Git-Flow

Git Flow is:

▸ **A Philosophy**
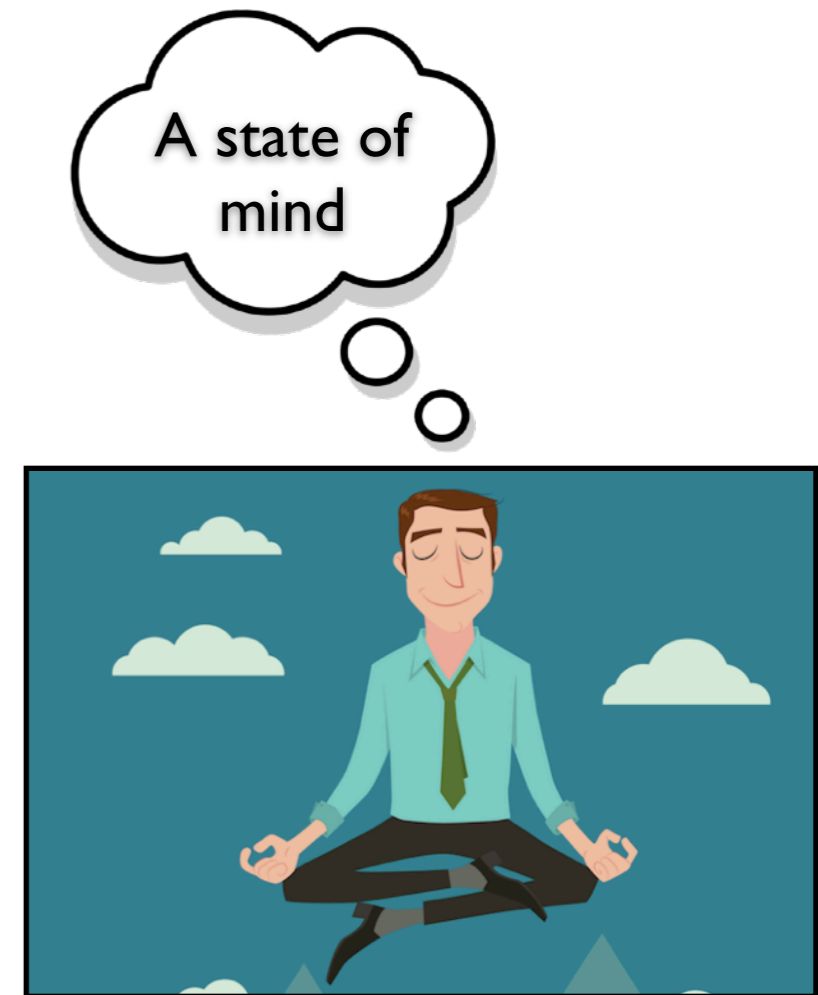
✦ **Optimal for Agile Software Development**
  - **Innovation**
  - **Continuous Delivery**

▸ **A Working Principle**

✦ **Please follow it**

▸ **An Application (extension to git)**

✦ **Already installed in Singularity Container**

✦ **brew install git-flow-avh # (Mac)**

✦ **sudo apt-get install git-flow # (linux)**

✦ **https://github.com/petervanderdoes/gitflow-avh**

A state of mind

The Git-Flow Manifesto

Vincent Driessen (2010)

http://nvie.com/posts/a-successful-git-branching-model/

git branching model

Highly Recommended!

The Git-Flow Manifesto: Takeaways

▸ **master is for releases only**

▸ **develop**
  - **Not ready for pubic consumption but compiles and passes all tests**

▸ *Feature branches*
  - *Where most development happens*
  - *Branch off of develop*
  - *Merge into develop*

▸ *Release branches*
  - *Branch off of develop*
  - *Merge into master and develop*

▸ *Hotfix*
  - *Branch off of master*
  - *Merge into master and develop*

▸ *Bugfix*
  - *Branch off of develop*
  - *Merge into develop*

Temporary
(terminated after merge)

1) Enable git flow for the repo

- ‣ **git flow init -d**

2) Start the feature branch

- ‣ **git flow feature start newstuff**
- ‣ **Creates a new branch called feature/newstuff that branches off of develop**

3) Push it to GitHub for the first time

- ‣ **Make changes and commit them locally**
- ‣ **git flow feature publish newstuff**

4) Additional (normal) commits and pushes as needed

- ‣ **git commit -a**
- ‣ **git push**

5) Bring it up to date with develop (to minimize big changes on the ensuing pull request)

- ‣ **git checkout develop**
- ‣ **git pull origin develop**
- ‣ **git checkout feature/newstuff**
- ‣ **git merge develop**

6) Finish the feature branch (don't use git flow feature finish)

- ‣ **Do a pull request on GitHub from feature/newstuff to develop**
- ‣ **When successfully merged the remote branch will be deleted**
- ‣ **git remote update -p**
- ‣ **git branch -d feature/newstuff**

Life cycle of a feature branch (using git-flow)

For more information

JEDI Git Flow page
https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/developer/developer_tools/getting-started-with-gitflow.html

The Git Flow manifesto (all you need to know about the philosophy):
http://nvie.com/posts/a-successful-git-branching-model/

Git Flow cheat sheet:
https://danielkummer.github.io/git-flow-cheatsheet/

Git avh (a fork of the original, with added features):
https://github.com/petervanderdoes/gitflow-avh

Atlassian git-flow tutorial (philosophy and application):
https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow