

Seminar

Gitflow

Dorian Janžetić, Antonio Babić, Antonio Puhanić

Tehnički Fakultet Rijeka

Uvod

Gitflow je posebna vrsta workflow-a.

Dizajniran je tako da striktno definira početne grane na kojemu se cijeli projekt bazira.

Pružá framework koji je efikasan za velike projekte, ili projekte koji imaju vremenski rok.

Stirktno definira ulogu svake grane.

Gitflow koristi sve normalne funkcije git-a uz svoje nadogradnje.

U gitflow projektima se definira pojedinačna grana za sve.

Npr: *bugfix*, *hotfix*, *feature* i sl.

Glavna grana na kojoj se radi je *develop* i na nju se nadovezuju ostale grane poput vec navedenih.

Kada je projekt spreman, *develop* grana se spaja u *master* granu.

Da bi se uopće moglo raditi na projektu uz Gitflow potrebno ga je prvo instalirati.

Jer je to zapravo nadogradnja na postojeći Git.

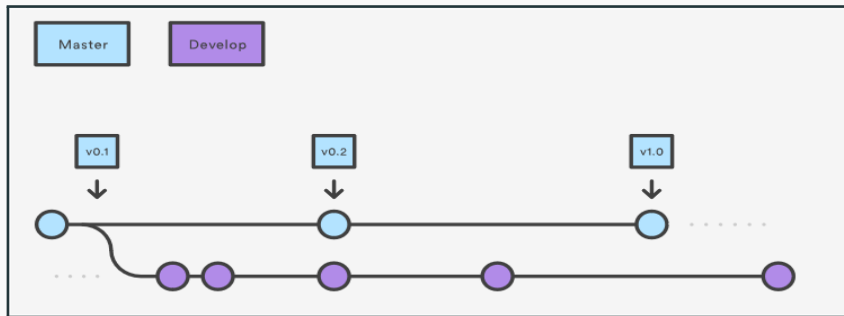
Instalacija je jednostavna, za OSX sisteme se može koristiti naredba *brew install git-flow*, a za Windows OS klasičnim preuzimanjem i instaliranjem.

Za inicijalizaciju Gitflow projekta umjesto *git init*, koristi se *git flow init*.

Ne mijenja ništa u projektu osim što stvara unaprijed definirane grane.

Gitflow grane

Umjesto uobičajene jedne glavne *master* grane, Gitflow koristi dvije. Druga je *develop* grana.



Develop grana je ona na kojoj se odvija većina projekta, iz koje se stvaraju i vraćaju nove grane.

Za objavljivanje tj. završavanje projekta sve se spaja u *develop* granu.

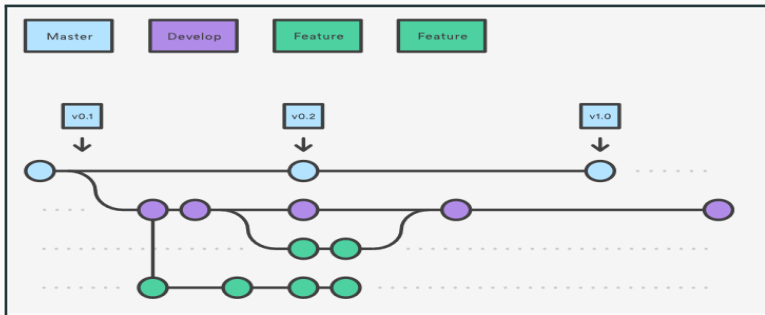
Zbog ovog načina workflow-a poželjno je svaku objavu ili commit tagirati.

Ovako se bilježi cijela povijest razvoja projekta ili programa.

Na repozitoriju treba postojati *develop* grana koju svi developeri prate.

Feature grana

Kad god radimo na novoj značajci (*featureu*) stvaramo novu *Feature* granu koja se grana iz *Develop* grane. Kada napravimo značajku spajmo ju sa *Develop* granom. Feature grane se **nikad** ne spajaju direktno u master granu.



Korištenje feature grane

ako nemamo git-flow ekstenzije: Stvaranje grane: `git checkout develop`
`git merge`

*feature_branchSpajanje grane : `git checkout develop`
`git merge featurebranch`*

ako imamo git-flow ekstenzije: Stvaranje grane: `git flow feature start`
feature_branchSpajanje grane : `git flow feature finish featurebranch`

Kada smo blizu izdavanju nove verzije *forkanjem Develop* grane stvaramo *Release* granu. U ovu granu se smiju dodavati samo *bug fixovi* i dokumentacija vezana za izdavanje nove verzije. Ne smiju se dodavati nikakve nove značajke.



Prednost ovakvog pristupa je u tome što dio tima može nastaviti raditi na značajkama za sljedeću verziju dok drugi dio tima dovršava detalje vezane uz verziju koja uskoro izlazi. Kada dovršimo *Release* granu spajamo ju i s *Master* i s *Develop* granom. S *Develop* granom ju spajamo jer je moguće da smo dodali bitna ažuriranja na *Release* granu koja želimo imati uključena u daljni razvoj projekata.

ako nemamo git-flow ekstenzije: Stvaranje grane: `git checkout develop`
`git checkout -b release/0.1.0` Spajanje grane: `git checkout develop` `git merge release/0.1.0`

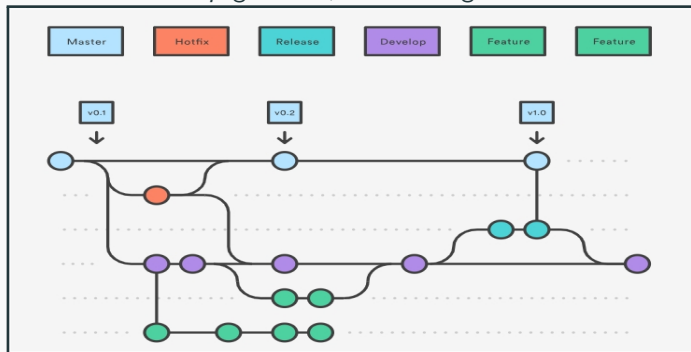
ako imamo git-flow ekstenzije: Stvaranje grane: `git flow release start 0.1.0` Switched to a new branch 'release/0.1.0' Spajanje grane: `git checkout master` `git checkout merge release/0.1.0` `git flow release finish '0.1.0'`

Hotfix grana

Još jedna vrsta grane u Gitflow-u je *hotfix* grana, te služi za održavanje ili hitne izmjene.

Hotfix grana je slična *release* ili *feature* grani, samo što se bazira na *master* grani umjesto *develop* grani.

Nakon što je popravak ili izmjena gotova *hotfix* grana se spaja (merge) sa *master* i *develop* granama, te *master* grana dobiva novi broj verzije.



Ako želimo napraviti *hotfix* granu, možemo na 2 načina ovisno imamo li git-flow ekstenziju:

- bez git-flow ext.: **git checkout master**

git checkout -b hotfix_branch

- sa git-flow ext.: **git flow hotfix start hotfix_branch**

Nakon što smo gotovi sa *hotfix* granom samo ju spajamo sa drugim granama te ju nakon toga obrišemo.

Gitflow je izvrstan za korištenje kod razvijanja softwarea koji se bazira na izdanjima, tj. verzijama i ima zasebnu granu na kojoj se mogu vršiti mali popravci i izmjene,

Radni tijek u gitflow-u ukratko je:

1. *develop* grana se stvara od *master* grane
2. *release* grana se stvara od *develop* grane
3. *feature* grane se stvaraju od *develop* grane
4. Kada je novi *feature* gotov spaja se u *develop* granu
5. Kada je *release* gotov spaja se u *develop* i *master* grane
6. Ako se nadje greška u *master* grani onda se stvara nova *hotfix* grana
7. Kada je greška otklonjena u *hotfix* grani, ponovno se spaja u *master* i *develop* grane

Bibliografija

- [1] Atlassian.
Gitflow workflow tutorials.
- [2] Vincent Driessen.
Introducing gitflow.
- [3] Vincent Driessen.
A successful git branching model.