

1

Upravljanje konfiguracijom

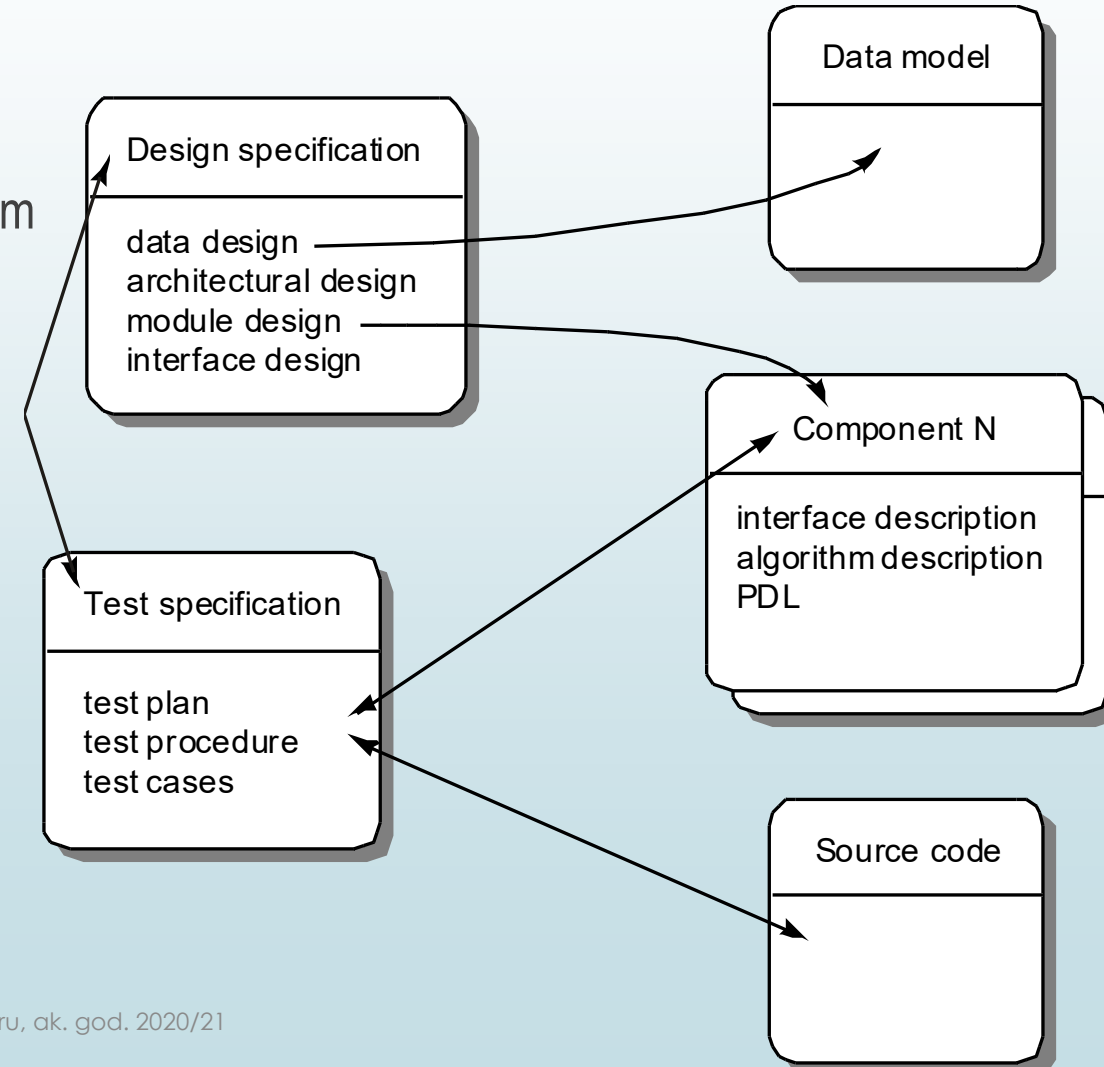
2020/21.03

➤ Konfiguracija

- imenovani skup konfiguracijskih elemenata u određenoj točki životnog ciklusa

➤ Element konfiguracije

- agregacija hardvera i/ili softvera koja se tretira kao jedinka u procesu upravljanja konfiguracijom
- može sadržavati bilo koji artefakt nastao u razvojnem procesu



Upravljanje konfiguracijom

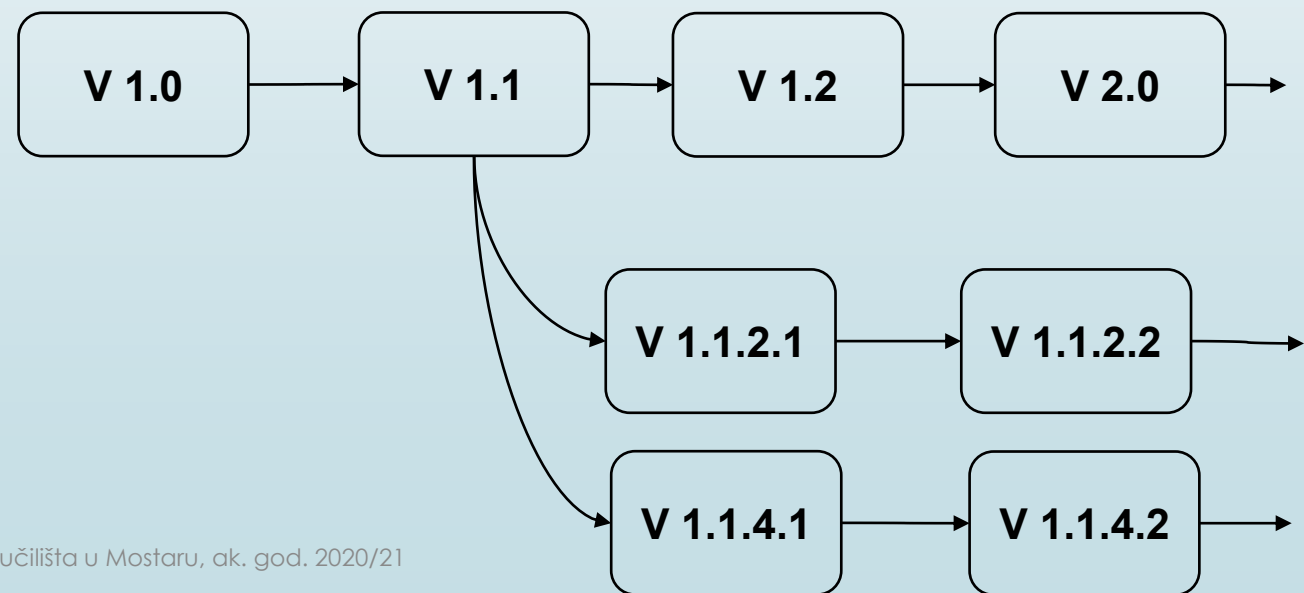
3

- Upravljanje konfiguracijom (engl. *configuration management*) je proces upravljanja promjenama prilikom razvoja softvera
- Sastoji se o 4 vrste aktivnosti
 - upravljanje zahtjevima za promjenu funkcionalnosti
 - proces procjene i odobravanja zahtjeva
 - upravljanje verzijama
 - evidentiranje različitih elemenata konfiguracije
 - upravljanje verzijama izvornog koda
 - integracija i (automatska) izgradnja sustava
 - upravljanje izdanjima
- Sustavi za podršku upravljanju verzijama i izgradnju obično omogućavaju i
 - praćenje problema (engl. *issues tracking*)
 - praćenje pogreški (engl. *bug tracking*)

Verzije konfiguracije

4

- verzija, inačica (version) – određeno izdanje (issue, release) proizvoda
- osnovica (Baseline) – specifikacija proizvoda formalno provjerena i usvojena, koja služi kao temelj razvoja i koja se mijenja samo kroz formalnu proceduru kontrole promjena, IEEE (IEEE Std. No. 610.12-1990)
- objava, isporuka (release) – verzija isporučena korisnicima
- revizija (revision) – služi umjesto originalne, podrazumijeva izmjene nastale kroz vrijeme (npr. zbog ispravljanja pogrešaka), npr. V1.2
- varijanta (variant)
 - alternativa originalu (hardverska platforma, različiti jezik),
 - živi paralelno s njim, npr. v1.1.2.1



Označavanje verzija (1)

5

- Postupak pridjeljivanja jedinstvene oznake, imena i broja određenoj verziji
 - ne mora biti ista za internu upotrebu i za isporuku
- Šarolika praksa označavanja, ali većinom u praksi dominira oblik koji počinje s *major.minor*
- Major version/release
 - predstavlja veliko izdanje.
 - vrijednost se povećava prilikom znatne promjene funkcionalnosti
 - Može izazvati prekid kompatibilnosti u odnosu na prethodnu verziju
- Minor version
 - malo izdanje
 - promjene mogu biti značajne, ali dodavanjem novih funkcionalnosti i uz zadržavanje kompatibilnosti s prethodnim verzijama

Označavanje verzija (2)

6

- Verzija objektne datoteke u .NET Frameworku (assembly) određena je s četiri broja:

`<major version>.<minor version>.<build number>.<revision>`

- major version - mijenja se prilikom znatne promjene u (npr. kod redizajna koji prekida vertikalnu kompatibilnost sa starijim verzijama)
- minor version - mijenja se prilikom znatne promjene, ali uz zadržavanje kompatibilnosti s prethodnim verzijama
- build number - predstavlja ponovno prevođenje istog koda (npr. prilikom promjene platforme, procesora i slično)
- revision - primjenjuje se npr. prilikom izdavanja sigurnosnih zakrpa i sličnih manjih promjena

- Primjer: Properties \ AssemblyInfo

- major.minor.* (ili major.minor.build.*) automatski određuje build number i revision
 - build number: broj dana od 1.1.2000.
 - revision: broj sekundi proteklih od ponoći aktualnog dana podijeljen s 2

Označavanje verzija (3)

7

- .NET Core koristi oblik major.minor.patch-sufix
 - Semantic versioning <https://semver.org/>
 - Sufiks može biti proizvoljan
 - alfa, beta, rc-2, ...
 - Nova verzija obično biva označena jednim brojem više na odgovarajućoj poziciji
- Aplikacije za Google Play Store
 - Korisnik vidi oznaku verzije koju razvojni tim odabere (versionName)
 - Interno se koristi pozitivni cijeli broj (versionCode)
 - raste neovisno o tome povećava li se malo ili veliko izdanje
 - služi za uspostavljanje odnosa među verzijama (prethodna ili nadogradnja)

Označavanje verzija (4)

8

- Tex – trenutna verzija 3.14159265
 - Svaka nova verzija je nova znamenka broja π
- .NET Core 3.* → .NET 5
 - Preskače se .NET Core 4 da ne izaziva zabunu s .NET Framework 4
 - Mijenja se naziv zbog unificiranja razvoja platformi
- Java 1.0 → Java 1.1 → ... → Java 1.4 → Java 5 (umjesto 1.5)
 - Isticanje srednje znamenke verzije
- Eclipse, Windows 10, Ubuntu, ... upotrebljavaju godinu i mjesec
 - Ubuntu 20.04, 20.10, ...
 - Windows 10 1909, 2004 + build number.update build revision
- Word 2.0 → Word 6.0 → Word 95 → ...
- ... i puno drugih varijanti kao posljedica usklađivanja varijanti, marketinga, izbjegavanja nesretnih brojeva i slično

Automatsko i ručno verzioniranje

9

➤ Automatsko označavanje

➤ prednosti:

- eliminacija ručnog rada (npr. pisanja i izvedbe skripti)
- ne postoje dvije inačice s istom oznakom

➤ nedostaci:

- oznaka elementa ne podudara se s oznakom cijelog sustava
- novi brojevi ovise o danu i vremenu prevođenja
- verzija se mijenja pri svakom prevođenju, neovisno o tome jesu li se dogodile promjene ili ne

➤ Ručno verzioniranje

➤ prednosti:

- potpuna kontrola nad brojevima verzije
- moguća je sinkronizacija između verzije pojedinih komponenti i verzije cijelog sustava

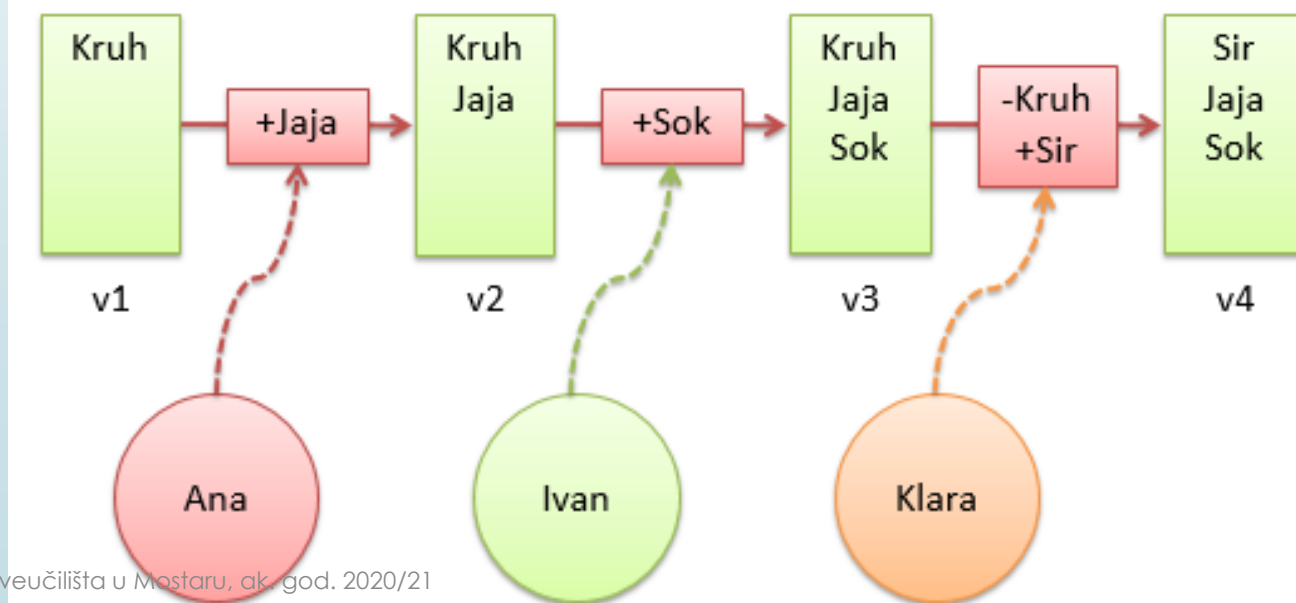
➤ nedostaci:

- verzioniranje se mora raditi ručno
- moguće je napraviti više različitih objektnih datoteka s istim oznakama

Upravljanje verzijama (izvornog koda)

10

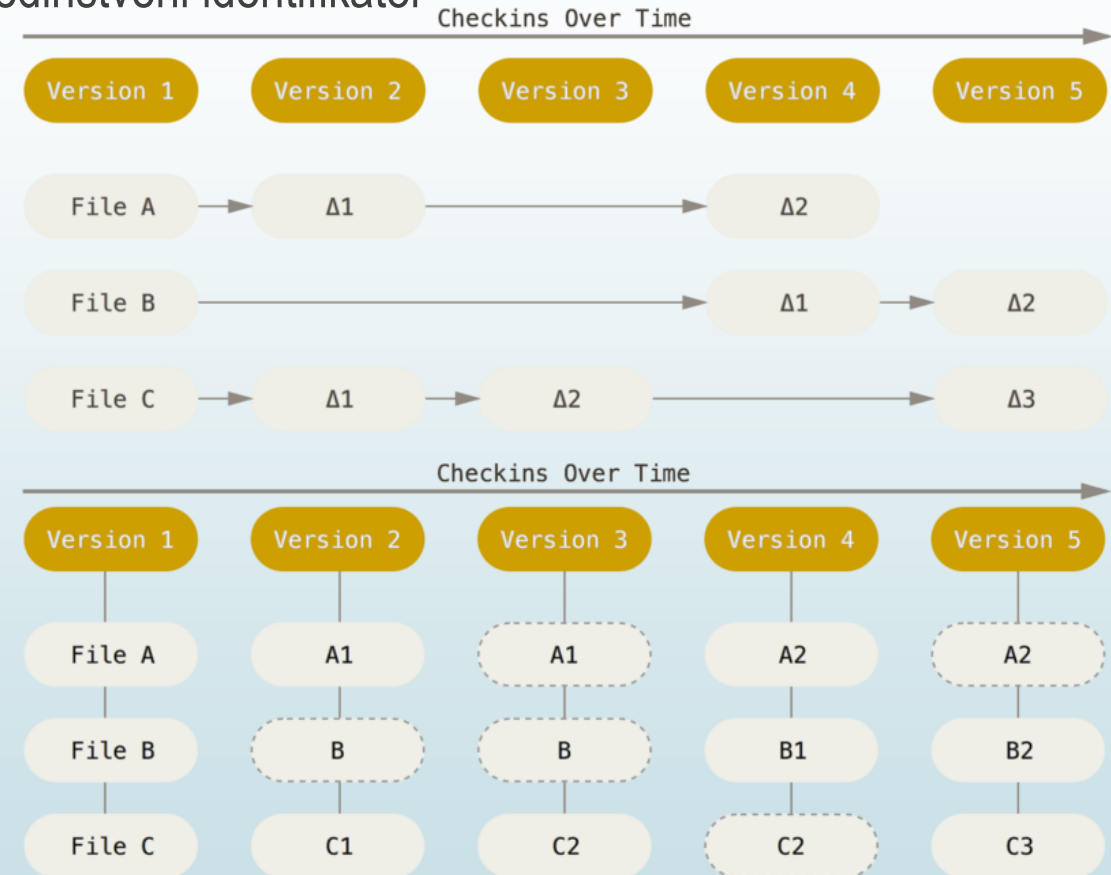
- Alati za upravljanje verzijama (*Version control systems*) obično omogućavaju
 - identifikaciju verzije i izdanja
 - bilježenje povijesti promjena
 - podršku za neovisni razvoj više programera nad istim softverom
- Najčešće kontrola promjena izvornog koda i konfiguracijskih datoteka
 - ne evidentirati automatski generirane datoteke!
- Repozitorij sadrži zadnju verziju, ali implementira mehanizam rekonstrukcije određene verzije iz povijesti (npr. evidentirajući razlike)
- Niz verzija nastalih izvođenjem iz prethodnih verzija naziva se *codeline*



Karakteristike sustava za upravljanje verzijama

11

- Identifikacija verzija i izdanja
 - Svaka verzija pohranjena u repozitoriju ima jedinstveni identifikator
- Jedan skup promjena može sadržati promjene na više datoteka
- Kompaktna pohrana
 - Umjesto kopije svake od verzija čuva se zadnja verzija te lista razlika između susjednih verzija



Chacon, Straub, ProGit 2nd Edition, Apress, 2014.

Centralizirani sustavi za upravljanje verzijama

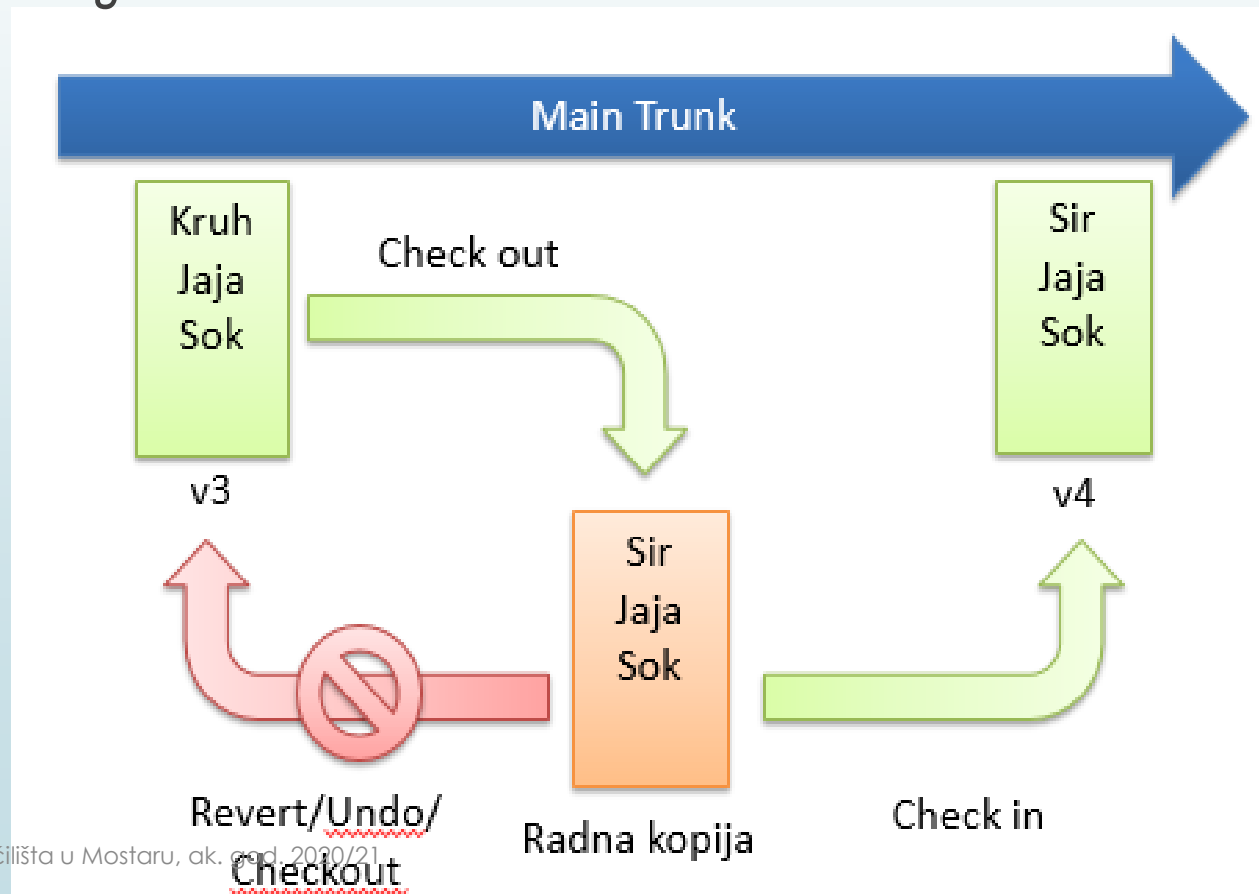
12

- Subversion, CVS, Team Foundation Server, ...
- Samo jedan centralni repozitorij
- Korisnik ima radnu kopiju (working copy, workspace)
 - Za svaku datoteku radna kopija sadrži određenu verziju iz centralnog repozitorija i informaciju o kojoj verziji se radi
 - Neki od alata (npr. TFS) dopuštaju da radna kopija sadrži samo dio centralnog repozitorija
- Promjene na radnoj kopiji započinju *checkoutom*
 - Najava izmjene datoteke
 - Može uključivati prethodni dohvat zadnje verzije datoteke i onemogućavanje ažuriranja te datoteke drugim korisnicima

Promjene na radnoj kopiji

13

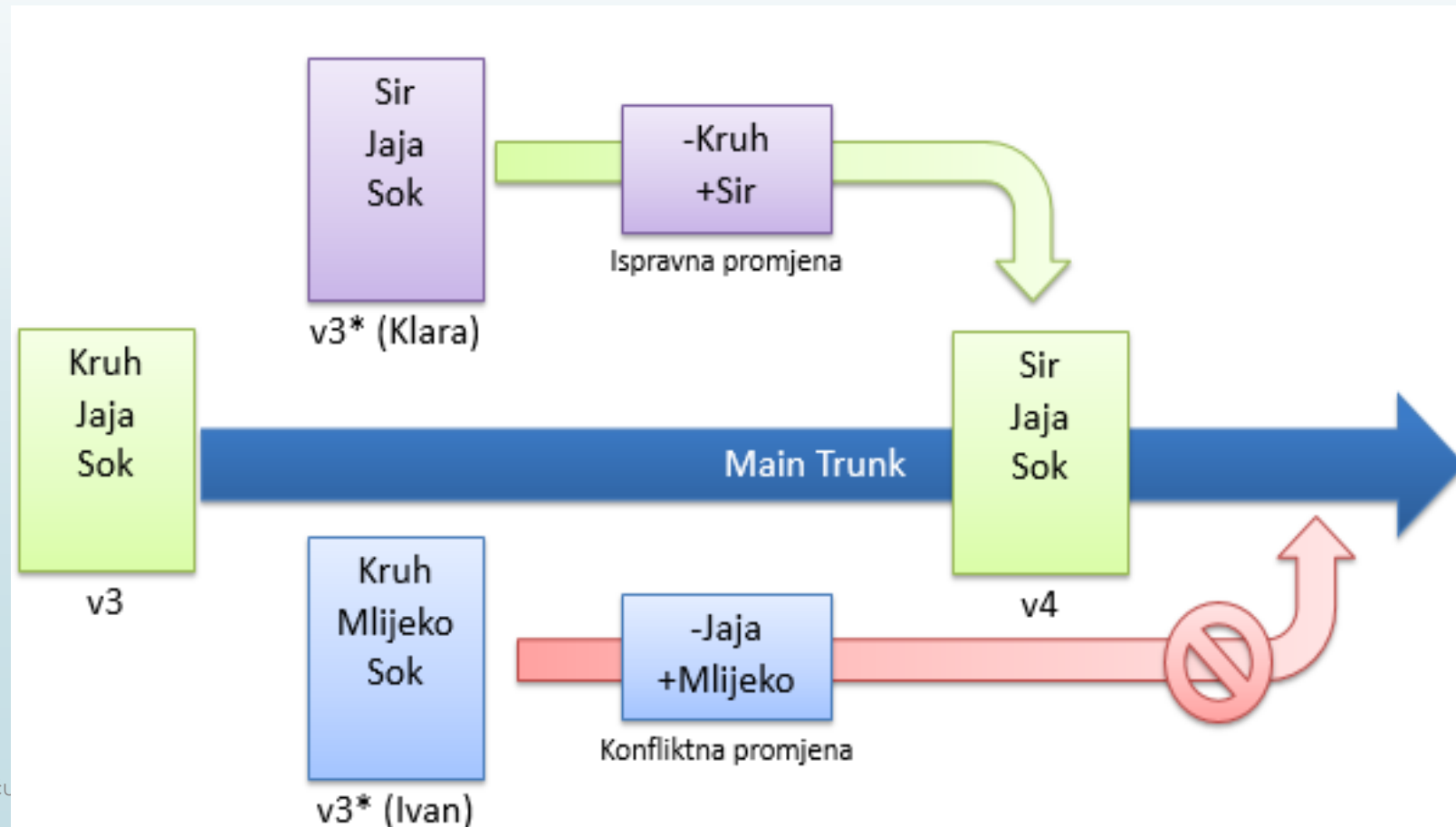
- Korisnik promjene na radnoj kopiji može
 - odbaciti (*undo, revert, checkout*)
 - potvrditi slanjem na centralni repozitorij (*check-in, commit*)
- Skup poslanih promjena naziva se *changeset*
- Svaki skup promjena dobiva jedinstveni identifikator
 - Kod centraliziranih sustava identifikator je redni broj koji se povećava za 1
- *Trunk* – jedinstvena linija razvoja koja nije ničija grana



Konfliktne promjene

14

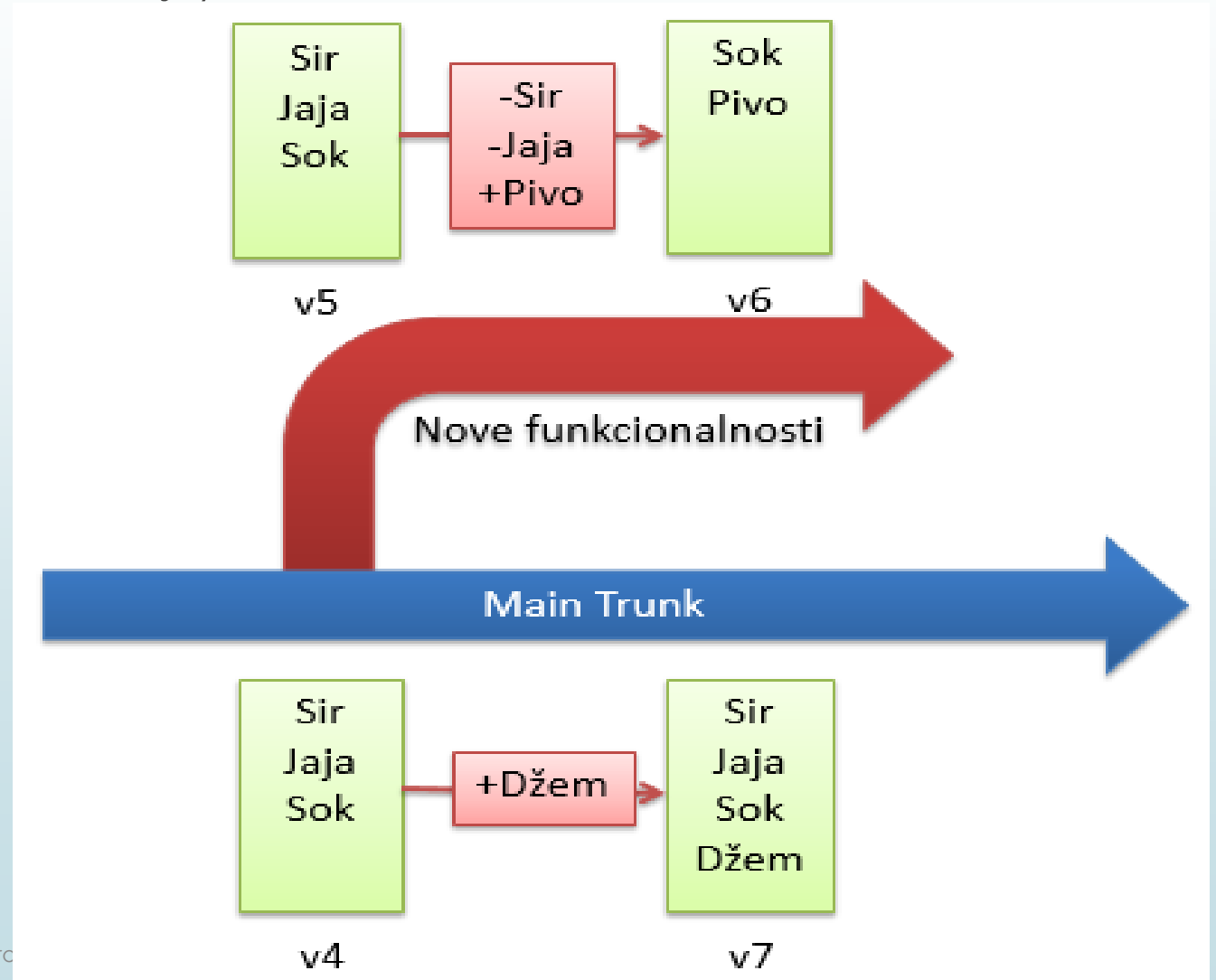
- Istovremene promjene mogu voditi do konflikta
 - Prvi korisnik će uspješno potvrdi promjene
 - Drugom korisniku će sustav dojaviti da su promjene napravljene u odnosu na verziju koja više nije najsvježija
- Rješenja
 - Odbaciti vlastitu verziju
 - Ignorirati verziju sa servera
 - Automatsko ili ručno rješavanje preklapanja



Paralelne grane

15

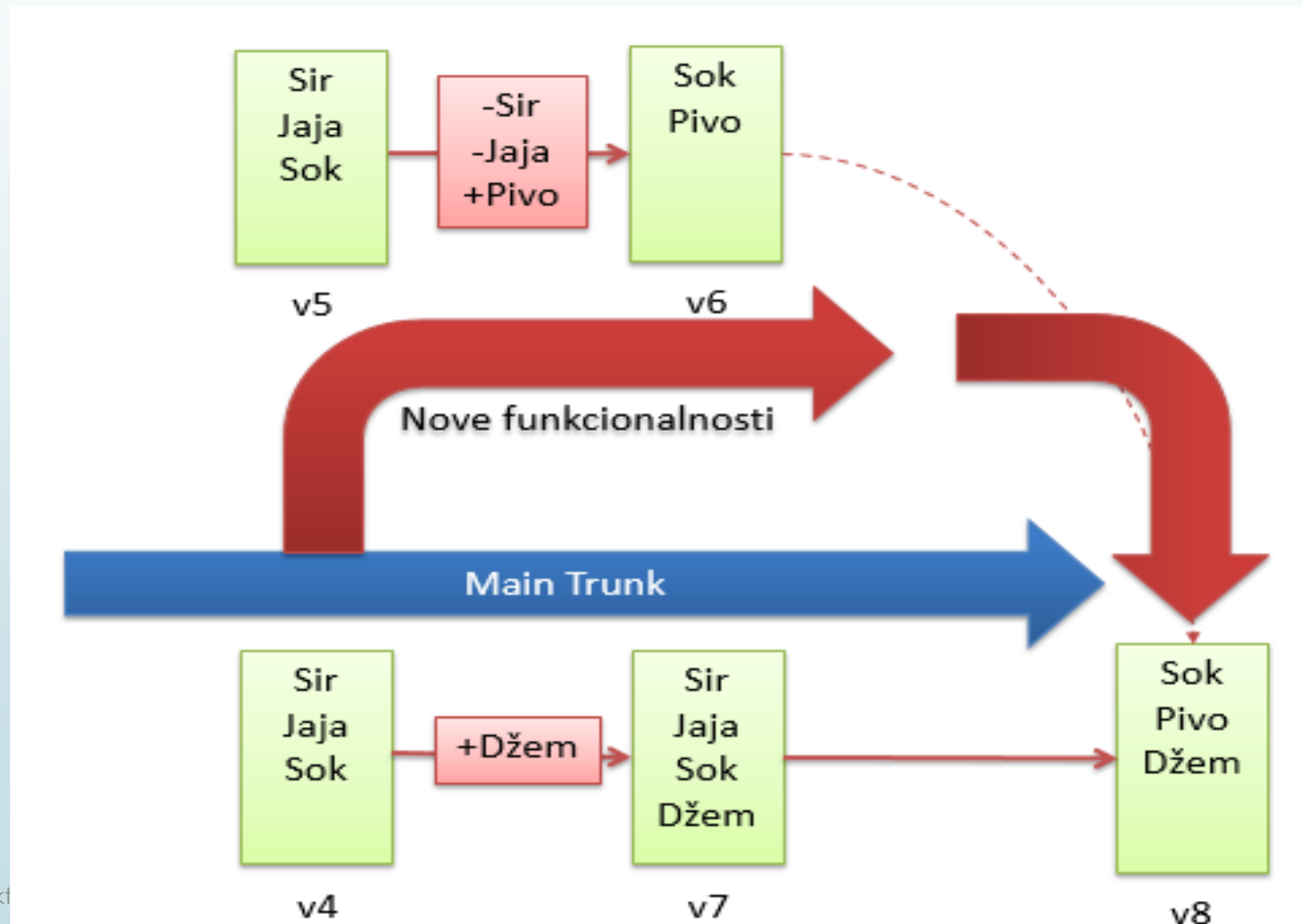
- Omogućava istovremeni rad na novim funkcionalnostima uz očuvanje ispravnosti i održavanje glavne (isporučene verzije).
- Ovisno o sustavu može se raditi o fizički odvojenim kopijama datoteka



Spajanje grana

16

- Nije nužno – paralelna grana ne mora se nikad spojiti natrag
- Ugradnja promjena iz paralelne grane s promjenama u međuvremenu obavljenim na glavnoj grani
 - paralelna grana se može, ali i ne mora obrisati
- Git dodatno omogućava reproduciranje samo dijela promjena iz paralelne grane (cherry-pick)

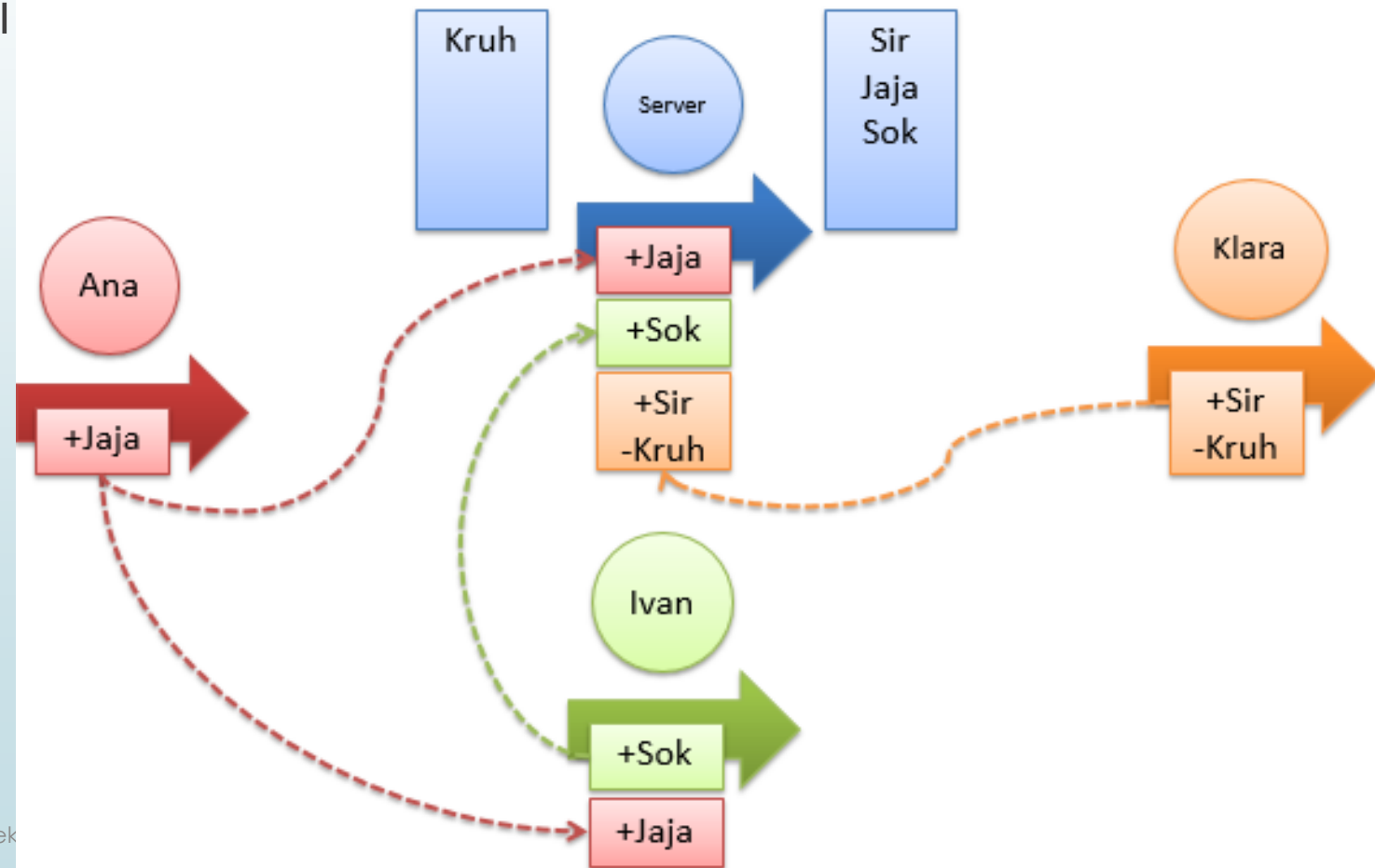


- Git, Mercurial
- Svaki korisnik ima kompletnu kopiju repozitorija
 - sadržaj verzioniranih datoteka, ali i cijela povijest promjena
 - lakše kopiranje repozitorija na neko drugo mjesto
- Korisnik može evidentirati promjene u izvanmrežnom načinu rada
 - naknadno prenosi svoju povijest na centralni server (push)
 - osvježava svoju verziju s verzijom na serveru (fetch i/ili pull)
 - *pull = fetch + merge*
- Inicijalno stvaranje kopije centralnog repozitorija naziva se kloniranje (engl. clone)
- Napomena: centralni repozitorij ne mora postojati!
- Ostali koncepti akcija i problema (konflikata) su isti ili slični kao kod centraliziranih sustava.

Pojava konflikta u distribuiranim sustavima

18

- Kod distribuirani sustav češće dolazi do konflikta
 - Različiti sadržaji repozitorija kod pojedinih korisnika + neovisne promjene
 - Potrebno spajanje promjena
- Svaka promjena mora imati jedinstveni identifikator
- Redni broj nije dovoljan
 - potrebna složenija oznaka (jedinstveni kod, npr. hash)



Neki od osnovnih pojmova

19

- Kloniranje centralnog repozitorij ili inicijalizacija samostalnog

- `git clone adresa_repozitorija`
- `git init`

- Dohvat zadnje verzije centralnog repozitorija

- `git pull origin naziv grane`
- origin označava udaljeni repozitorij, naziv glavne grane je master

- Datoteke čije se promjene žele evidentirati dodaju se u međuspremnik (Staging Area)

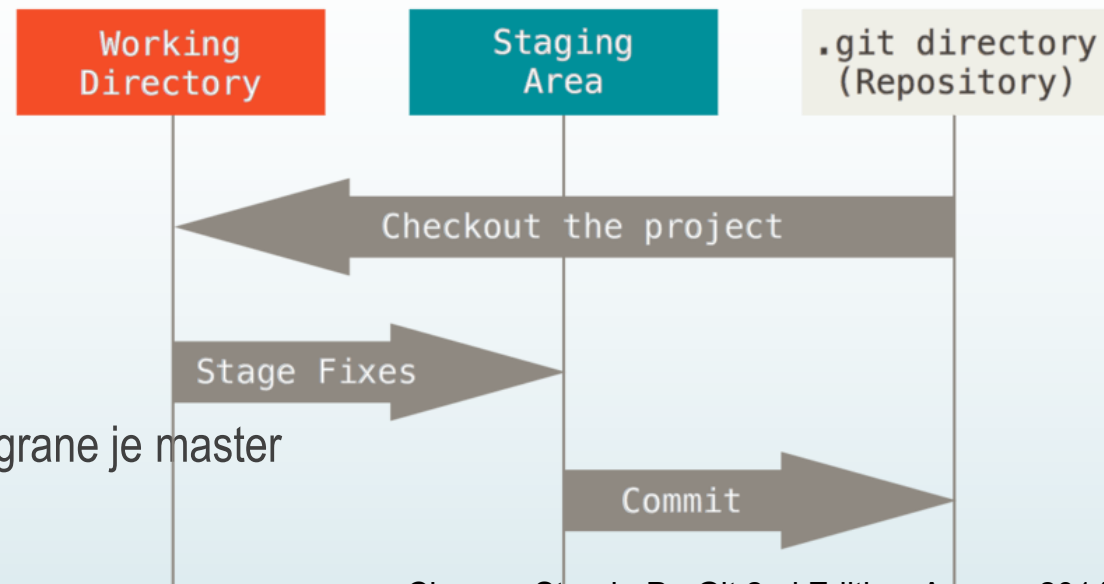
- `git add naziv datoteke ili putanja do više datoteka`

- Korisnik

- potvrđuje `git commit -m "opis promjena"`
- Ili odbacuje `git checkout naziv datoteke ili putanja`

- Korisnik šalje svoje promjene na server

- `git push origin naziv grane`



Chacon, Straub, ProGit 2nd Edition, Apress, 2014.

Neki od ostalih pojmova i literatura

20

➤ Stash i Unstash

- Privremeno spremanje izmjena, ali bez potvrđivanja promjena (tj. bez commita)
- Slično kao Shelve/Unshelve kod TFS-a

➤ Pull request/Merge request

- Zahtjev nekog korisnika za pregledom njegovog koda i spajanje s nekom drugom (najčešće glavnom) granom

➤ Literatura za Git

- <http://tkrajina.github.io/uvod-u-git/git.pdf>
- <https://git-scm.com/book/en/v2/>
- <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>

- Prikuplja relevantne objekte i proizvodi određenu verziju softvera
 - Dohvat određene verzije iz repozitorija
 - Kompilacija i povezivanje biblioteka
 - Pokretanje testova (ako postoje)
 - Objava verzije
 - Interno
 - Spremno za instalaciju kod korisnika
 - Automatska isporuka korisnicima
- Pokretanje može biti
 - na zahtjev
 - dnevno (npr. nightly build)
 - nakon svake promjene koda u repozitoriju
 - Ili nakon nekoliko akumuliranih promjena
 - ...

Primjer kontinuirane isporuke korištenjem alata TFS (1)

22

➡ Definira se niz koraka koji služe za automatsku izgradnju i isporuku

The screenshot displays the TFS Build interface for defining a build step. The main window is titled "Definitions / RPPP01 | Builds" and includes tabs for "Build", "Options", "Repository", "Variables", "Triggers", "General", "Retention", and "History". The "Repository" tab is active, showing settings for "Repository type" (Git), "Repository" (01), "Default branch" (master), "Clean" (true), "Label sources" (Don't label sources), and "Checkout submodules" (unchecked).

On the left, a list of build steps is shown, with "Command Line" selected. The selected step is configured with the following details:

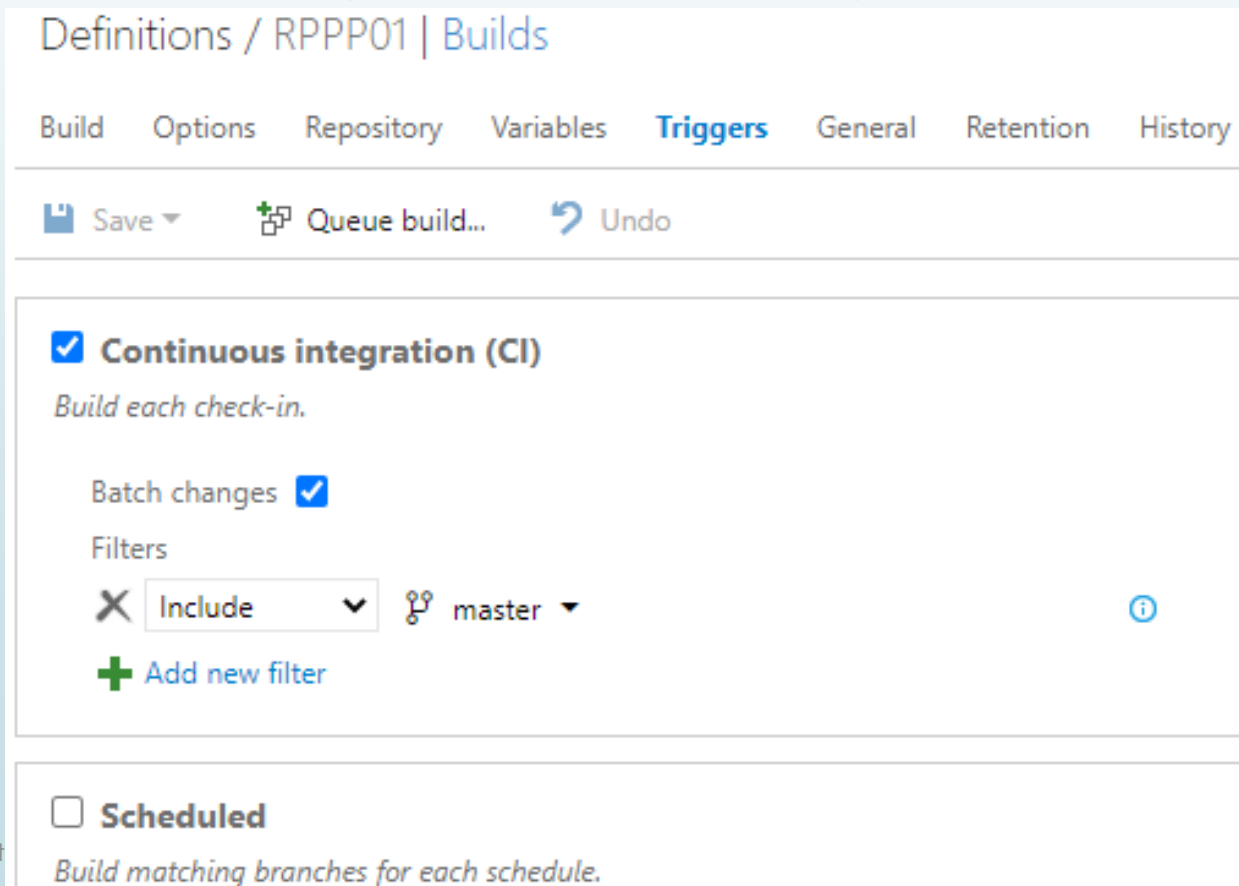
- Run dotnet publish** (with an edit icon)
- Tool:** dotnet
- Arguments:** publish -c=\$(BuildConfiguration) -o=\$(build.stagingDirectory)\\$(build.BuildNumber)
- Advanced** (expanded)
- Control Options:**
 - Enabled: ☒
 - Continue on error: ☐
 - Always run: ☐

Below the build steps list, other options like "Copy and Publish Build Artifacts" and "PowerShell" are visible.

Primjer kontinuirane isporuke korištenjem alata TFS (2)

23

- Izgradnja se pokreće nakon svake promjene u repozitoriju
 - Continuous integration (CI)
 - Iako bez testova to je više Continuous build nego integration
- U definiciji s prethodnog slajda uključena i isporuka (objava web-aplikacije)
 - Continuous delivery (CD)



The screenshot shows the 'Builds' configuration page for a build definition named 'RPPP01'. The 'Triggers' tab is selected, showing the 'Continuous integration (CI)' trigger is enabled with the description 'Build each check-in.'. Below this, the 'Batch changes' checkbox is checked. Under the 'Filters' section, there is a dropdown menu set to 'Include' and a branch selector set to 'master'. An 'Add new filter' button is visible. The 'Scheduled' trigger is disabled.

Definitions / RPPP01 | Builds

Build Options Repository Variables Triggers General Retention History

Save Queue build... Undo

☒ **Continuous integration (CI)**
Build each check-in.

Batch changes ☒

Filters

☒ Include

+ Add new filter

☐ **Scheduled**
Build matching branches for each schedule.

Alati za kontinuiranu integraciju i isporuku (1)

24

- BitBucket Pipeline, Azure Pipelines, Semaphore CI, GitHub Actions ...
- Jenkins, Travis, ...

Pipelines

All branches





















OTHER BRANCHES

master **MAIN BRANCH**

big-refactor

fix/frontend


Pipeline


#82		minor fix Ivan Juren  2bd1c2c  master		Successful	8 months ago	4 min 48 sec
#81		Merged in add-reviews (pull request #15) Add reviews Ivan Juren  0791112  master		Successful	8 months ago	5 min 24 sec
#78		Merged in feature/user_cancel_trip (pull request #14) add cancel ... Petra Zavrski  d3f7838  master		Successful	9 months ago	5 min 6 sec
#77		Merged in feature/user_join_trip (pull request #13) Feature/user j... Petra Zavrski  f168df3  master		Successful	9 months ago	5 min 45 sec
#73		create trip fix Ivan Juren  1896de2  master		Successful	9 months ago	5 min 9 sec


Boris Milašinović / ... / Pipelines

✓ #82

Rerun

 2bd1c2c minor fix

 master

 [Learn more about reports](#)

 4 min 48 sec  8 months ago



Pipeline



✓ Build and test

32s

✓ Deploy Backend to prod... 4m 14s

Redeploy

Build

Build setup

```
apt-get update -qy
```

```
apt-get install -y ruby-dev
```

```
gem install dpl
```

```
dpl --provider=heroku --app=$HEROKU_APP_NAME --api-key=$HEROKU_API_KEY
```

```
cd frontend
```

```
dpl --provider=heroku --app=$HEROKU_FRONTEND_APP_NAME --api-key=$HEROKU_API_KEY
```