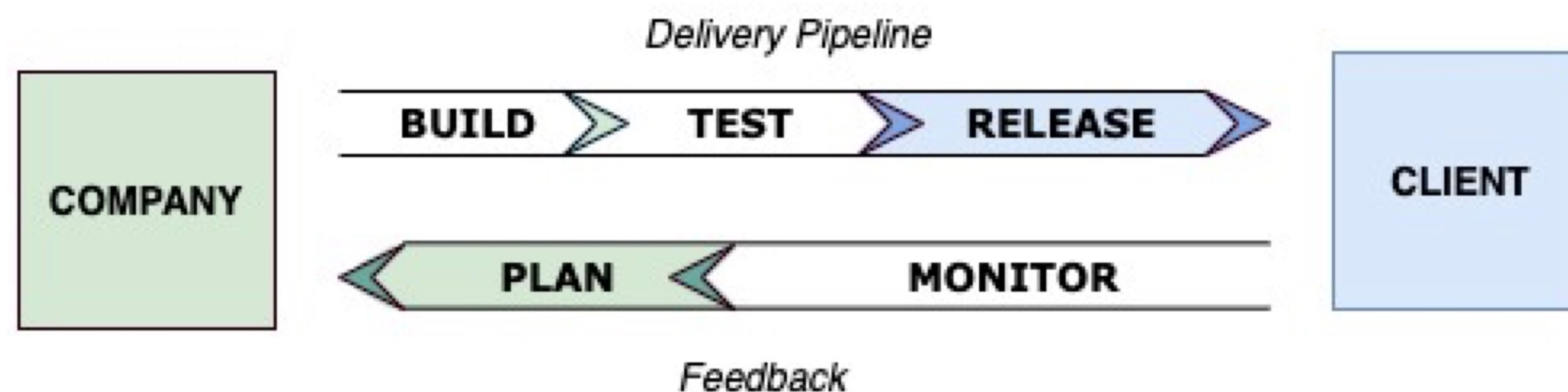


# UVOD U DEVOPS

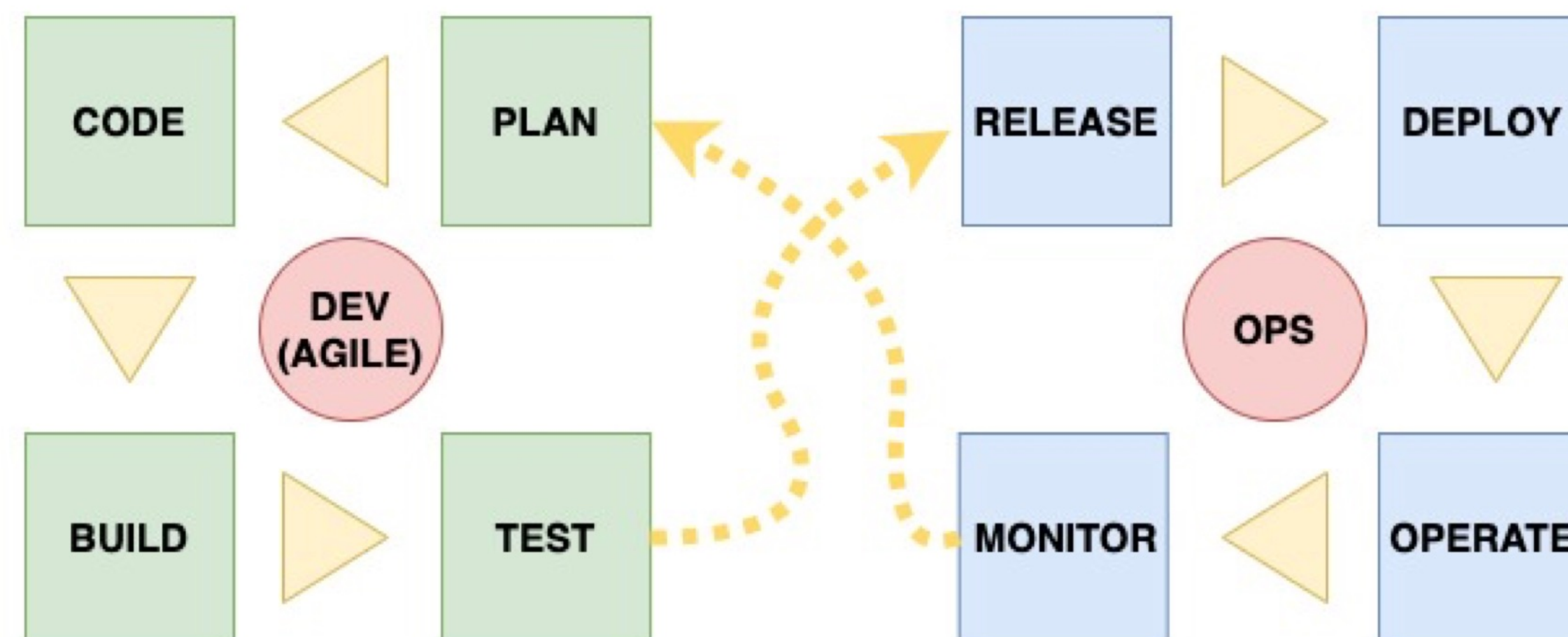
## ◆ ŠTA JE DEVOPS?

- Development and Operations
- Princip u modernoj metodologiji razvoja softvera koji se zasniva na puštanju u rad manjih promena na aplikaciji kako bi se što više smanjio uticaj grešaka na krajnje korisnike, za razliku od velikih *deployment*-a koji prave dosta promena koje mogu da dovedu do velikih problema



## ◆ ŠTA JE DEVOPS?

- Kombinuje različite alate i prakse kako bi kompanija bila u mogućnosti da isporučuje softverska rešenja brže u odnosu na tradicionalni model isporuke softvera
- Prema DevOps modelu, razvojni i operativni timovi sarađuju tokom čitavog životnog ciklusa softvera, od razvoja i testiranja preko *deployment*-a i puštanja u rad





# TRADICIONALNI MODEL ISPORUKE VS DEVOPS

4

## ◆ KOJE SU RAZLIKE?

- Kod tradicionalnog modela, razvojni tim piše i testira kod u svom privatnom okruženju, bez da je potpuno svestan praksi koje operativni timovi praktikuju
- Klijentu se brže isporučuju rešenja, brži odziv na zahteve
- Automatizacijom procesa smanjuje se prostor za greške

**ODVOJENI RAZVOJNI I  
OPERATIVNI TIMOVI**

**INTEGRACIJA RAZVOJNIH I  
OPERATIVNIH TIMOVA**

**ISPORUKA U FAZAMA**

**KONTINUALNA ISPORUKA**

**MANUALNI PROCESI**

**AUTOMATIZACIJA PROCESA**



# PREDNOSTI

5

## ◆ **BRZA ISPORUKA**

- Povećanjem frekvencije isporuke manjih celina, proizvod se brže razvija
- Brži odziv na zahteve klijenata i ispravku grešaka
- Korisnici brže dobijaju nove funkcionalnosti i ispravke grešaka
- Prilagođavanje promenama na tržištu

## ◆ **POUZDANOST**

- Kontinualnom integracijom i isporukom novih rešenja povećava se verovatnoća da je svaka nova promena funkcionalna i bezbedna za upotrebu





# PREDNOSTI

6

## ◆ UNAPREĐENJE SARADNJE

- Razvojni i operativni timovi sarađuju i dele odgovornosti
- Ušteda vremena

## ◆ BEZBEDNOST

- Bezbednost treba da bude uključena u svaku fazu DevOps životnog ciklusa, uključujući faze dizajna, razvoja, testiranja, *deployment*-a, održavanja,...
- Danas se ovo popularno naziva DevSecOps koji ima za cilj unapređenje bezbednosnih aspekata kroz čitav DevOps workflow



# CI VS CD

7

## ◆ **CONTINUOUS INTEGRATION (CI)**

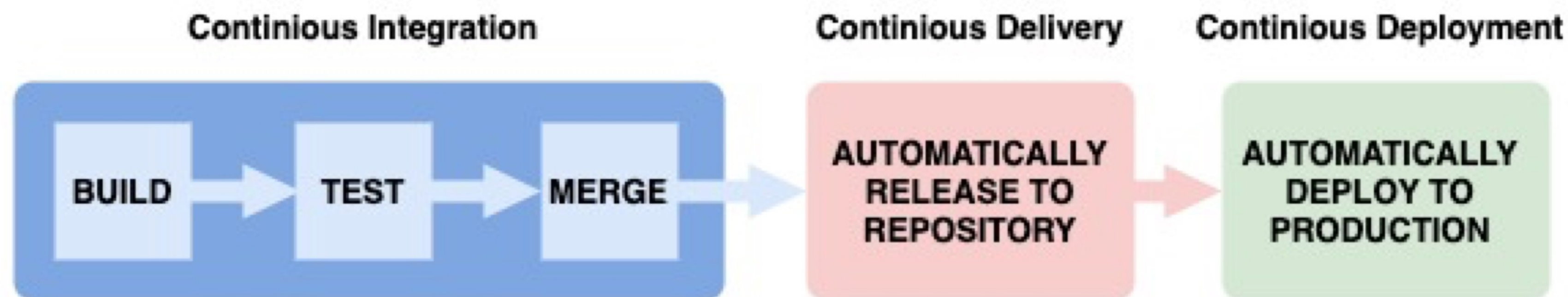
- Praksa razvoja softvera gde se praktikuje redovno spajanje izmena koda na centralni repozitorijum, nakon čega se pokreću automatizovane *build* skripte i testovi
- Cilj je brže pronalaženje i rešavanje grešaka, poboljšanje kvaliteta softvera i smanjenje vremena potrebnog za validaciju i *release* nove verzije

## ◆ **CONTINUOUS DELIVERY (CD)**

- Praksa gde se izmene koda automatski *build*-uju, testiraju i pripremaju za puštanje u produkciju
- Proširuje CI praksu tako što se sve izmene direktno puštaju na testno i/ili produkciono okruženje nakon *build* faze

## ◆ ŠTA JE CI/CD PIPELINE?

- Niz koraka koji se moraju izvršiti da bi se isporučila nova verzija softvera
- Uvodi automatizaciju za unapređenje procesa razvoja aplikacija, posebno u fazama integracije, testiranja i isporuke



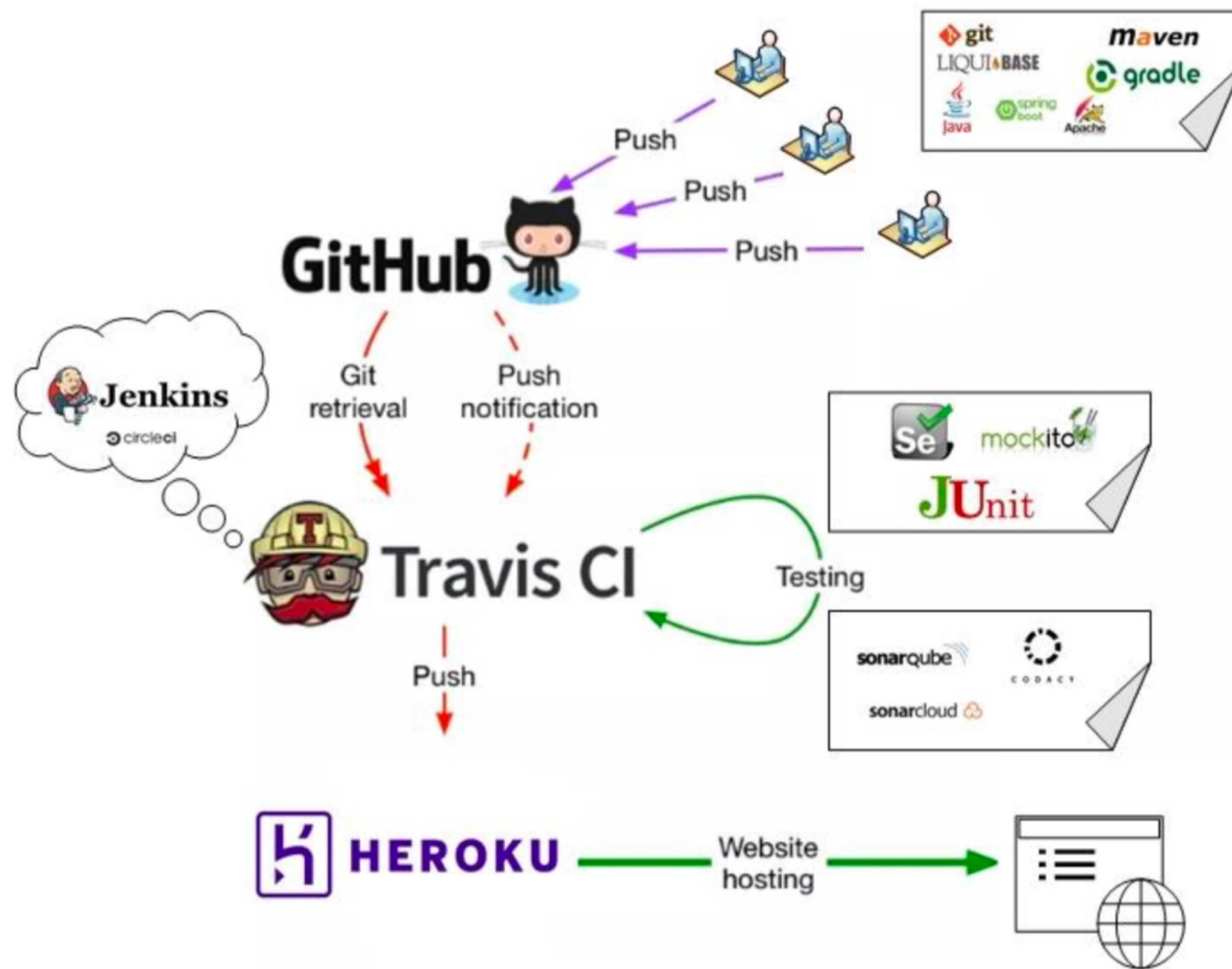




# ALATI

<b>Source Code control</b>	<b>CI/CD</b>	<b>Containers</b>
GitHub GitLab Bitbucket	Travis Jenkins Circle CI	Docker
<b>Cloud computing and storage</b>	<b>Monitoring, alerting and incident response tools</b>	<b>IT ticketing</b>
AWS Azure Google Cloud	New Relic Datadog Instana	Jira Trello HubSpot
<b>Communication and collaboration</b>	<b>Documentation</b>	<b>Testing</b>
Slack Microsoft Teams Rocket Chat	Confluence Slite	Selenium Gremlin

# PRIMER





## ◆ ŠTA JE TRAVIS CI?

- Continuous Integration platforma
- Kod se *build*-uje i testira na potpuno novom, neutralnom okruženju
- Informacija da li je ciklus uspešno završen

## ◆ ALTERNATIVE

- Jenkins
- CircleCI
- Semaphore CI
- ...



## ◆ ŠTA JE SONARQUBE?

- Platforma koja omogućava kontinualnu kontrolu kvaliteta koda
- Statička analiza koda za otkrivanje grešaka u kodu, *code smells* i bezbednosnih propusta

## ◆ ZAŠTO JE STATIČKA ANALIZA KODA VAŽNA?

- Analiza koda bez njegovog izvršavanja
- Detekcija problema u ranim fazama razvoja softvera
- Dodavanje novih pravila kako bi se analiza prilagodila specifičnostima softvera

## ◆ ALTERNATIVE

- Covertly
- Codacy
- ...





## ◆ ŠTA JE HEROKU?

- Cloud rešenje – *Platform as a Service* (PaaS)
- Pruža mogućnost *build*-ovanja, isporuke, nadgledanja i skaliranja aplikacija
- Razni dodaci (Data stores, Caching, Message Queues,...)

## ◆ ALTERNATIVE

- Amazon Web Services
- Azure
- Google Cloud Platform
- DigitalOcean
- ...





# REFERENCE

14

- ◆ **PRIMER PO UZORU NA** <https://github.com/stojkovm/isa-deployment-test>
- ◆ **WHAT IS DEVOPS?** <https://aws.amazon.com/devops/what-is-devops/>
- ◆ **WHAT IS DEVOPS?** <https://www.crestdatasys.com/services/devops-services/>
- ◆ **WHAT IS CI/CD PIPELINE?** <https://www.redhat.com/en/topics/devops/what-cicd-pipeline>
- ◆ **TRAVIS CI** <https://docs.travis-ci.com/user/for-beginners/>
- ◆ **SONARQUBE** <https://docs.sonarqube.org/latest/>
- ◆ **HEROKU** <https://devcenter.heroku.com/categories/reference>

**KOJA SU VAŠA  
PITANJA?**