GitHub

**Kaj je GitHub?**

**GitHub** je spletna platforma za shranjevanje, vodenje in sodelovanje na projektih programske opreme, ki uporablja različico nadzora, imenovano **Git**. GitHub omogoča razvijalcem, da svoje projekte shranjujejo v **repozitorijih** (shrambah kode), hkrati pa ponuja funkcije za sodelovanje, revizijo in sledenje spremembam kode.

GitHub je še posebej priljubljen zaradi naslednjih funkcionalnosti:

* **Vodenje različic (version control)**: Omogoča sledenje spremembam v kodi, vračanje na prejšnje različice in sodelovanje več razvijalcev na istem projektu brez prepisovanja dela.
* **Sodelovanje**: Razvijalci lahko na enostaven način sodelujejo na istih projektih z uporabo t.i. "pull requestov", kjer predlagajo spremembe, ki jih nato preverijo drugi člani ekipe.
* **Javna in zasebna skladišča (repositories)**: Projekti so lahko javni (dostopni vsem) ali zasebni (dostopni le določenim uporabnikom).
* **Integracija z drugimi orodji**: GitHub se integrira z orodji za avtomatizacijo (CI/CD), analitiko kode in komunikacijske platforme (npr. Slack, Microsoft Teams).

**Kako deluje GitHub?**

GitHub temelji na različici nadzora **Git**, ki deluje po decentraliziranem modelu. Vsak razvijalec ima lahko kopijo celotnega repozitorija na svojem računalniku, kjer dela lokalno. Nato spremembe sinhronizira s centralnim strežnikom (GitHub), kjer lahko drugi člani ekipe preverijo in sprejmejo spremembe.

Postopek običajno poteka tako:

1. **Kloniranje (clone)**: Razvijalec si prenese repozitorij na svoj računalnik.
2. **Urejanje (commit)**: Uporabnik lokalno naredi spremembe v kodi in jih zabeleži z ukazom commit. Vsaka sprememba je označena z opisom, kaj je bilo spremenjeno.
3. **Oddaja sprememb (push)**: Ko so spremembe pripravljene, jih razvijalec naloži nazaj v GitHub repozitorij.
4. **Pregled (pull request)**: Če gre za sodelovanje več razvijalcev, lahko nekdo predlaga spremembo z "pull requestom", kjer drugi člani pregledajo kodo in predlagajo izboljšave.
5. **Združevanje (merge)**: Ko so spremembe odobrene, se združijo z glavno vejo kode.

**Zakaj se uporablja GitHub?**

GitHub je uporaben za več namenov:

* **Sodelovanje**: Olajša delo v ekipah, saj omogoča več razvijalcem, da delajo na istem projektu, ne da bi prišlo do konflikta med spremembami.
* **Sledenje spremembam**: Pomaga slediti vsaki posamezni spremembi kode, kar omogoča vračanje na prejšnje različice, če pride do napake.
* **Odprtokodne skupnosti**: GitHub je dom mnogih odprtokodnih projektov, kjer lahko razvijalci iz vsega sveta prispevajo k projektom.
* **Automatizacija**: Z integracijo s CI/CD orodji omogoča avtomatizacijo nalog, kot so testi in uvajanje programske opreme.

### Kako bi ekipa razvijalcev uporabila GitHub?

Uporaba GitHub-a za razvoj je prilagodljiva in jo lahko ekipa razvijalcev uporablja v različnih oblikah:

#### 1. **Web development (spletni razvoj)**

Ekipa, ki razvija spletno stran ali aplikacijo, bi lahko na GitHub-u ustvarila repozitorij, kjer bi hranila HTML, CSS, JavaScript in druge datoteke. Lahko bi uporabljali GitHub v naslednjih korakih:

* **Razvoj funkcionalnosti v različnih vejah (branches)**: Vsaka funkcionalnost (npr. nova stran ali funkcionalnost prijave) bi bila razvita v svoji veji, da se loči od glavne različice.
* **Pregled kode (Code review)**: Ko nekdo predlaga spremembe, drugi člani pregledajo kodo pred združitvijo.
* **Integracija z orodji za avtomatizacijo (CI/CD)**: GitHub lahko uporabi integracijo z orodji, kot je Jenkins ali GitHub Actions, za avtomatsko testiranje in uvajanje aplikacije na strežnik ob vsakem potisku sprememb.

#### 2. **Game development (razvoj iger)**

Pri razvoju iger se GitHub uporablja za vodenje projekta in sledenje spremembam v kodeksu in umetniških sredstvih (assets). Ekipa bi lahko delala na naslednji način:

* **Veje za različne funkcije**: Razvijalci bi delali na različnih delih igre (npr. fizika, umetna inteligenca, uporabniški vmesnik) v ločenih vejah.
* **Skladiščenje sredstev**: Uporabljali bi Git LFS (Large File Storage), ki omogoča shranjevanje velikih datotek, kot so teksture, modeli, glasba in video posnetki.
* **Sodelovanje z oblikovalci in razvijalci**: GitHub omogoča enostavno sodelovanje med programerji in umetniki z jasno zgodovino sprememb.

#### 3. **Desktop development (razvoj namiznih aplikacij)**

Ekipa, ki razvija namizno aplikacijo, bi GitHub uporabila za:

* **Veje za različne funkcionalnosti**: Vsak razvijalec bi delal na svoji veji, ne da bi motil druge razvijalce.
* **Rešitve napak in izboljšave**: GitHub omogoča enostavno sledenje hroščem (bugs) in nalogam prek GitHub Issues ali z integracijo z orodji, kot je Jira.
* **Objave različic**: GitHub omogoča ustvarjanje oznak (tags) in izdajanje novih različic aplikacije, kar olajša nadgradnje in distribucijo.

**Merge** (združitev) v okolju GitHub je proces združevanja dveh različnih vej (branches) v eno. Običajno se spremembe iz ene veje (npr. razvijalčeve delovne veje) združijo v glavno vejo projekta (pogosto imenovano **main** ali **master**). Merge je ključen za sodelovanje ekip razvijalcev, saj omogoča sočasno delo več ljudi na istem projektu, ne da bi prišlo do konfliktov med spremembami.