

# Git-Github



# Git - distribuirani sistem za kontrolu verzije

\* Git je trenutno najpopularniji i najrasprostranjeniji sistem za kontrolu verzije, namenjen praćenju promena izvornog koda tokom procesa razvoja softvera.

\* Iako je razvijen sa idejom podrške koordinaciji tima programera koji zajedno radi na određenom projektu, može biti korišćen i u generalne svrhe, za praćenje promena u okviru proizvoljne kolekcije datoteka. Kako se projekat razvija i evoluira, članovi tima obavljaju testiranje, otklanjaju greške i dodaju nove delove koda znajući da bilo koja verzija koda može biti rekonstruisana u proizvoljnom trenutku.

\* Osobe koje učestvuju u razvoju mogu na osnovu uvida u projektну istoriju da saznaju:

☐ Koje su promene unete?

☐ Ko ih je uneo?

☐ Kada su promene nastale?

☐ Zašto su promene bile neophodne?

Git je primer distribuiranog sistema za kontrolu verzije koji omogućava pun pristup svakoj datoteci,

grani i iteraciji razvoja i svakom korisniku dozvoljava pristup potpunoj i zaokruženoj istoriji svih promena.

Za razliku od centralizovanih sistema za kontrolu verzije, distribuiranim sistemima kao što je Git nije neopodna permanentna veza sa centralnim repozitorijumom.

Repozitorijum, odnosno Git projekat predstavlja kolekciju datoteka i direktorijuma u vezi sa projektom, uključujući i istoriju revizija za svaku pojedinačnu datoteku.

Istorija predstavlja niz verzija iste datoteke u različitim vremenskim trenucima kada je izvršena operacija postavljanja (eng. commit).

Svaka novoizvršena operacija postavljanja nadovezuje se na prethodnu, tako da se u logičkom smislu dobija povezana lista koja može biti razgranata tako da prati različite grane razvoja istog projekta.

Svaki učesnik koji poseduje kopiju repozitorijuma ima na raspolaganju kompletnu bazu koda koji pripada projektu, zajedno sa istorijom njegovog razvoja.

Korišćenjem osnovnih alata u komandnoj liniji, ili nekog od grafičkih okruženja (GUI), korisnicima su omogućene osnovne operacije interakcije sa istorijom, kloniranja repozitorijuma, postavljanja novih verzija koda, kreiranja i spajanja grana, poređenja različitih verzija koda i sl.

Rad sa repozitorijumima olakšava organizovanje i zaštitu koda, bez čega bi rad na velikim open-sourceprojektima bio praktično nezamisliv.

Učesnici u razvoju mogu ispravljati greške ili dodavati nove funkcionalnosti bez bojazni da će time pokvariti ili poremetiti glavni tok razvoja projekta.

Ovo je omogućeno primenom tematskih grana koje se mogu na jednostavan način kreirati, pripojiti glavnom toku ili proglasiti zastarelim, po potrebi.

Uz pomoć mrežnih platformi kao što su GitHub ili Bitbucket, otvaraju se dodatne mogućnosti kada je u pitanju saradnja i transparentnost tokom razvoja projekta.

Javni repozitorijumi omogućavaju razvojnim timovima da saraduju i doprinose zajedničkom cilju stvaranja najboljeg mogućeg proizvoda.

# Osnovne Git komande

Odličan izvor informacija u vezi sa osnovnim i naprednim funkcionalnostima Git-a dostupan je u formi elektronske knjige: <https://git-scm.com/book/en/v2>

Nakon instalacije odgovarajuće verzije Git-a (za Linux, Windows ili macOS), operacije je moguće obavljati

putem komandi koje se izvršavaju iz komandne linije.

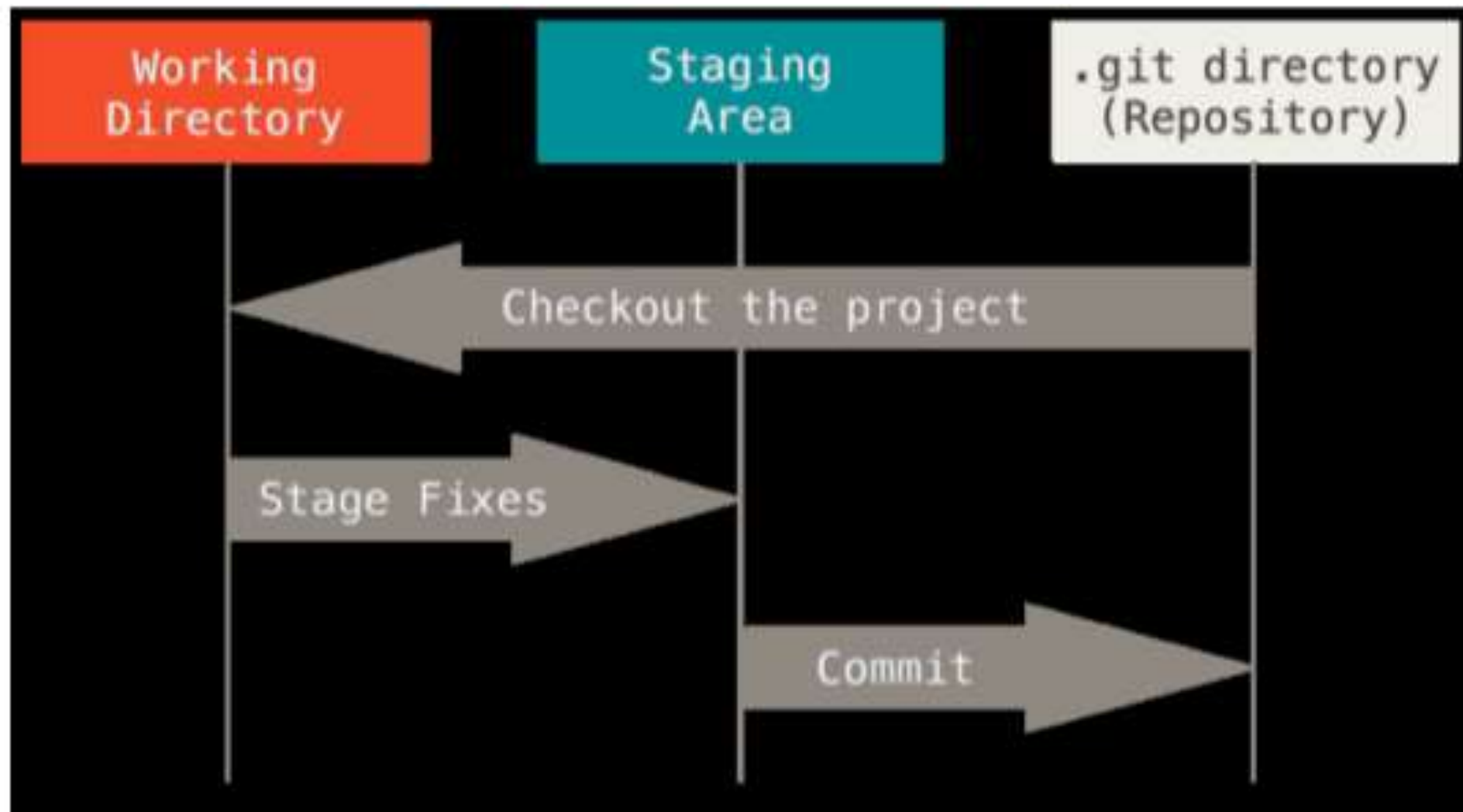
Datoteke koje pripadaju projektu u svakom trenutku se mogu nalaziti u jednom od tri osnovna stanja:

☐ Postavljeno (committed) – podrazumeva da je datoteka bezbedno smeštena u lokalnu bazu podataka

☐ Izmenjeno (modified) – sadržaj datoteke je izmenjen od poslednjeg postavljanja, ali sadržaj još nije pohranjen u bazu

☐ Spremno za postavljanje (staged) – izmenjena datoteka je označena za postavljanje

U skladu sa tim, razlikujemo tri glavne sekcije Git projekta: Git direktorijum (.git directory), radni direktorijum (working directory) i pripremna zona (staging area).



Git direktorijum je mesto gde se čuvaju metapodaci i baza podataka u vezi sa projektom.

Ovo je najvažniji deo Git sistema i tu se smeštaju podaci prilikom kloniranja repozitorijuma, kao i prilikom postavljanja novih verzija.

Radni direktorijum sadrži datoteke koje predstavljaju stanje tekuće verzije projekta. Tekuća verzija se generiše na osnovu podataka iz baze i na raspolaganju je korisniku za unošenje izmena.

Pripremna zona je datoteka, koja je obično smeštena u Git direktorijumu i koja sadrži informacije o tome šta tačno treba da sadrži sledeća verzija koja će biti generisana prilikom postavljanja (commit).

U Git terminologiji ova zona se naziva još i „index“.



Tok rada se obično odvija u tri faze koje se sukcesivno smenjuju:

1. Korisnik modifikuje sadržaj datoteka u radnom direktorijumu
2. Korisnik selektivno označava one datoteke koje trebaju biti postavljene u repozitorijum, čime se one prebacuju u pripremnu zonu.
3. Obavlja se operacija postavljanja (commit), čime se preuzimaju datoteke iz pripremne zone i permanentno se smeštaju u Git repozitorijum.

## **git init**

Komanda git init Inicijalizuje novi Git repozitorijum u tekućem direktorijumu i počinje sa njegovim praćenjem.

Na ovaj način se dodaje skriveni direktorijum (.git direktorijum) u kojem je smeštena interna baza podataka neophodna za praćenje verzije:

```
git init
```

## **git clone**

Komanda git clone kreira lokalnu kopiju postojećeg repozitorijuma sa servera.  
Klonirani repozitorijum

sadrži sve izvorne datoteke, zajedno sa istorijom promena i granama razvoja.

Primer:

```
git clone https://github.com/me/repo.git
```

## git status

Svaka datoteka u radnom direktorijumu može se nalaziti u jednom od dva stanja:

❑ Nepraćena datoteka (untracked)

❑ Praćena datoteka (tracked)

Praćena je ona datoteka o čijim eventualnim promenama Git sistem vodi računa. Takva datoteka može

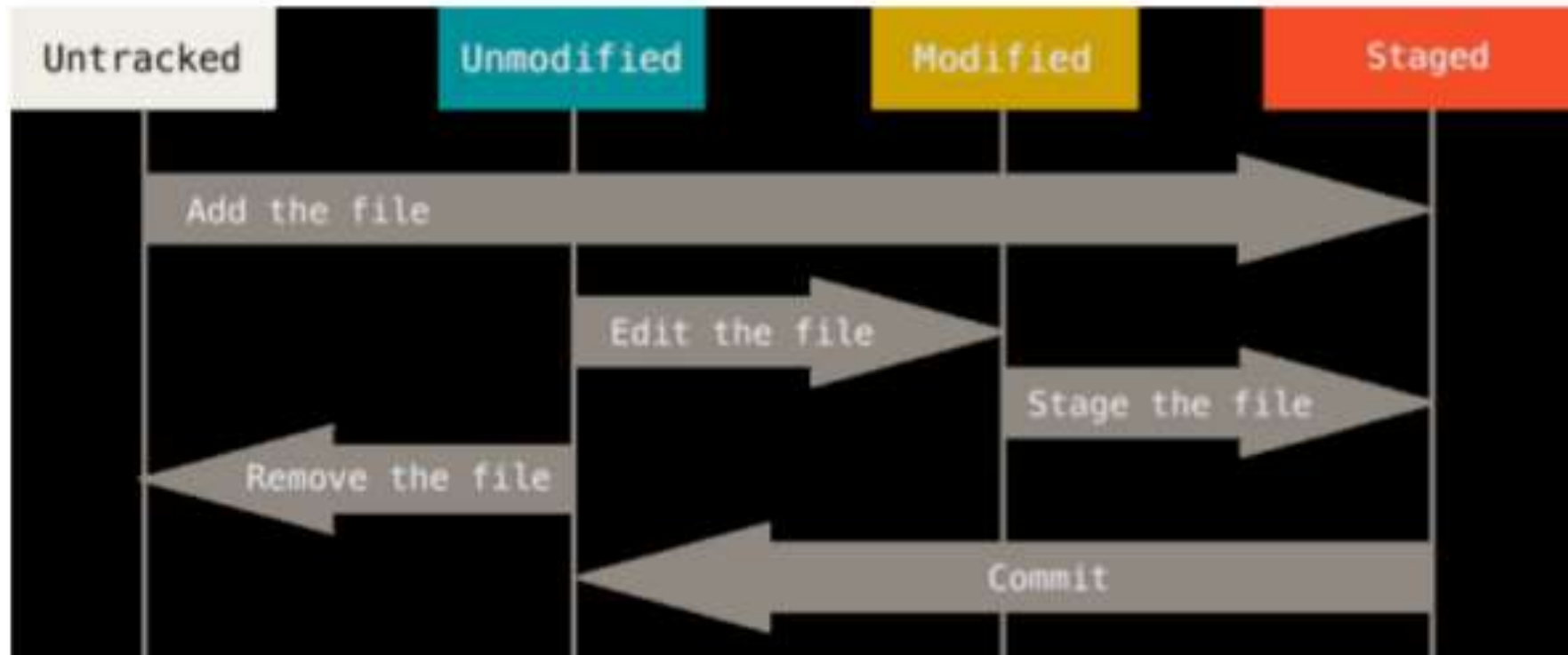
se nalaziti u jednom od tri stanja:

❑ **Neizmenjena** (unmodified) – datoteka čiji sadržaj nije izmenjen od poslednjeg postavljanja

❑ **Izmenjena** (modified) – datoteka čiji sadržaj je promenjen

❑ **Pripremljena za postavljanje** (staged) – prilikom sledećeg postavljanja, biće postavljene samo one datoteke koje se nalaze u ovom stanju, tj. koje se nalaze u pripremnoj zoni (staging area)

Promene stanja datoteka i operacije koje dovode do njih prikazane su na slici:



Komanda git status služi za proveru stanja datoteka u radnom direktorijumu. Primer:

```
$ git status  
On branch master  
Your branch is up-to-date with 'origin/master'.  
nothing to commit, working directory clean
```

## **git add**

Komanda `git add` označava izmenjene datoteke za postavljanje, čime ih prebacuje u pripremnu zonu.

Ovo je prvi korak u kreiranju nove verzije koda. Sve promene koje su prebačene u pripremnu zonu će prilikom postavljanja postati integralni deo sledeće verzije, a samim tim i projektne istorije.

## **git commit**

Komanda `git commit` snima promene iz pripreme zone u repozitorijum, čime se kompletira proces kreiranja nove verzije. Novostvorena verzija predstavlja presek stanja projekta u određenom vremenskom trenutku.

Sve što je prethodno prebačeno u pripremnu zonu pomoću komande **git add** postaće deo nove verzije kreirane komandom **git commit**.

## **git branch**

Komanda git branch kreira novu granu razvoja koda.

## **git merge**

Komanda git merge spaja promene u kodu nastale u dvema odvojenim granama razvoja koda.

## **git pull**

Komanda git pull ažurira sadržaj lokalnog repozitorijuma sadržajem sa udaljenog servera.

## **git push**

Komanda git push ažurira repozitorijum na udaljenom serveru promenama napravljenim u lokalu.

# Šta je GitHub

GitHub (originalno poznat kao — Logical Awesome LLC) je veb-baziran hosting servis za kontrolu verzije, Git. Pruža Git funkcionalnosti: distribuiranu kontrolu revizija i menadžment izvornog koda (eng. Source Control Management — SCM), dodajući dodatne funkcije.

Za razliku od Git-a, koji je striktno alat koji se koristi iz komandne linije, GitHub pruža veb grafički interfejs, radnu površinu i mobilnu integraciju. Takođe pruža kontrolu pristupai nekoliko funkcija za saradnju, kao što su praćenje grešaka (eng. bug tracking), zahteve za dodavanje novih karakteristika (eng. feature request), upravljanje zadacima (eng. task management) i mogućnost pravljenja vikidokumentacije za svaki projekat.



GitHub pruža planove za privatna skladišta kao i besplatne naloge, koji se obično koriste kao hostovi za softverske projekte otvorenog koda. GitHub izveštaji iz aprila 2016. godine tvrde postojanje više od 14 miliona korisnika i 35 miliona skladišta, što GitHub čini najvećim hostom izvornog koda na svetu.

Razvoj GitHub platforme je počeo 1. oktobra 2007. Sajt je pokrenut u aprilu 2008. od strane Tom Preston-Venera, Krisa Vanstrata i P.J. Hajeta, par meseci pošto je bio dostupan u beta izdanju.

Projektima na GitHub-u se pristupa i manipuliše koristeći standardni Git interfejs iz komandne linije preko kojeg su dostupne i sve standardne Git komande. GitHub dopušta registrovanim i neregistrovanim korisnicima da pregledaju javna skladišta na sajtu. Nekoliko desktop klijenata i Git priključaka su kreirani pomoću GitHub-a koji se integrišu sa platformom.

GitHub sajt pruža funkcije slične društvenim mrežama kao što su: dovodi (engl. feeds), pratioci (engl. followers), viki (koristeći softver Gollum) i grafikone saradnje koji ukazuju kako programeri rade na svojim verzijama ("račvama") skladišta i koja račva (i grana te račve) je najnovija.

Korisnik mora da napravi nalog kako bi doprineo stranici, ali javna skladišta mogu biti pregledana i preuzeta od strane bilo koga.

Sa registrovanim korisničkim nalogom, korisnici mogu da diskutuju, upravljaju skladištima, prave nova skladišta, postavljaju doprinose drugim skladištima i pregledaju izmene u kodu.

Softver koji pokreće GitHub je pisan koristeći Ruby on Rails i Erlang od strane GitHub, Inc. razvojnog tima: Kris Vanstrat, P.J. Hajet i Tom Preston-Verner.

GitHub se najviše koristi za izvorne kodove.

Osim izvornog koda, GitHub podržava sledeće formate i funkcije:

Dokumentaciju, uključujući automatski generisane README fajlove u raznim Markdown fajl formatima (pogledati README fajlovi na GitHub-u).

Praćenje problema (uključujući zahteve za novim karakteristikama) sa labelama, prekretnicama, zastupnicima i pretraživačem.

Viki podrška.

Zahteve za pregled izmena sa recenzijama koda i komentarima.

Istorija izvršavanja.

Grafikoni: puls, saradnici, frekvencije koda, mreže, članovi.

Adresar integracija.

E-mail notifikacije.

Opcija prijave za notifikacije o nekome, koristeći znak @.

Emoji podrška.

Mali veb-sajtovi se mogu hostovati sa javnih skladišta na GitHub-u. URL format je: <http://nalog.github.io>.

Ugnježdene liste zadataka unutar fajlova

Vizuelizacija geoprostornih 3D podataka koji se mogu pregledati koristeći novointegrirani STL fajl pregledač koji otvara fajlove na 3D platnu. Pregledač pokreću WebGL i Three.js

Photoshop-ov nativni PSD format se može pregledati i porediti sa prethodnim verzijama istog fajla

# Licenciranje skladišta

GitHub-ovi uslovi korišćenja ne zahtevaju da javni projekti na GitHub-u zadovoljavaju Open Source Definiciju. Iz tog razloga, preporučuje se korisnicima i developerima koji nameravaju da koriste neki softver sa GitHub-a da pročitaju softver licencu u skladištu (obično se nalazi u prvom fajlu nazvanom "LICENSE", "LICENSE.txt", i sl.) kako bi utvrdili da li ispunjava njihove zahteve.

# GitHub Enterprise

GitHub Enterprise je sličan GitHub-ovom javnom servisu ali je dizajniran za razvoj softvera velikih razmera od strane razvojnih timova gde preduzeća žele da hostuju skladišta iza korporacijskig zaštitnog zida.

GitHub takođe vodi i druge servise: sajt u pastebin stilu nazvan **Gist** koji se koristi za hostovanje fragmenata koda, i servis Speaker Deck.

Tom Preston-Verner je prezentovao tada novu Gist karakteristiku tokom Ruby konferencije 2008. godine .

Gist je zasnovan na jednostavnom konceptu pastebin-a sa dodatkom kontrole verzija za fragmente koda, SSL enkripciju za privatne fragmente. Zbog toga što svaki "gist" ima svoje Git skladište, više fragmenata koda se može nalaziti u jednom gist-u i mogu se obraditi od strane Git-a. Dakle, gist-ovi mogu postati mini-projekti.

# Edukacioni program

GitHub je pokrenuo novi program sa nazivom GitHub Student Developer Pack kako bi studentima dao besplatan pristup popularnim alatima za razvoj softvera.

GitHub je pokrenuo ovaj program u partnerstvu sa kompanijama kao što su: Bitnami, Crowdfunder, DigitalOcean, DNSimple, HackHands, Namecheap, Orchestrate, Screenhero, SendGrid, Stripe, Travis CI i Unreal Engine.



# Istorija kompanije

GitHub je prvobitno bio poznat kao Logical Awesome LLC.

- 24. februar 2009: Članovi GitHub tima su najavili, u razgovoru u sedištu [Yahoo!](#)-a, da je prve godine na mreži, GitHub sakupio preko 46,000 javnih repozitorijuma od kojih je 17,000 formirano samo u prethodnom mesecu. U to vreme, oko 6,200 repozitorijuma je odvojeno barem jednom, a 4,600 je spojeno.
- 5. jul 2009: GitHub je objavio da sajt sada koristi više od 100,000 korisnika.
- 27. jul 2009: U drugom razgovoru održanom u [Yahoo!-u](#), [Tom Preston Verner](#) objavio je da je GitHub porastao da bude host za 90,000 jedinstvenih javnih repozitorijuma, 12,000 što se odvaja barem jednom, za ukupno 135,000 repozitorijuma.
- 25. jul 2010: GitHub je saopštio da je host 1 miliona repozitorijuma.
- 20. april 2011: GitHub je saopštio da je host 2 miliona repozitorijuma.
- 2. jun 2011: [ReadWriteWeb](#) je izvestio da je GitHub prevazišao SourceForge i Google Code u ukupnom broju doprinosa za period od januara do maja 2011. godine.
- 9. jul 2012: Piter Levin, generalni partner GitHub-a Andresen Horvic, naveo je da prihodi rastu 300% godišnje od 2008. godine "profitabilan skoro ceo put".
- 16. januar 2013: GitHub je objavio da je prešao 3 miliona označenih korisnika i tada je bio host više od 5 miliona repozitorijuma.
- 23. decembar 2013: GitHub je objavio da je dostigao 10 miliona repozitorijuma.
- U junu 2015. GitHub je otvorio kancelariju u Japanu i to im je prva kancelarija izvan [SAD](#) -a.
- 29. jul 2015: GitHub je objavio da je skupio 250 miliona dolara iz fonda Sequoia Capital. Procenjena vrednost kompanije na približno 2 milijarde dolara.

## Github Octocat

Zaštitna maskota GitHub-a je Octocat, mačka u ljudskom obliku sa udovima hobotnice, crtana u manga stilu.

Organizaciona struktura Do decembra 2012. GitHub je bila horizontalno organizovana bez srednjeg menadžmenta; drugim rečima, "svako je menadžer" (radničko samoupravljanje). Zaposleni mogu da biraju da rade na projektima koji ih interesuju (otvorena raspodela). Međutim, plate je određivao izvršni direktor.

2014. godine je uveden sloj srednjeg menadžmenta.

Finansije GitHub je start-up kompanija koja u svojoj prvoj godini obezbedila dovoljno prihoda da finansira jedino svoja tri osnivača i da počne da zapošljava radnike. U julu 2012. godine, četiri godine nakon što je kompanija osnovana, Andresen Horvic investira 100 miliona dolara početnog kapitala.

