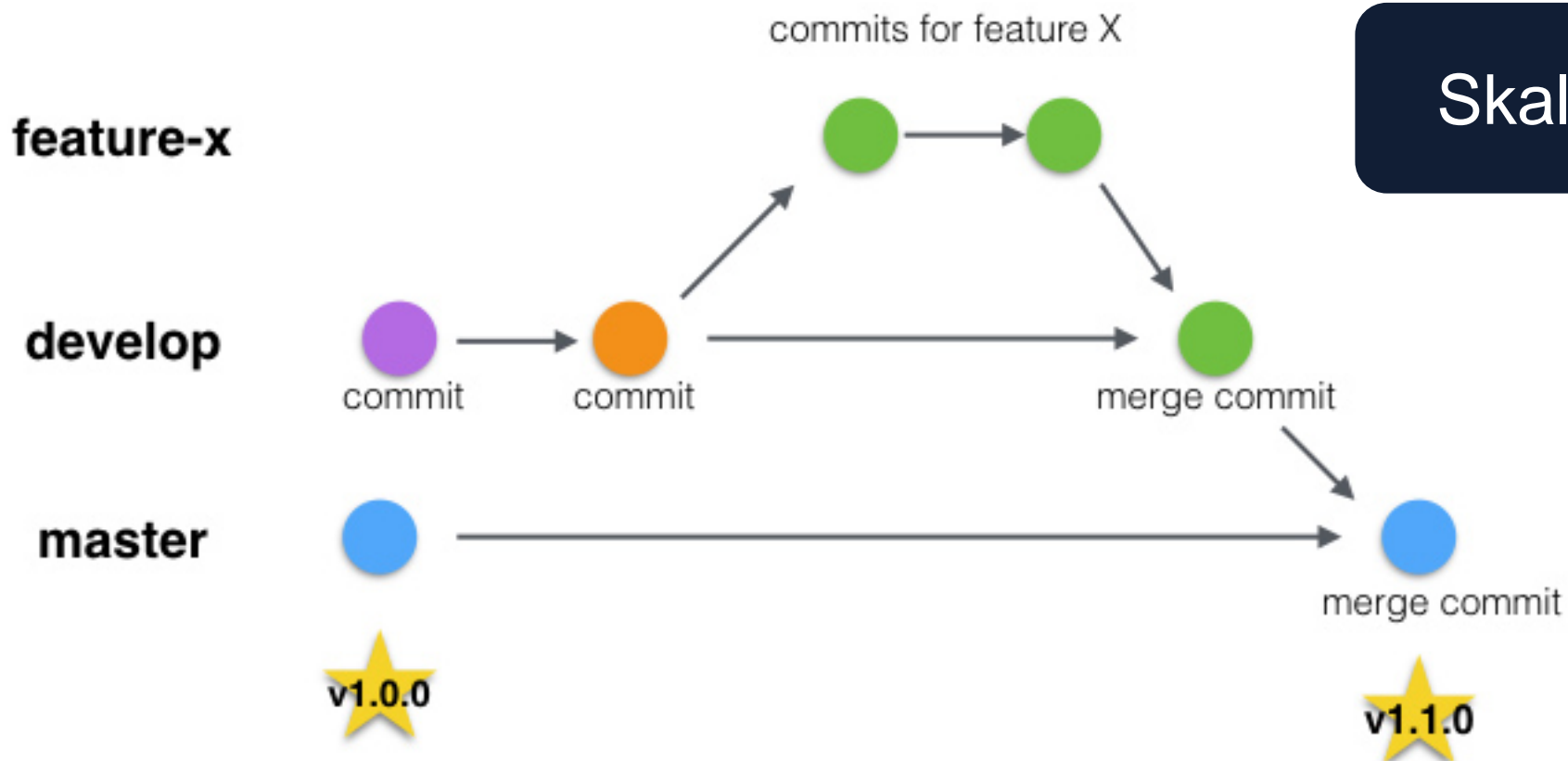
eksperiment uspeo?
git checkout master
git merge t1

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

Grananje razvoja



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.

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“
- Objašnjava šta je suštinska promena u implementaciji

*Posebno bitno
u timskom radu*

Skalabilnost

Filters is:pr is:open

Labels 9

Milestones 0

New pull request

3 Open ✓ 24 Closed

Author Label Projects Milestones Reviews Assignee Sort

Arbitrary AE event tracking ✓

#29 opened 5 days ago by milana-todorovic • Draft

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 + }
```



milana-todorovic 6 days ago

Collaborator Author ...

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.



Luburic 6 days ago

Member ...

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).



1

5

20

13

Pull request

Alat za reviziju
koda i kolaboraciju

Skalabilnost

Select workflow

New workflow

Showing runs from all workflows

Filter workflow runs

98 workflow runs

Event ▾ Status ▾ Branch ▾ Actor ▾

✓	KcMastery pass, completion, and satisfact	Build #110: Pull request #24 synchronize by milana-tod
✓	AE selection tracking, and more data added ...	Build #109: Pull request #25 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#

GitHub Actions
Alat za automatizaciju
razvoja softvera



Pro Git

Poglavlja 1, 2, 3

*Pragmatičan
Programer
The Basic Tools*

