



Razvoj elektroenergetskog softvera



Primenjeno softversko inženjerstvo



Uvod

Izvođači nastave

- Aleksandar Selakov – aselakov@uns.ac.rs
- Stefan Dejanović – stefan.dejanovic@uns.ac.rs
- Zoran Janković – zoran.jankovic@uns.ac.rs
- Zorana Babić – zbabic@uns.ac.rs

- Materijali i obaveštenja su dostupni na Teams platformi i sajtu:
 - <http://www.esi.ftn.uns.ac.rs/index.php/predmeti/77-kvalitet>

Cilj vežbi

- Sticanje osnovnih znanja o:
 - Timskom radu
 - Kontrolni revizija
 - Statičkoj analizi koda
 - Stilovima pisanja koda
 - Pisanju unit testova
 - Principu automatizacije
 - Dizajnu i arhitekturi softvera
 - Agilnoj metodologiji razvoja softvera

Polaganje predmeta

- Teorijski deo – 20 poena
- Praktični deo – 80 poena
 - Timski projekat
 - Projekat se sastoji od faza:
 - Arhitektura i dizajn rešenja
 - Planiranje i estimiranje (Scrum)
 - Implementacija rešenja
 - Unit testovi i pokrivenost koda
 - Prisustvo na vežbama je obavezno i ulazi u ukupan broj poena iz praktičnog dela
 - 2 termina odbrane projekta
 - Jun
 - Septembar



Revision control

Kontrola revizije

- **Kontrola revizije** (eng. *Revision control (RC), Version control (VC)*) je proces upravljanja, praćenja i kontrole izmena i verzija podataka (dokumenata, fajlova, ...)
- Svaka izmena se naziva **revizijom** (eng. *revision*) i identifikovana je nekim ID-jem
- U softverskom inženjeringu, kontrola revizije se najčešće vrši nad izvornim kodom
- Alati ili aplikacije pomoću kojih se upravlja verzijama koda, i pomoću kojih se izmene u kodu prate i kontrolišu su **Sistemi za kontrolu revizija** (eng. *Revision Control Systems, RCS*)

Sistemi za kontrolu revizije

- **Sistem za kontrolu revizija** je softverska implementacija **RC**-a koja omogućava i automatizuje proces nadzora, identifikovanja, čuvanja, spajanja i primenjivanja izmena u fajlovima
- Zašto koristiti RC sisteme?
 - Anotacija – beleže se korisnici koji izvršavaju izmene, svaka izmena ima svoj ID i svaka izmena može imati prateći komentar
 - Reverzibilnost – mogućnost jednostavnog vraćanja na neko prethodno stanje, tj. vraćanje na prethodne verzije pre određenih izmena
 - Konkurentnost – mogućnost višekorisničkog rada na istim fajlovima ili mogućnost da svaki korisnik radi na svojim kopijama fajlova

Terminologija RC sistema

- **Repository** (repozitorijum/repo) – lokacija ili folder gde se nalaze interni metapodaci potrebni za rad sistema, sačuvana istorija izmena fajlova, kao i fajlovi izvornog koda (opciono)
- **Working copy** – lokalna radna kopija fajlova preuzeta sa nekog repozitorijuma
- **Trunk** – glavno stablo/sekvenca sa revizijama fajlova
- **Branch** – odvojena i nezavisna sekvenca dobijena kreiranjem nove sekvence ili dobijena nakon grananja postojeće sekvence
- **Head** – poslednja revizija na repozitorijumu tj. vrh sekvence
- **Conflict** – situacija kada postoje dve različite izmene koje su izvršene od strane dva različita korisnika u istom fajlu

Operacije RC sistema (Git terminologija)

- **Clone** – kreiranje radne kopije na osnovu podataka preuzetih sa nekog (udaljenog) repozitorijuma
- **Commit** – smeštanje izmena sa radne kopije na lokalni repozitorijum
- **Fetch** – učitavanje izmena sa udaljenog repozitorijuma
- **Switch/checkout** – prebacivanje sa branch-a na branch
- **Merge** – spajanje i primenjivanje dve različite revizije koje se nalaze na dva različita *branch-a*
- **Pull** – Učitavanje izmena sa udaljenog repozitorijuma i njihova primena na radnu kopiju (Fetch + Merge)

Podela RC sistema

- RC sistemi mogu se podeliti u sledeće dve grupe:
 - Centralizovane
 - Distribuirane

Centralizovani RC sistemi

- Postoji jedan referentni centralni repozitorijum na nekom serveru gde se čuvaju svi potrebni fajlovi, zajednički za sve korisnike sistema
- Na centralnom repozitorijumu se čuva istorija izmena svih fajlova
- Koriste sledeća dva mehanizma za sinhronizaciju izmena od strane više korisnika:
 - **Zaključavanje** – fajl koji se menja od strane jednog korisnika se zaključa, sprečavajući ostale korisnike da vrše izmene, sve dok korisnik ne otključa fajl koji je zaključao
 - **Spajanje** – različite izmene izvršene od strane više korisnika se spajaju i primenjuju automatski ako je moguće, ili ručno ako su izmene u istom fajlu

Osobine centralizovanih RC sistema

- Korisnici prosleđuju svoje izmene drugim korisnicima preko centralnog repozitorijuma
- Potrebno je vršiti sinhronizaciju izmena na repozitorijumu za višekorisnički režim rada
- Kada se nove izmene preuzmu sa centralnog repozitorijuma, one se odmah primenjuju na radnu kopiju

Distribuirani RC sistemi

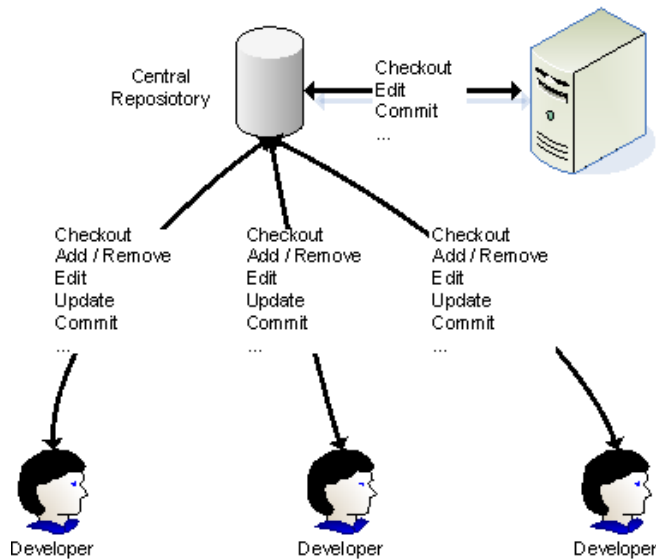
- Svaki korisnik sistema, pored radne kopije, poseduje svoj lokalni repozitorijum
- Na lokalnom repozitorijumu su sačuvane informacije i podaci kao što je istorija izmena fajlova
- Izmene sa repozitorijuma jednog korisnika mogu se slati na/preuzeti sa repozitorijuma drugog korisnika
- Po dogovoru, jedan repozitorijum se može proglasiti referentnim za sve korisnike sistema

Osobine distribuiranih RC sistema

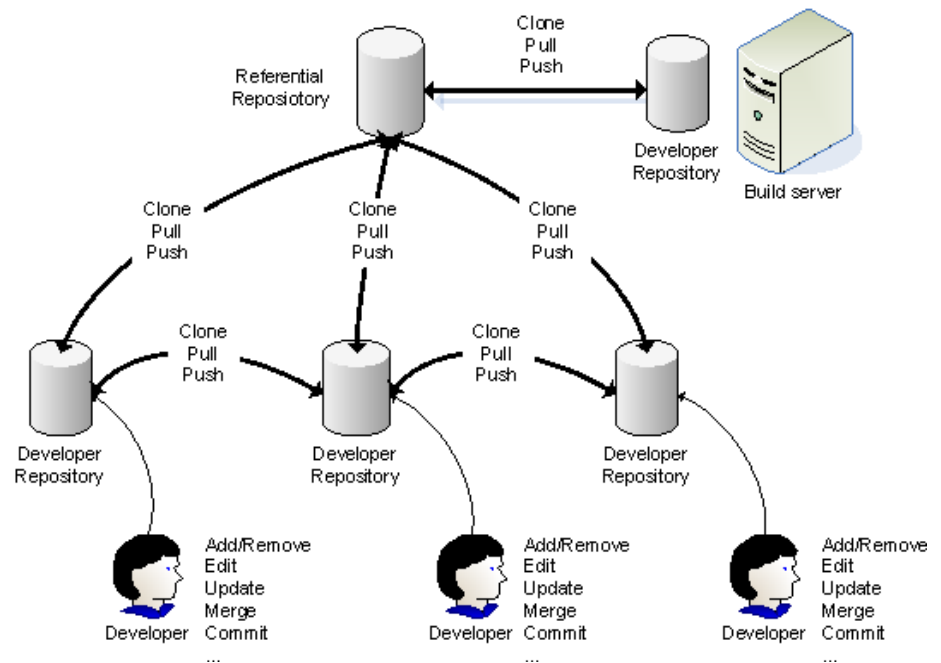
- Lokalni repozitorijum se nalazi na istoj mašini gde je i radna kopija
- Komunikacija između radne kopije i lokalnog repozitorijuma se vrši *offline*, tj. ne mora se uspostaviti posebna veza sa repozitorijumom s obzirom da je na lokalnoj mašini
- Kreiranje nove revizije na lokalnom repozitorijumu i njeno objavljivanje su dve nezavisne operacije s obzirom da kreirana revizija postaje javna samo ako je preuzeta sa ili poslata na drugi repozitorijum
- Kada se izmene sa repozitorijuma jednog korisnika preuzmu i smeste na repozitorijum drugog korisnika, ne moraju se automatski primeniti na radnu kopiju drugog korisnika

Arhitektura

Centralizovani RC sistem (CVS, SVN, ...)

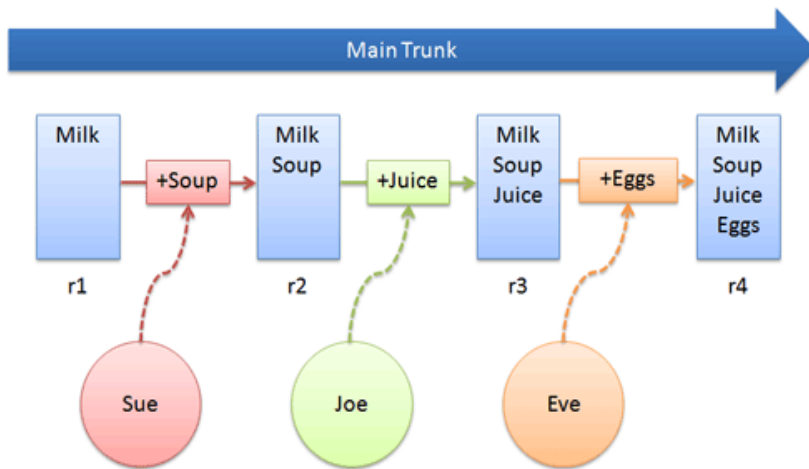


Distribuirani RC siste (Hg, Git, ...)

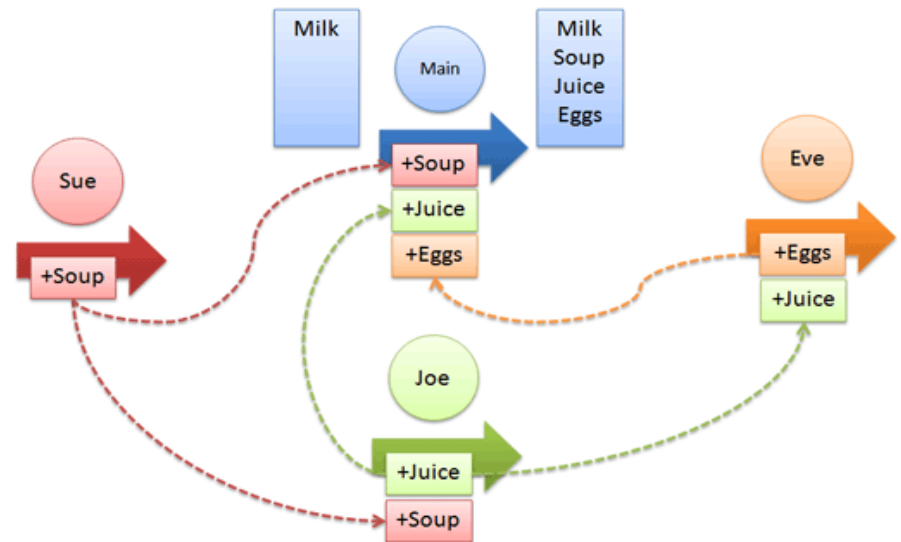


Repozitorijum

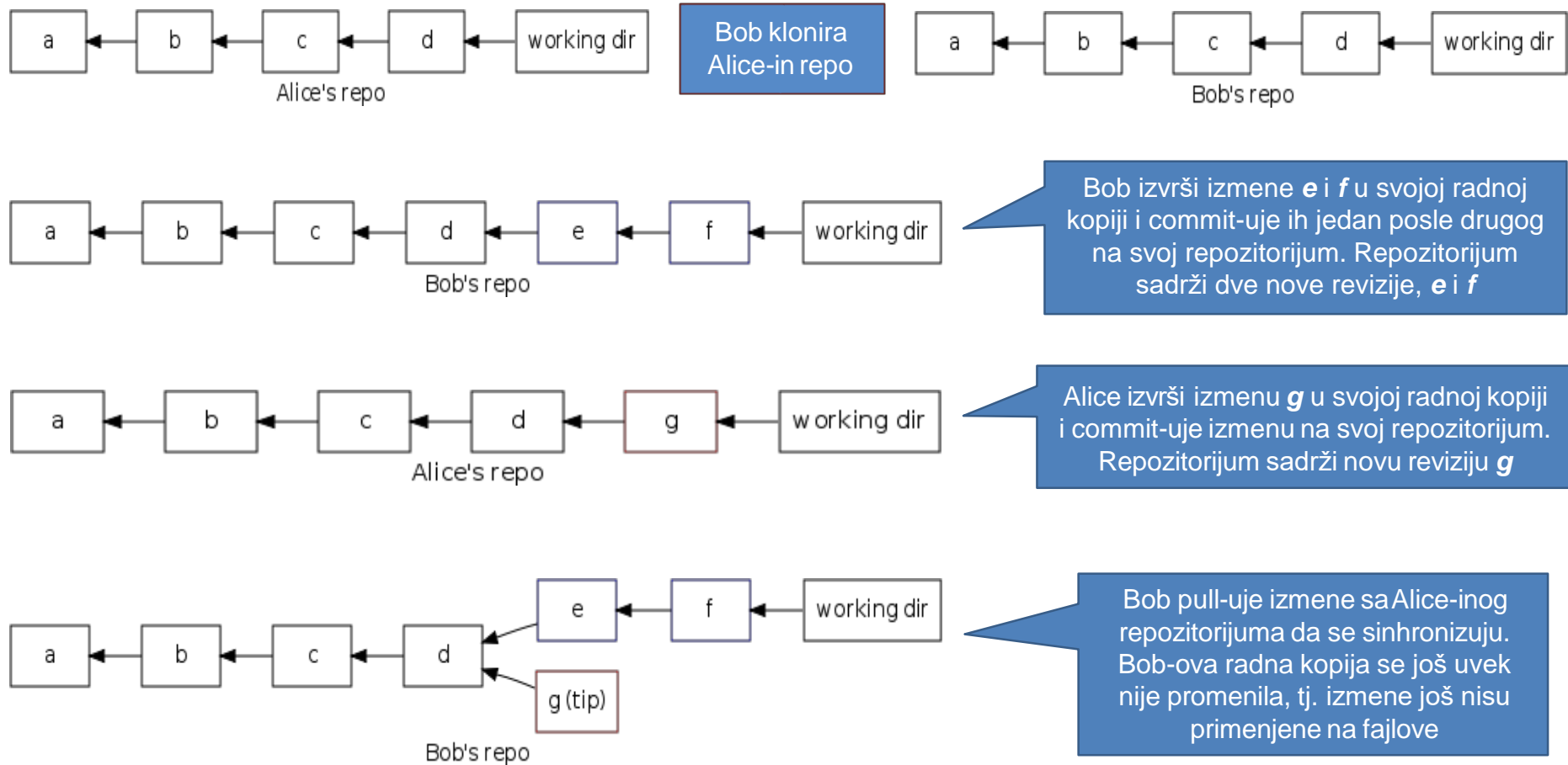
Centralizovani RC sistem



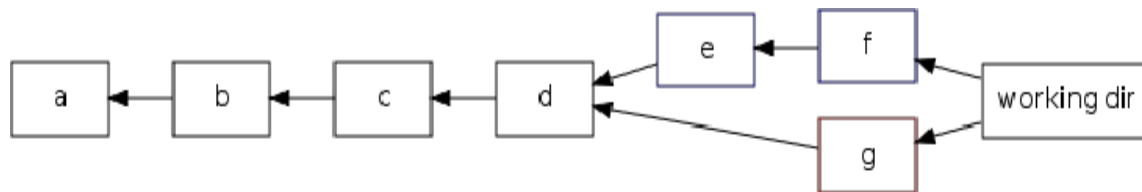
Distribuirani RC sistem



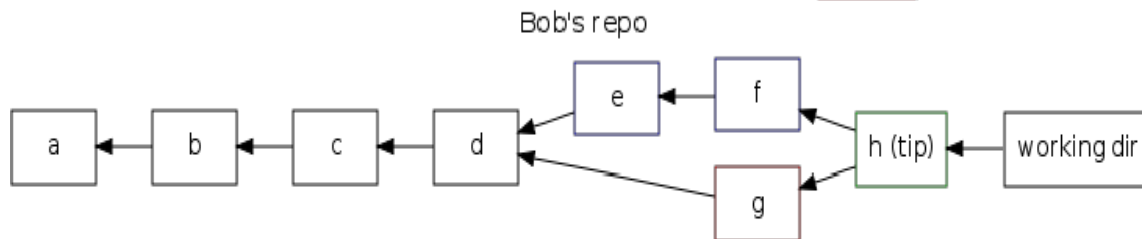
Primenjivanje i spajanje – Primer



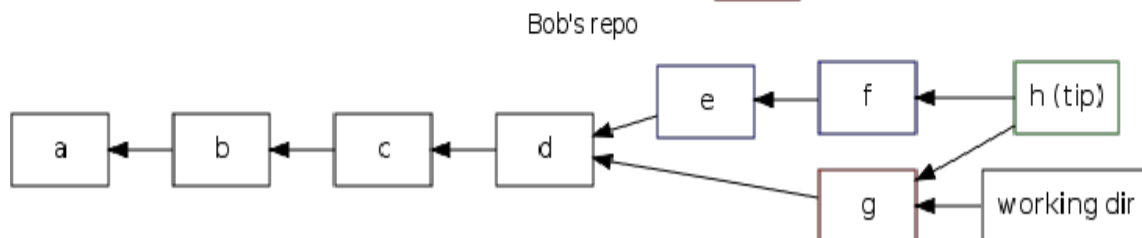
Primenjivanje i spajanje – Primer (nastavak)



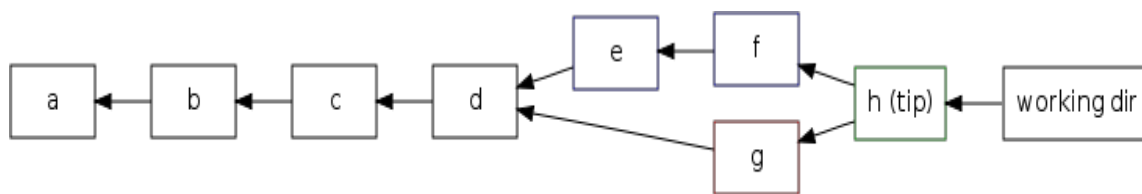
Zbog toga što je Bob commit ovao svoje izmene nakon clone-a, mora da merge-uje svoje izmene sa Alice-inim izmenama



Bob commituje merge-ovane izmene na svoj lokalni repozitorijum. Sad su njegova radna kopija i repozitorijum poravnati



Alice fetch-uje izmene sa Bob-ovog repo-a na svoj repo. Njena radna kopija još uvek nije poravnata sa njenim repozitorijumom



Alice se switch-uje na Bobove izmene, tako da je njena radna kopija poravnata sa najnovijim izmenama

Hvala na pažnji

