

## 9. Git/Github sistem

Git/Github je jedan od najpoznatijih i najčešće korištenih CVS sistema tj. spremišta, skladišta, repozitorijuma za čuvanje i organizaciju izvornog koda softvera, softverskih paketa, odakle se isti može preuzeti, distribuirati i instalirati. Memoriše sve revizije izvršene na softveru radi kontrole promena u softveru, obezbeđuju komunikaciju između članova tima.

Postoji preko 10,6 miliona kreiranih repozitorijuma od 2014. godine, što ga čini jednim od najvažnijih izvora koda na Internet-u. Autor je Linus Torvalds, tvorac Linux OS kernela. Git je besplatan, slobodan i softver otvorenog je koda koji je distribuiran pod uslovima GNU General Public License verzije 2 (GPL).

Postoje dva tipa Git-a: jedan je sistem za kontrolu distribuirane verzije (DVCS - DistributedVersion Control System), a drugi je centralizovani sistem za kontrolu verzija (CVCS - CentralizedVersionControlSystem).

Git-ova izuzetno značajna funkcija je model grananja. Git dozvoljava i podstiče programere da imaju više lokalnih grana koje mogu biti potpuno nezavisne jedna od druge. Kreiranje, spajanje i brisanje tih linija razvoja softvera traje izuzetno brzo i jednostavno.

Većini operacija u Git-u su potrebne samo lokalne datoteke i resursi za rad - obično nema informacija iz drugog računara u mreži. Pošto imamo čitavu istoriju projekta na lokalnom disku, većina operacija izgleda gotovo trenutno. Da bi se pregledala istorija projekta, Git ne mora da odlazi na server da bi dobio istoriju i prikaže je - jednostavno ga čita direktno iz lokalne baze podataka. To znači da se vidi istorija projekta skoro odmah. Ako želite da vidite promene koje su uvedene između trenutne verzije datoteke i datoteke pre mesec dana, Git može da pregleda datoteku pre mesec dana i da napravi lokalni izračun razlike, umesto da zahteva od udaljenog servera da to učini. Takođe se može povući stara verzija datoteke sa udaljenog servera da bi se to uradilo lokalno. Sve u Git-u se proverava pre nego što se sačuva. To znači da je nemoguće promeniti sadržaj bilo koje datoteke ili direktorijuma bez informacije o tome. Ova funkcionalnost je ugrađena u Git na najnižim nivoima i sastavni je deo filozofije repozitorijuma. Ne može se desiti gubitak informacija ili desiti korupcija datoteke a da Git nije u mogućnosti da je otkrije.

Iako se može ručno raditi, Git ima brojne prednosti koje ručno ne mogu da se urade:

- Mogućnost poništavanja promena - ako se napravi greška, možete se vratiti na prethodnu tačku u vremenu da bi se oporavila verziju rada.
- Potpuna istorija svih promena - ako ikada poželimo da vidimo kako je projekat izgledao dan, nedelju, mesec ili godinu pre, možemo proveriti prethodnu verziju projekta da bismo videli tačno kakvo je stanje datoteka tada bilo.
- Dokumentacija o tome zašto su napravljene promene - često je teško zapamtiti zašto je došlo do promene. Sa porukama o urezivanju u Git-u, lako je dokumentovati za buduću referencu zašto pravite promene.
- Sigurnost da se nešto promeni - pošto je lako vratiti prethodnu verziju projekta, može se imati povjerljivost bilo kakve promene koje želite. Ako ne uspeju, uvek se može vratiti na raniju verziju rada.
- Više tokova istorije – mogu se kreirati različite grane istorije da bi se eksperimentisalo sa različitim promenama sadržaja ili da bi se samostalno izgradile različite funkcije. Tada se mogu ponovo spojiti u istoriju glavnog projekta (master, tj. glavna grana) ili ih obrisati ako na kraju ne funkcionišu.

Git je privukao velik broj iskusnih programera zahvaljujući robusnim i pouzdanim funkcijama, kao i neverovatno brza mogućnost grananja.