

CI/CD

Prof. dr Miloš Milić

Sadržaj

- Uvod
- Git
- CI/CD
- GitHub Actions
- Raspoređivanje softvera

Uvod

- Složenost procesa razvoja softvera
- Praćenje i upravljanje verzijama softvera
- Sistemi za upravljanje:
 - Version Control System (VCS)
 - Source Code Management (SCM)
 - Revision Control System (RCS)

Uvod

- Prednosti korišćenja VCS alata
- Podela
 - Centralizovani VCS
 - Distribuirani VCS

Git

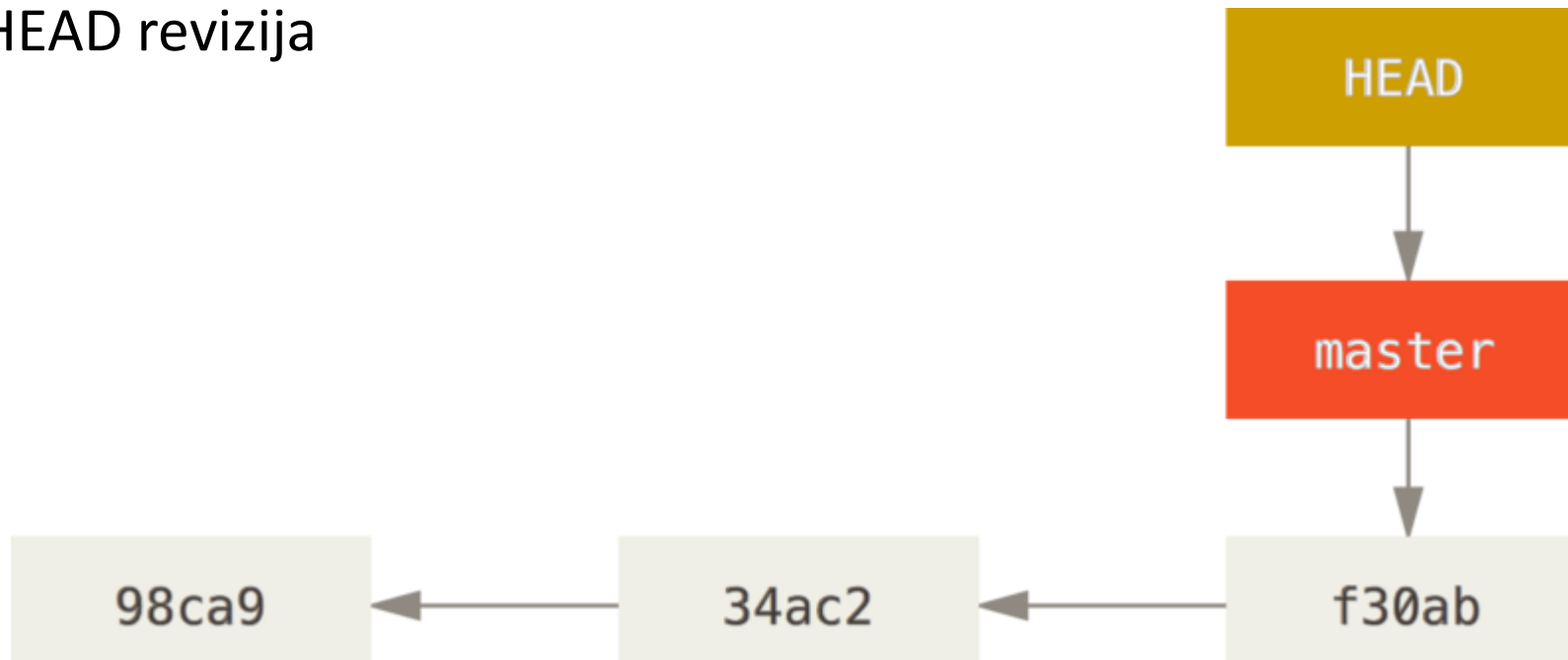
- Open Source: <https://git-scm.com>
- Podrжан od strane velikog broja alata i kompanija
- Platforme: GitHub, GitLab, BitBucket

Git

- Instalacija
 - Git
 - SourceTree
 - Razvojno okruženje ima ugrađenu podršku
 - .gitconfig

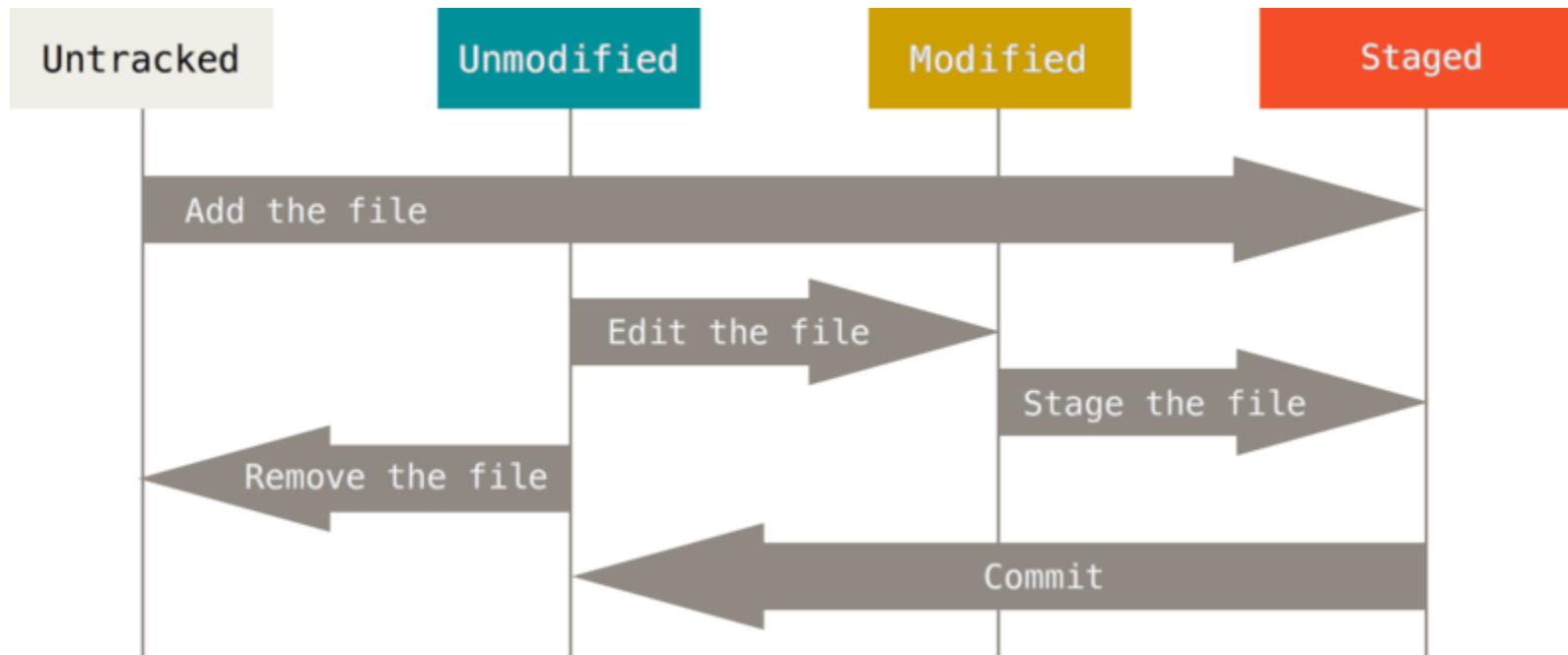
Git

- Repozitorijum (engl. Repository)
- Radna kopija/direktorijum (engl. Working Copy/Directory)
- Revizija (engl. Revision, Commit)
 - HEAD revizija



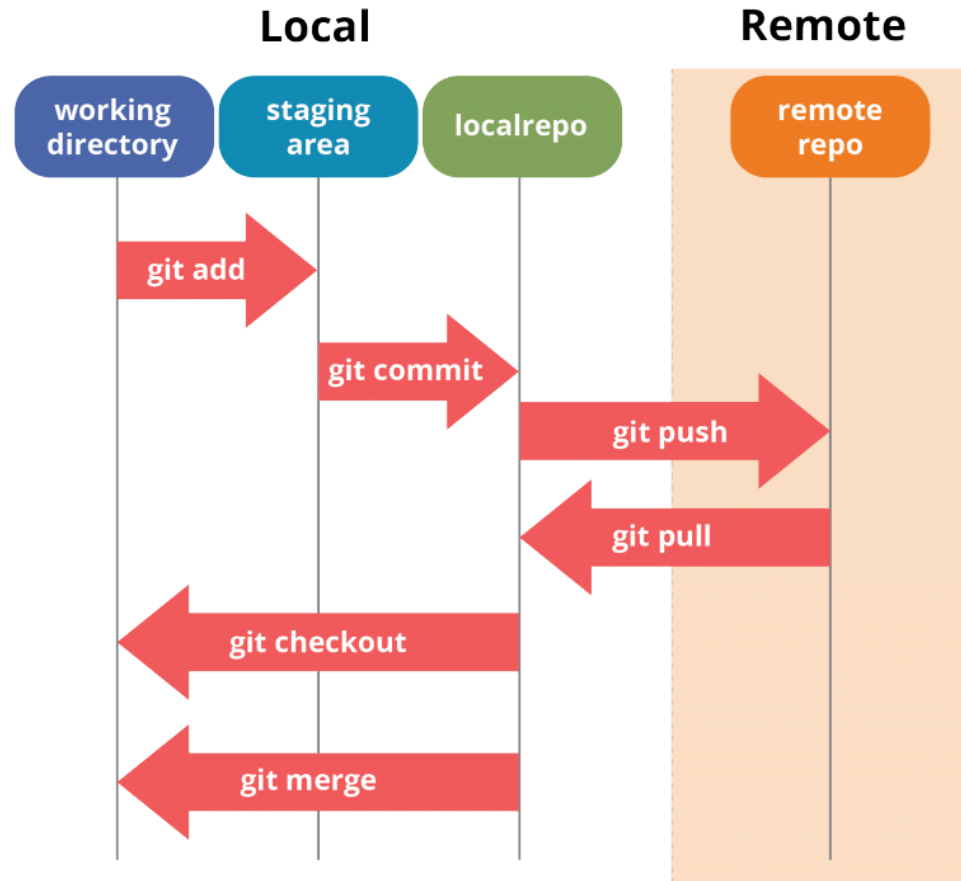
Git

- Statusi datoteka



Git

- Komande

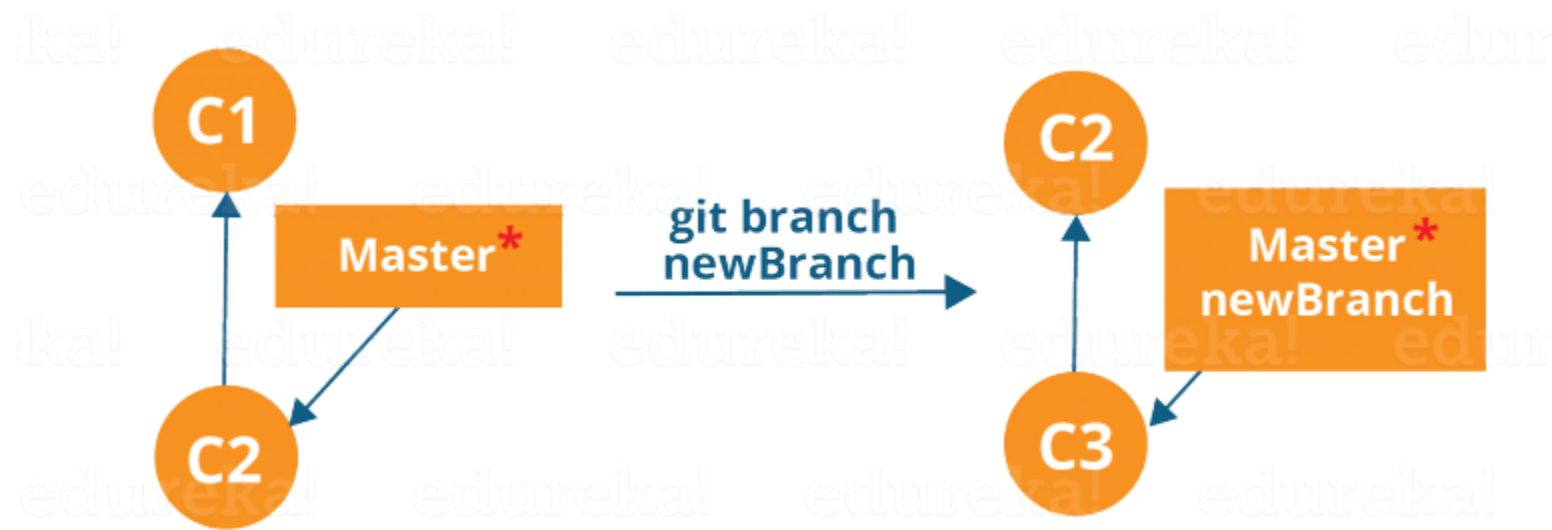


Git

- Logovi
- Upoređivanje verzija

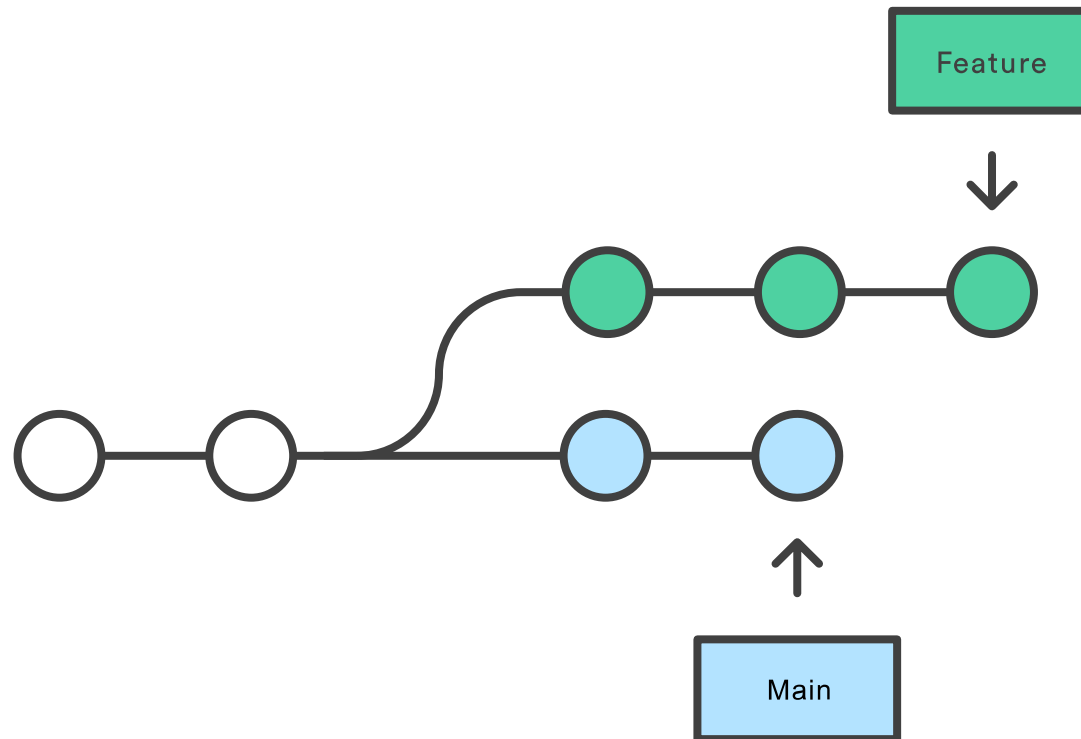
Git

- Branching



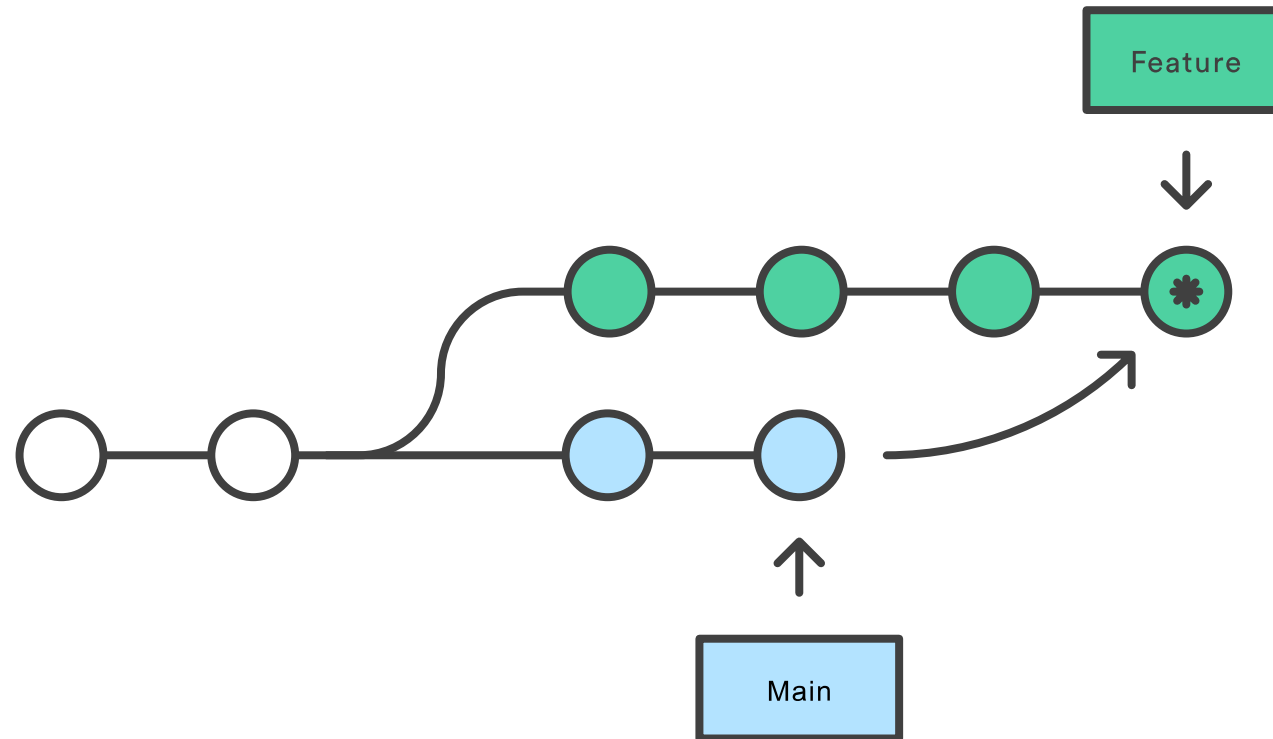
Git

- Merge vs Rebase
- Postojeća struktura



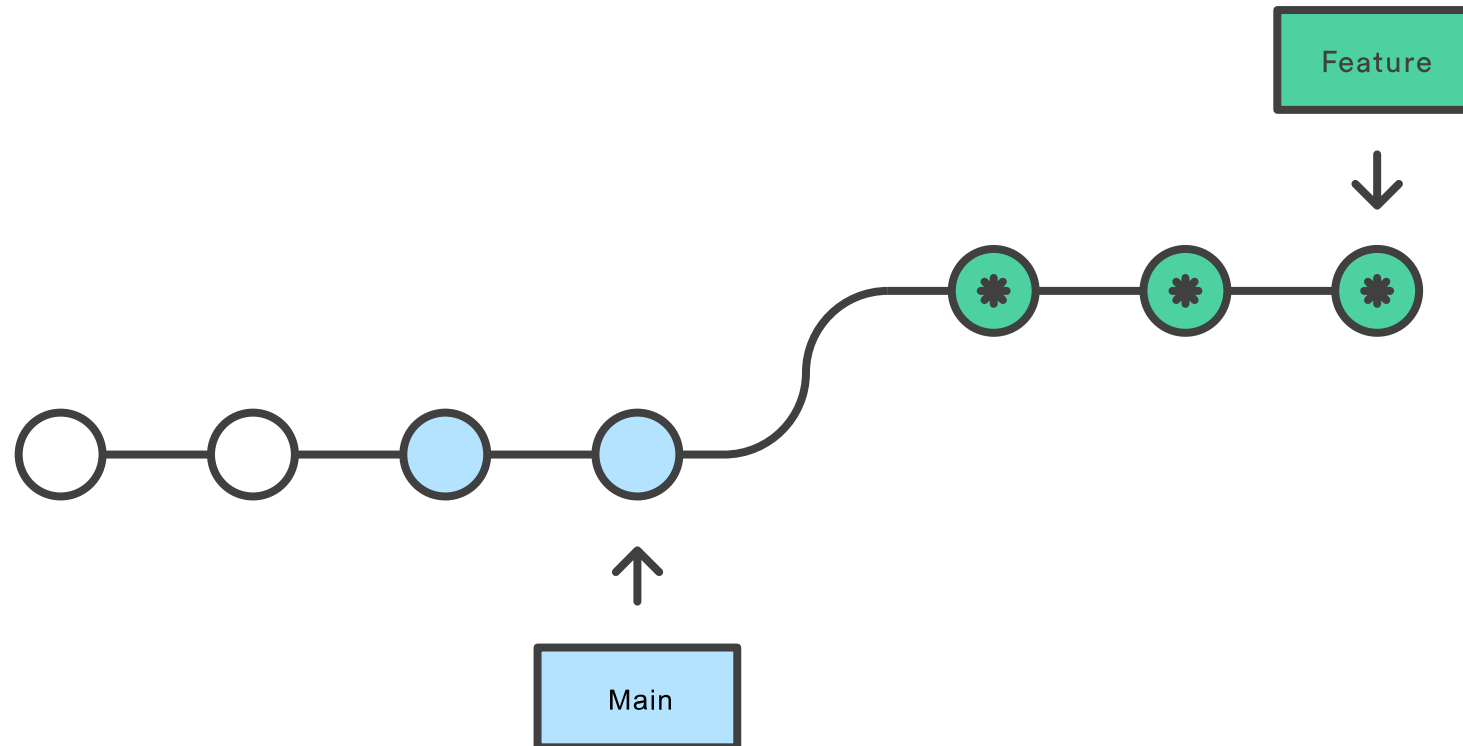
Git

- Merge
- Prednosti i nedostaci



Git

- Rebase
- Prednosti i nedostaci



Git

- Pravilo: ne raditi rebase na *public* granama

Zadatak 1

- Podešavanje projekta i repozitorijuma
 - Struktura projekta
 - Repozitorijum
 - Inicijalni commit

Zadatak 2

- Event (id, name, budget, startDate, endDate, description)

Zadatak 3

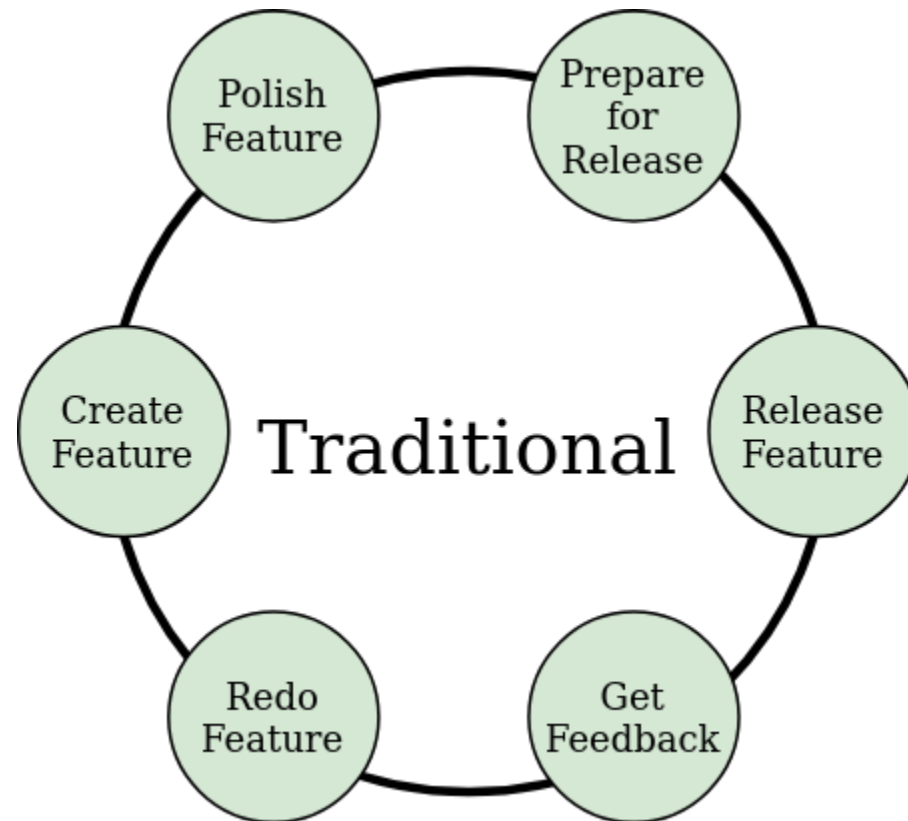
- Event Management Feature

CI/CD

- Softverski proces
- Softverski sistem kao proizvod
- Problemi u procesu razvoja

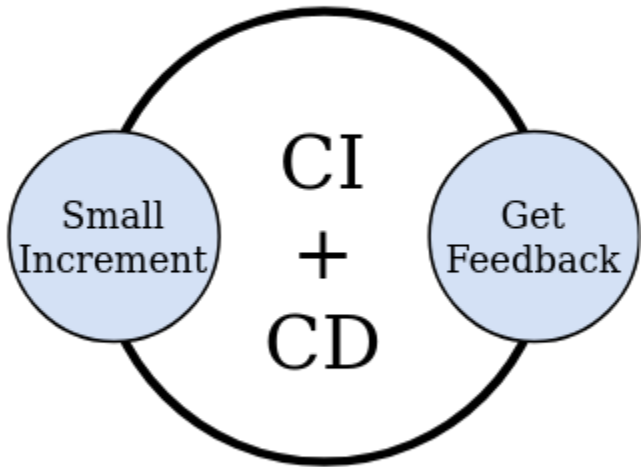
CI/CD

- Tradicionalni pristup



CI/CD

- Savremeni pristup
- Iterativno-inkrementalni model



CI/CD

- Continuous Integration – CI
 - Integracija artefakata koje kreiraju softverski inženjeri u programski kod (engl. mainline), što proizvodi automatsku izgradnju i testiranje softvera
- Continuous Delivery – CD
 - Izrada softvera u kratkim ciklusima
 - Softver se može pouzdano isporučiti
 - Production-like Environment [Chen, 2015]
- Continuous Deployment – CD
 - Izrada softvera u kratkim ciklusima
 - Softver se isporučuje frekventno
 - Automatska isporuka softvera i u produkciji [Chen, 2015]

CI/CD

- Izvršavanje nekog *radnog toka* (engl. Workflow) u slučaju izvršenja nekog *dogadjaja*
- Tipično, događaj se dešava u okviru repozitorijuma:
 - Commit
 - Pull Request
 - Nastao je neki problem i sl.
- Unutar radnog toka može postojati više elemenata koje treba izvršiti:
 - Izvršavanje u sekvencijalnom redosledu
 - Paralelno izvršavanje

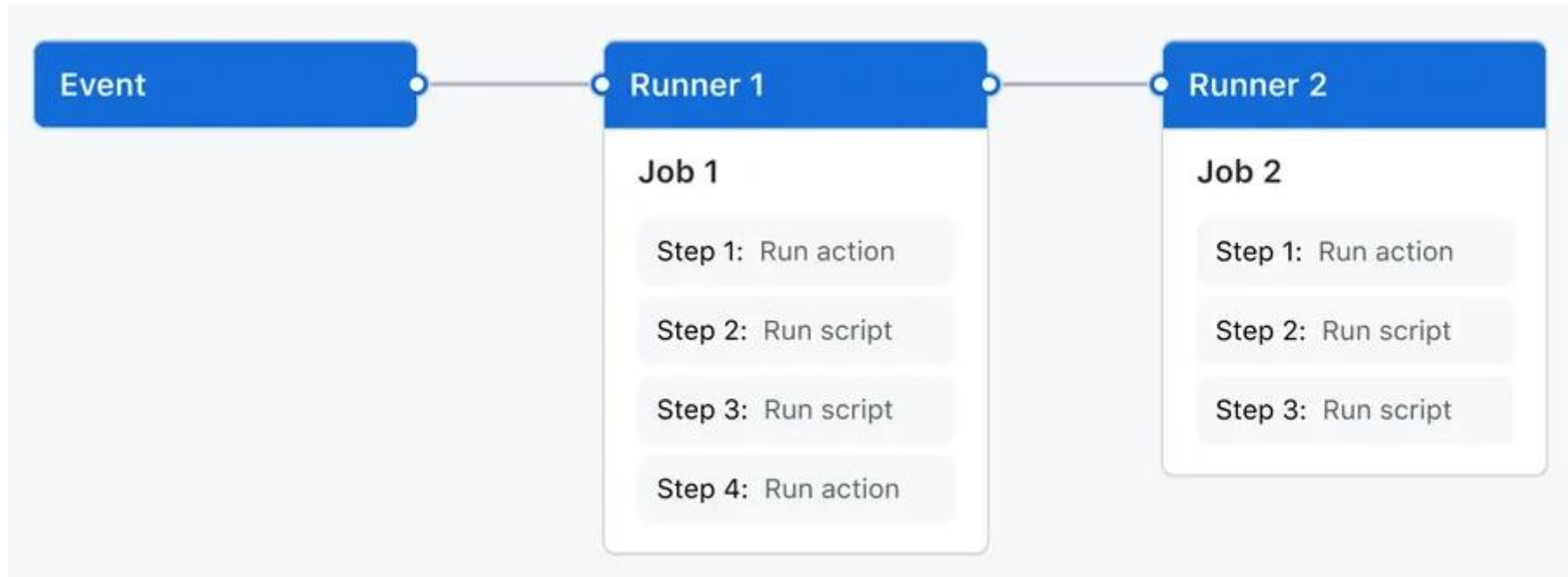
CI/CD

- Unutar radnog toka (*workflow*) može postojati više poslova (*job*)
- Svaki *job* se izvršava unutar svoje virtualne mašine ili unutar kontejnera
- Svaki *job* se sastoji od jednog ili više koraka (*step*)
 - Korak izvršava skriptu ili može da pozove neku drugu akciju
 - Može se ostvariti ponovno korišćenje (engl. Reusability)
- Koncepti su slični, implementacija se može razlikovati i zavisi od konkretne platforme

GitHub Actions

- GitHub platforma daje mogućnost korišćenja GitHub Actions akcija koje omogućavaju CI/CD
- Radni tok (*workflow*) predstavlja automatizovani proces unutar koga se izvršava jedan ili više *job*-ova
- U okviru repozitorijuma može postojati više radnih tokova
 - Folder `.github/workflows`
- Radni tok se definiše kroz *YAML* konfiguracioni fajl

GitHub Actions



GitHub Actions

- Podrazumevano, radni tokovi se automatski pokreću nakon nastanka nekog događaja
- Međutim, moguće je i njihovo ručno pokretanje
 - U okviru samog repozitorijuma
 - Putem GitHub REST API poziva

GitHub Actions

- *Job* – predstavlja skup koraka (*steps*) koji se izvršavaju u okviru istog *runner* okruženja
- Koraci se odvijaju definisanim redosledom
- Koraci su međusobno zavisni
- Uzimajući u obzir da se koraci izvršavaju unutar istog *runner* okruženja – postoji mogućnost deljenja podataka
 - Primer
- *Job* može biti nezavisan ili može zavisiti od drugih *job*-ova

GitHub Actions

- Podrazumevno su nezavisni -> paralelno izvršavanje
- U slučaju zavisnosti čeka se da „nadređeni“ *job* završi svoje izvršavanje
 - Primer

GitHub Actions

- *Actions* – specijalizovana aplikacija za GitHub Actions platformu koja izvršava kompleksan i repetitivan zadatak:
 - Preuzimanje podataka iz GitHub repozitorijuma
 - Podešavanje programa i alata (engl. Toolchain) okruženja za izgradnju
 - Podešavanje konekcije sa Cloud provajderom itd.
- Moguće je pisanje sopstvenih akcija
- Postojeće akcije se mogu naći u okviru GitHub Marketplace
- Reusability

GitHub Actions

steps:

- name: Set up Java version

```
uses: actions/setup-java@v4
```

```
with:
```

```
  java-version: '11'
```

GitHub Actions

- *Runner* – predstavlja okruženje unutar koga se izvršava *workflow*
- Može izvršavati jedan *job* u datom trenutku
- Svaki *workflow* se izvršava u sopstvenoj virtualnoj mašini
- Podrazumevano se mogu koristiti Ubuntu Linux, Microsoft Windows i macOS *runner* okruženja
- U slučaju potrebe za specifičnim operativnim sistemom i/ili specifičnom hardverskom konfiguracijom moguće je hostovati sopstveni *runner* (engl. self-hosted runners)

GitHub Actions

jobs:

Test:

runs-on: ubuntu-latest

CI/CD

Prof. dr Miloš Milić