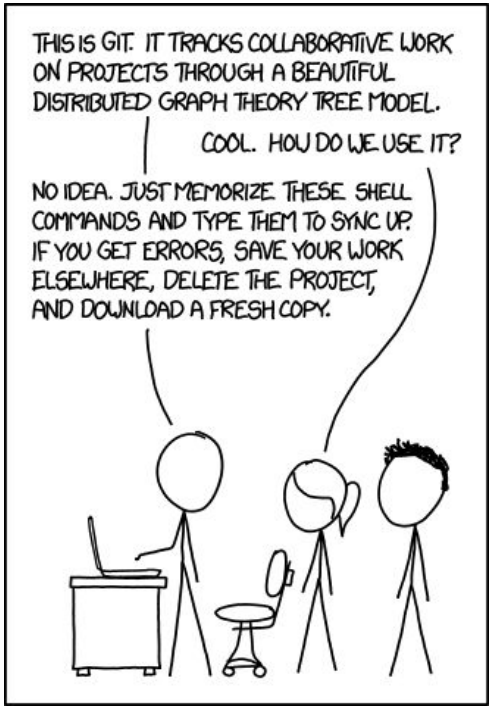
An intro to Git

Orlando Code Camp 2018

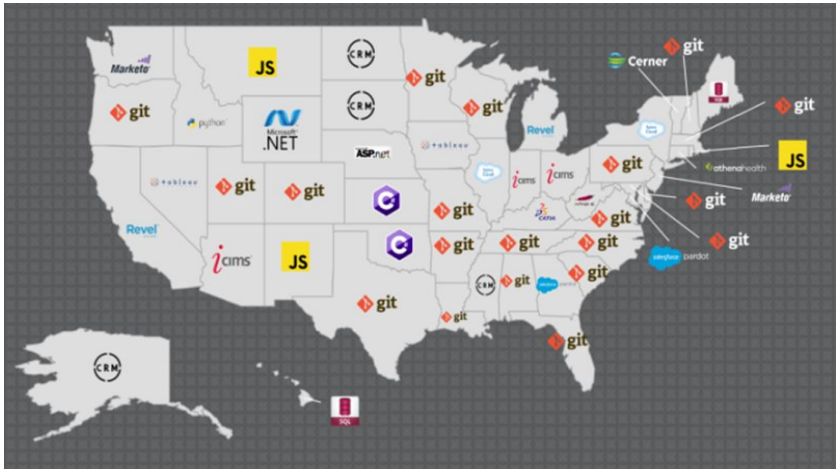
Misnomers about Git?

- It's too hard to use.
- It's only command line based.
- Takes too long to use.
- My project is too big and complicated for that.



Why use this tool?

- Multiple backups fast
- Work in parallel on the same file
- Develop features in different ‘versions’ or branches
- Fast rollback and feature switching
- It’s a skill in high demand



My Path to Version Control

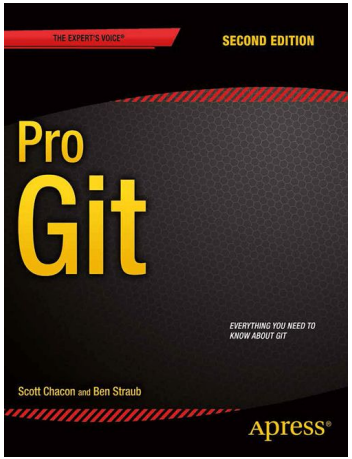
- Lots of information.
- What's the 'best'?
- How do I do 'X'?
- GIT, SVN, etc?
- What do I want out of version control?

Code School
a Pluralsight company

PLURALSIGHT

GitHub Training

Atlassian



Git Words

add
am
apply
archive
bisect
blame
branch
bundle
cat-file
checkout
cherry-pick
clean
clone
commit
commit-tree
config
count-objects
daemon
describe
diff
diff-index
fast-import
fetch
filter-branch
for-each-ref
format-patch
fsck
gc
gitk
grep
hash-object
help
init
instaweb
log
ls-files
ls-tree
merge
merge-base
mergetool
mv
pop
prune
pull
push
read-tree
rebase
reflog
remote
request-pull
reset
revert
rev-list
rev-parse
rm
send-email
shortlog
show
show-ref
stage
stash
status
submodule
svn
symbolic-ref
tag
update-index
update-ref
update-server-info
verify-pack
write-tree

Git Words

add
am
apply
archive
bisect
blame
branch
bundle
cat-file
checkout
cherry-pick
clean
clone
commit
commit-tree
config
count-objects
daemon
describe
diff
diff-index
fast-import
fetch
filter-branch
for-each-ref
format-patch
fsck
gc
gitk
grep
hash-object
help
init
instaweb
log
ls-files
ls-tree
merge
merge-base
mergetool
mv
pop
prune
pull
push
read-tree
rebase
reflog
remote
request-pull
reset
revert
rev-list
rev-parse
rm
send-email
shortlog
show
show-ref
stage
stash
status
submodule
svn
symbolic-ref
tag
update-index
update-ref
update-server-info
verify-pack
write-tree

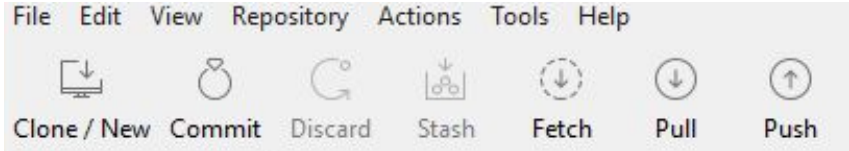
Tool that we'll use

Let's Start !

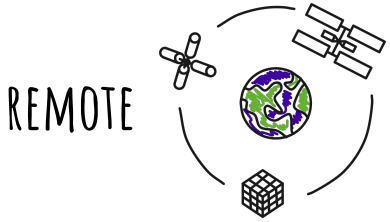
Clone/ Init

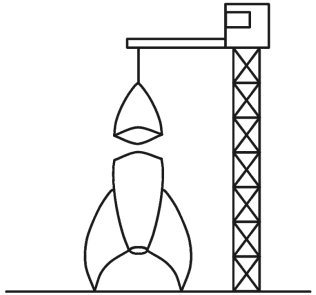
Creates a new repo or copies an existing one

You can always add ‘remotes’ but it's easier to start from there.

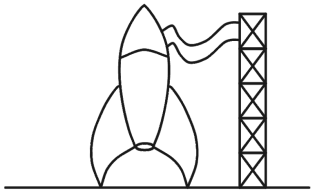


Typical Workflow

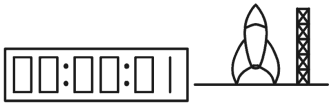




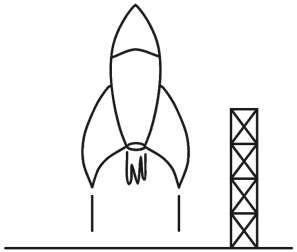
Add



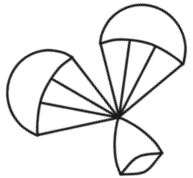
Stage



Commit



Push

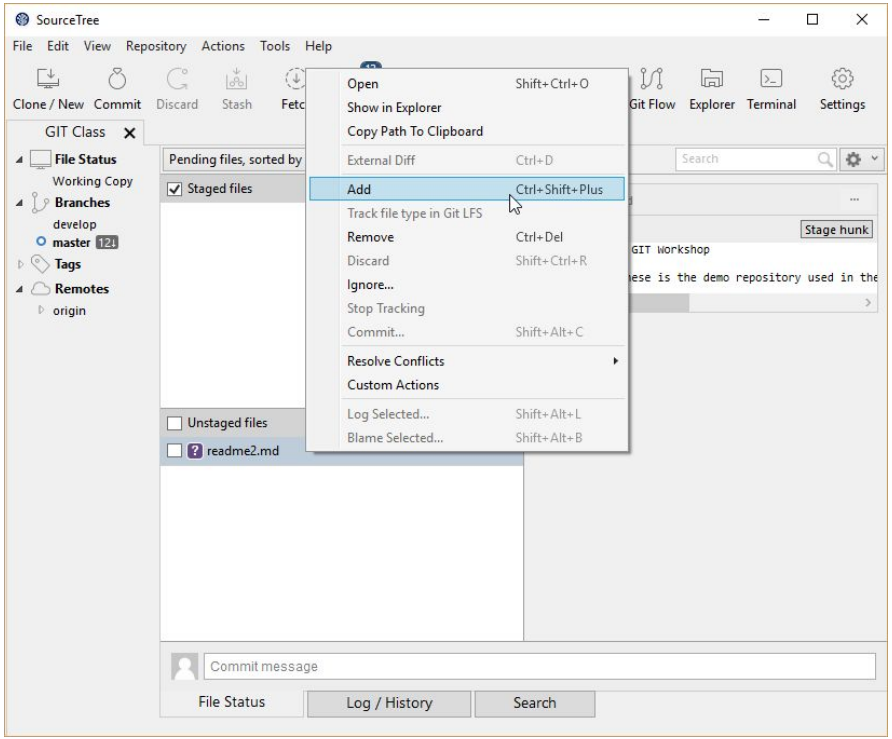


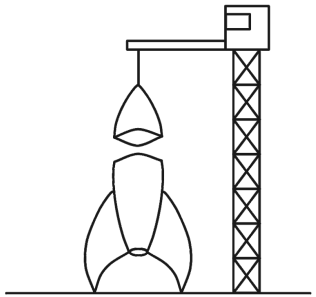
Pull

ADD

Tells GIT to start tracking the files.

Ignore allows you to not track files in a folder - compiled files or temp files typically.



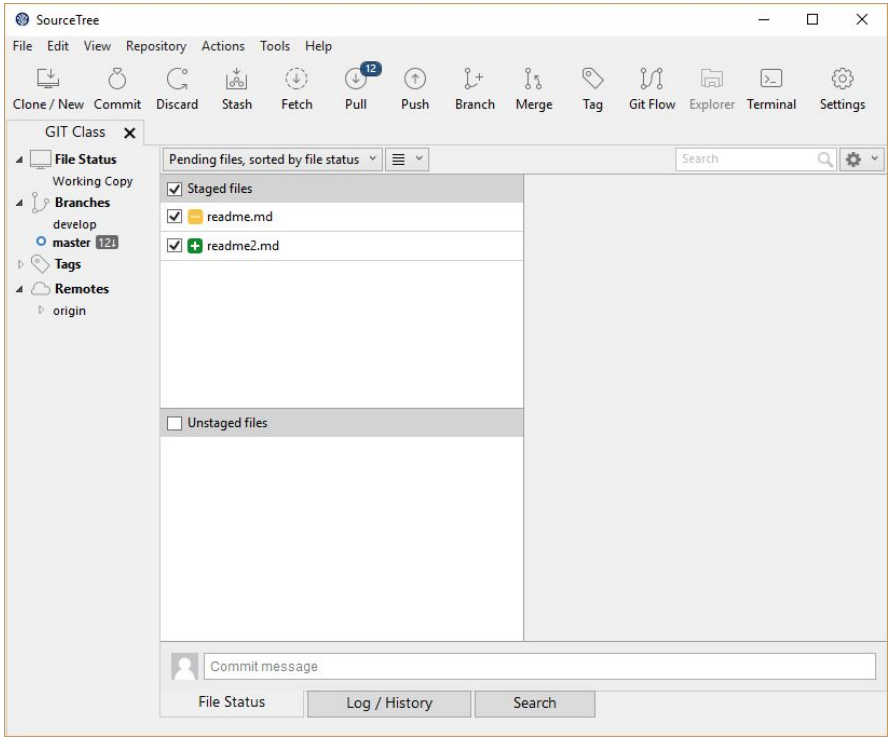


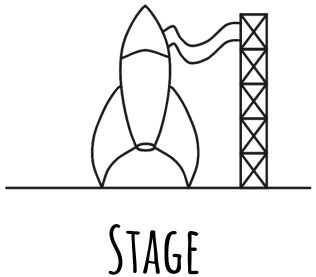
ADD

STAGE

Lets you get your commit where you want it.

TIP:
You can split your changes into multiple commits for tracking.

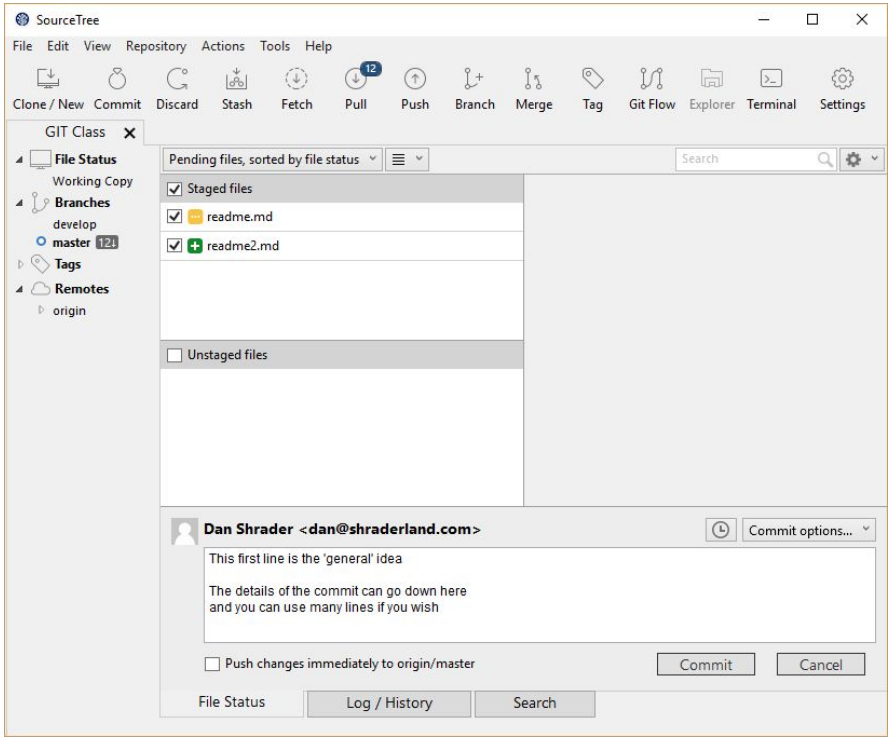


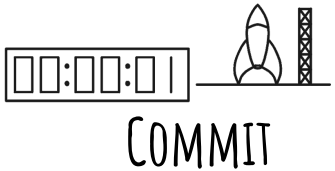


COMMIT

Snapshot in time of your project.

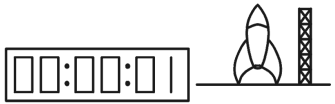
Be detailed on your descriptions. You'll thank yourself later and so will the other contributors to the project.





Commit

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

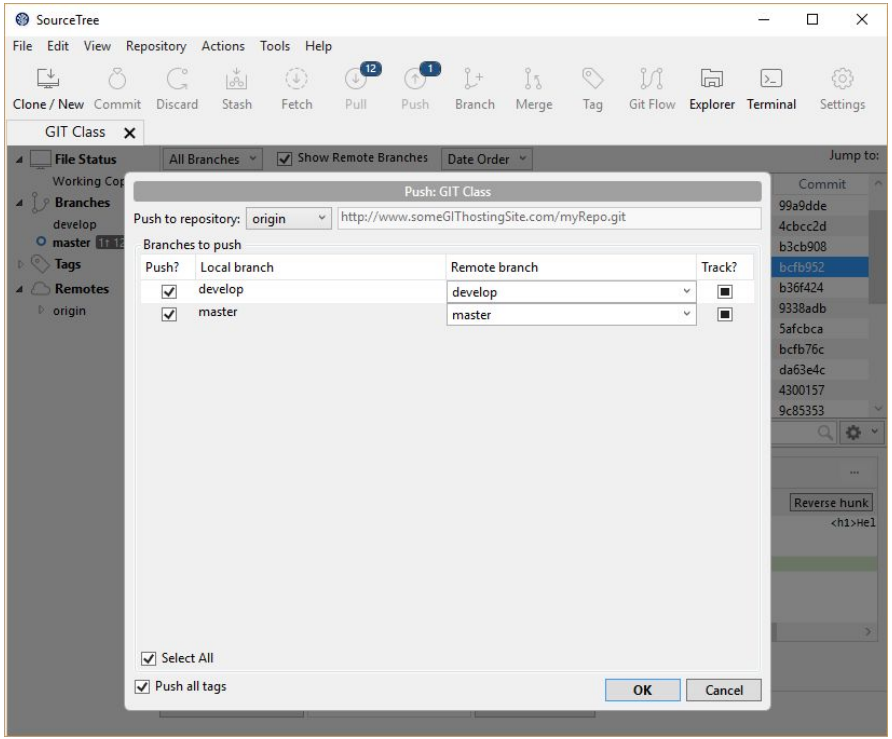


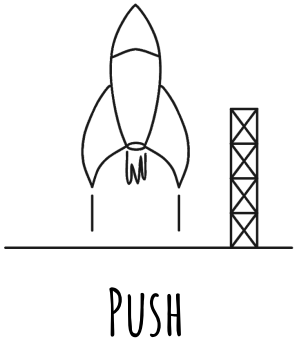
Commit

PUSH

Sends your commits to a centralized server.

Note: there are different types of workflows. We are referencing the ‘centralized’ workflow.





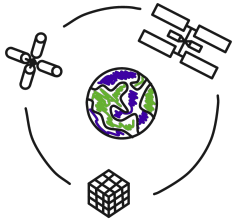
Remote

A place where your code is hosted.

Bitbucket

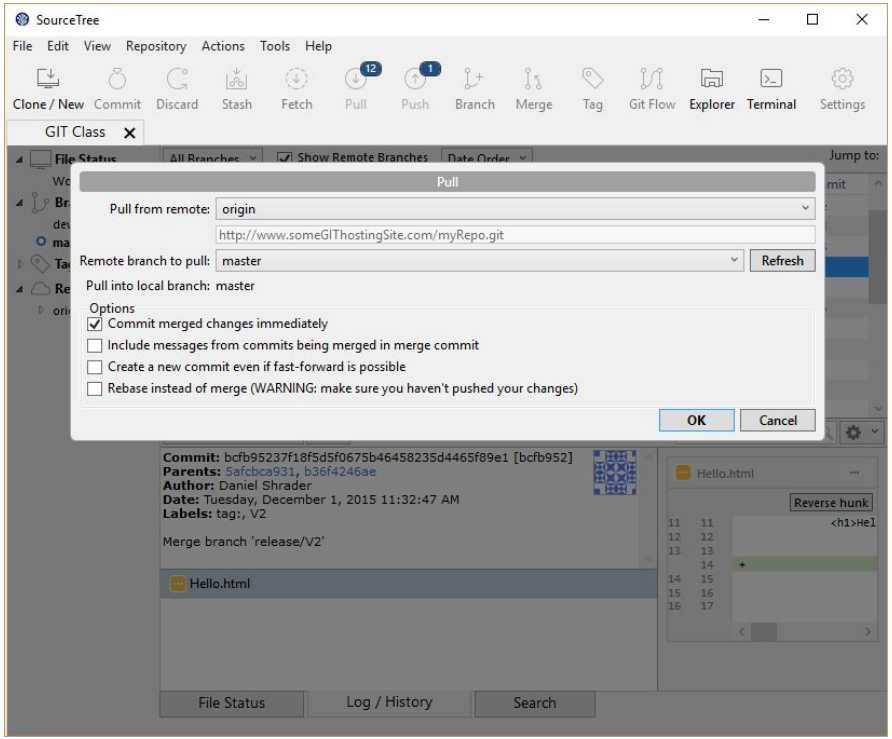
GitHub

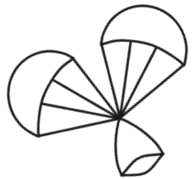
GitLab



PULL

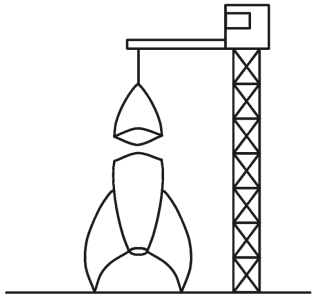
Brings changes committed to the remote into your branch.



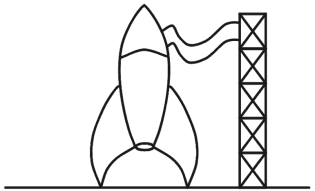


PULL

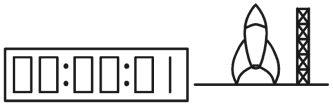
Workflow Once More



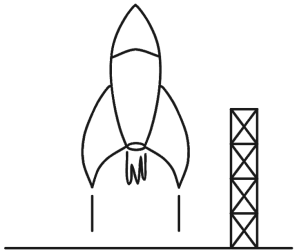
Add



Stage

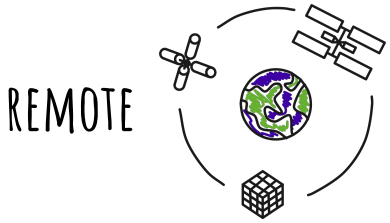


Commit



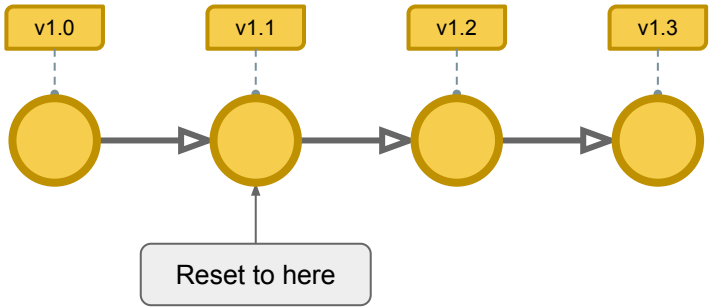
Push

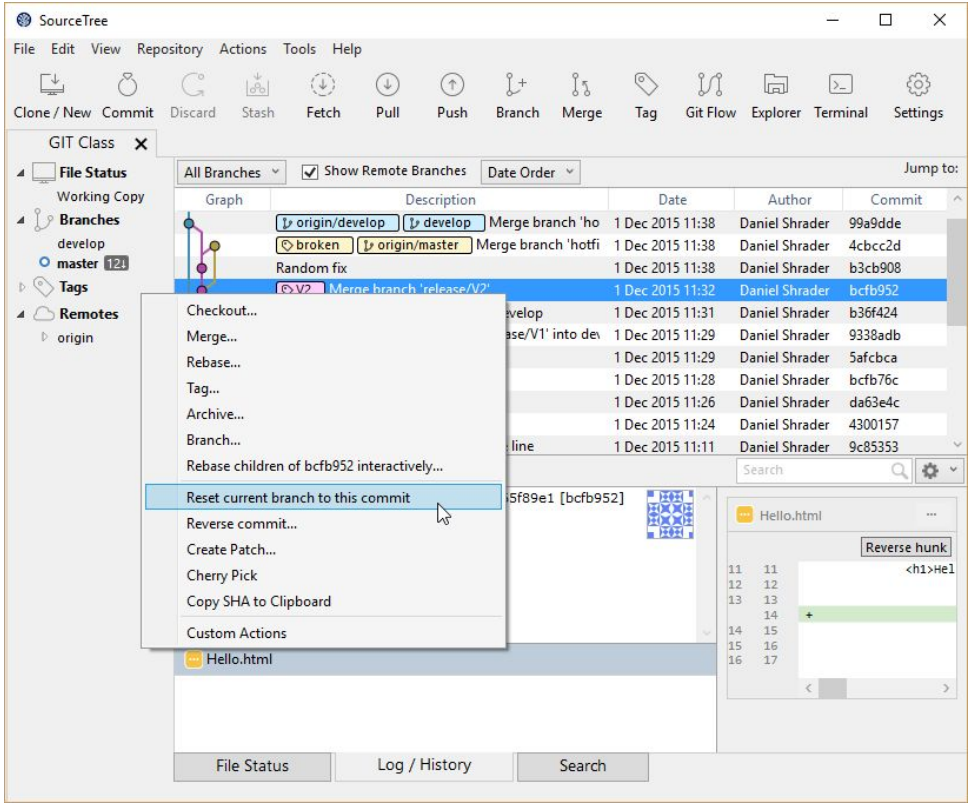
Pull



RESET

Lets you move around in time on your project. Really useful when looking for bug introductions.

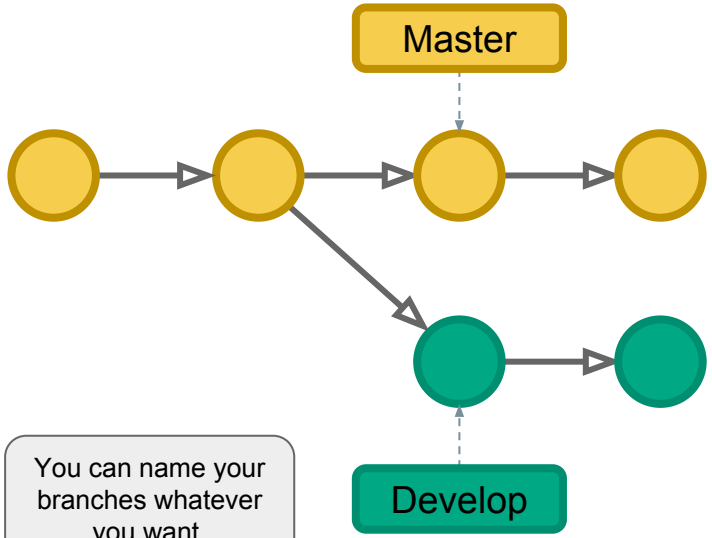




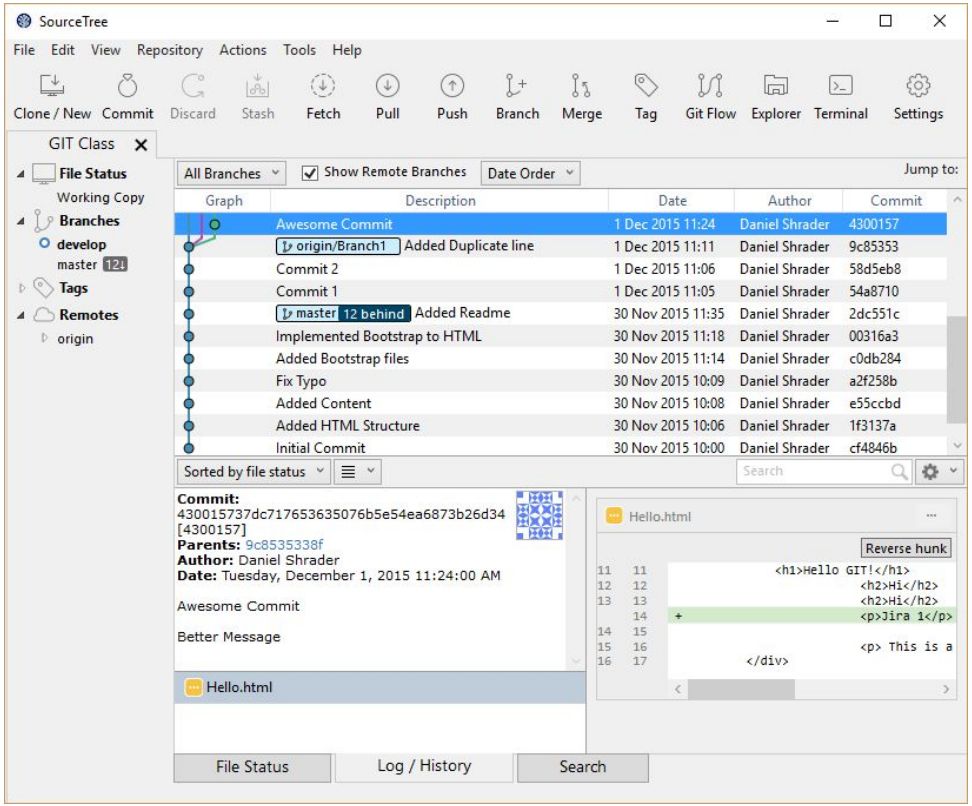
Branches!

BRANCH

A place to work which does not affect the main project.

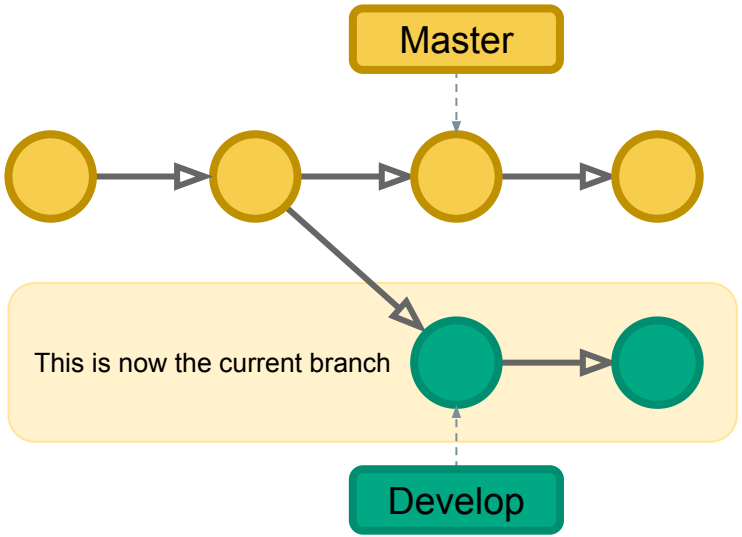


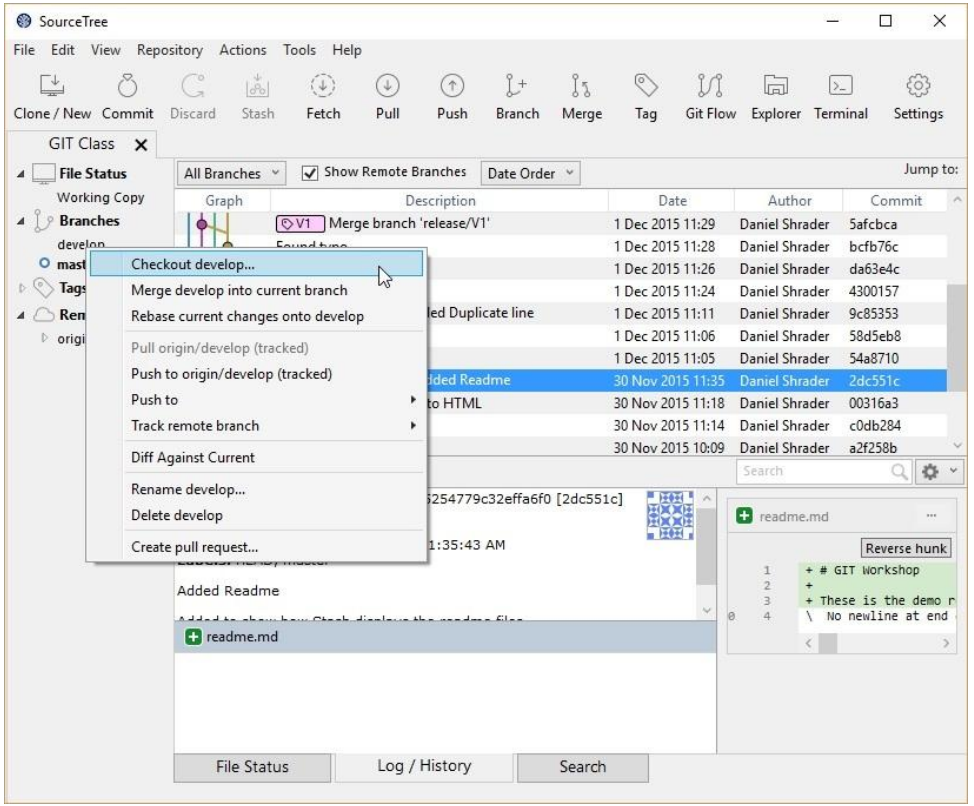
You can name your branches whatever you want.



CHECKOUT

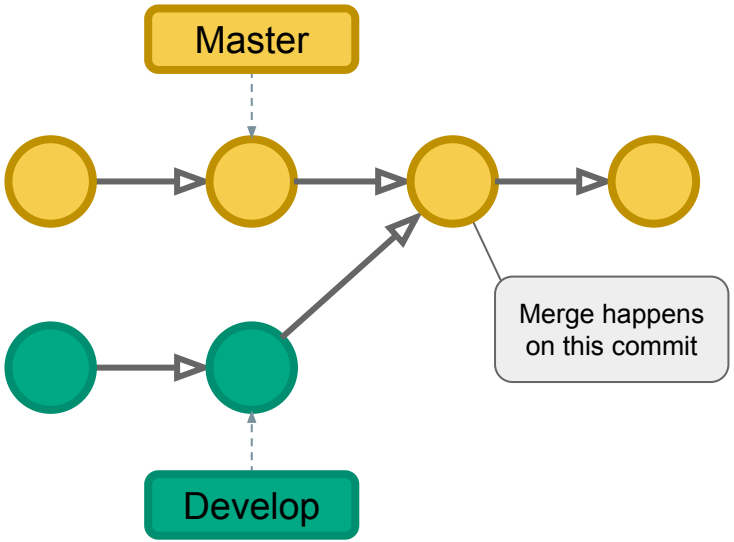
Switches the branch you are on.

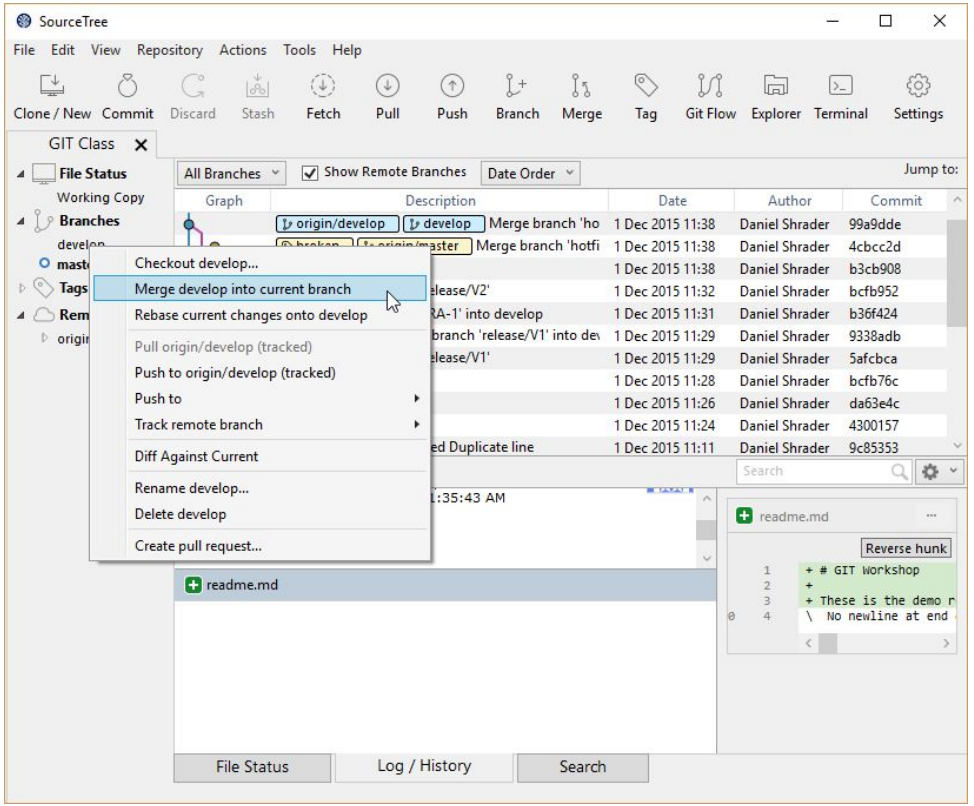




MERGE

Brings a branch into another. Be cautious of conflicts.



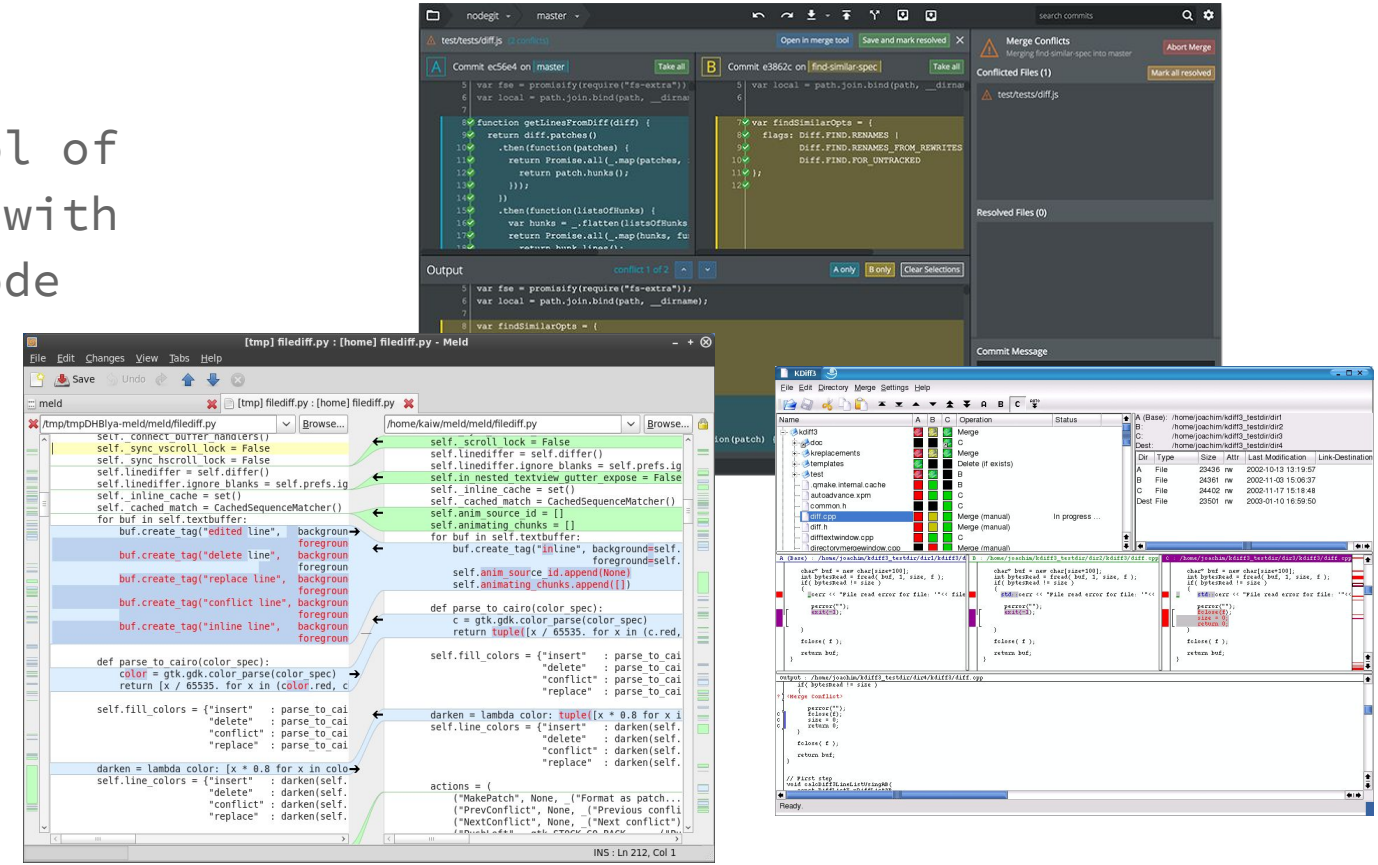


Mergetool - Conflicts

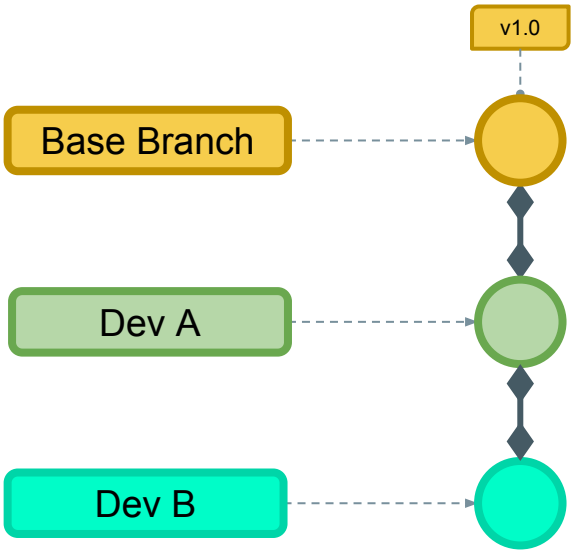
Simply your tool of choice to deal with the eventual code conflicts

A few are:

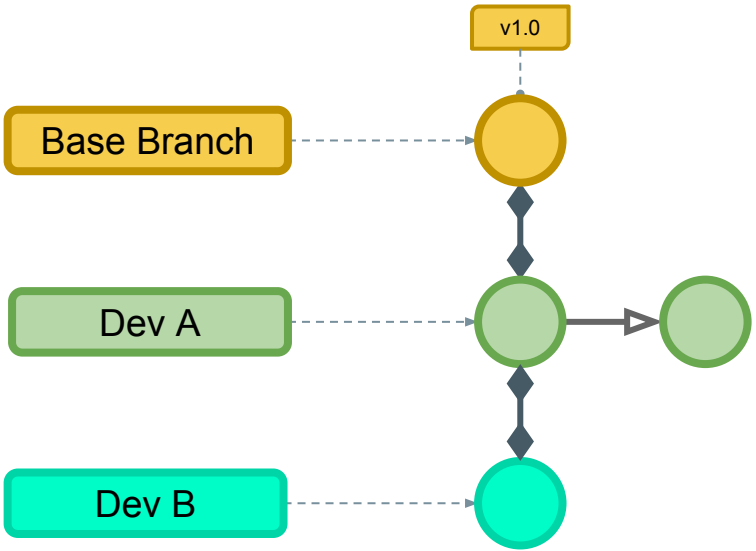
- meld
- kdiff3
- gitKraken



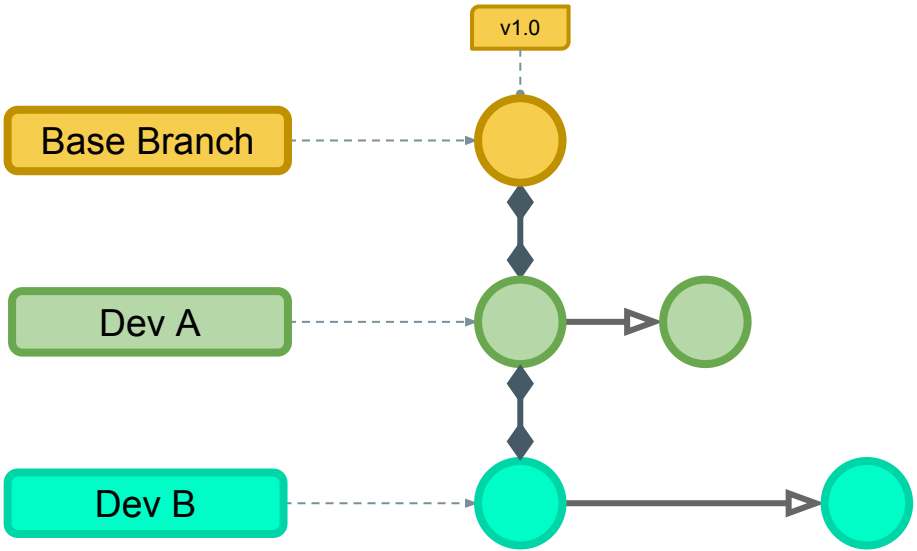
Merge - Conflicts



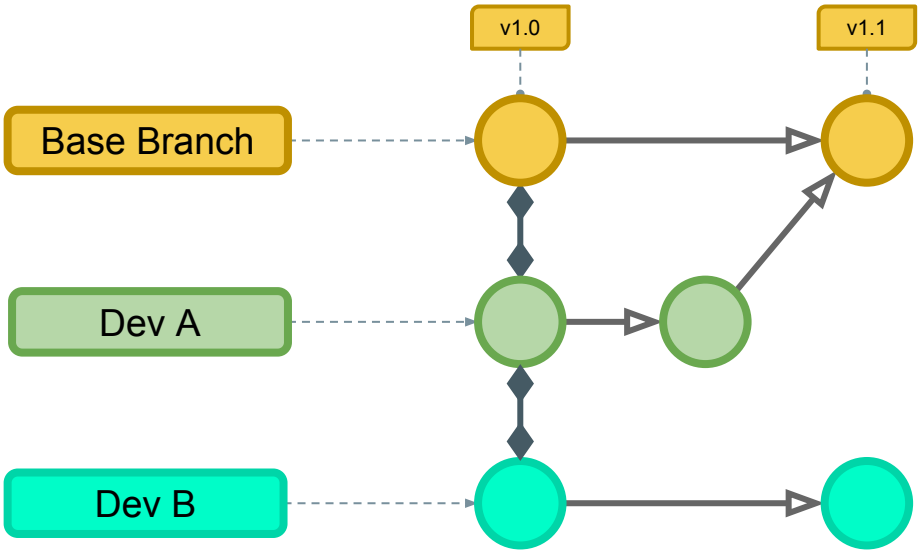
Merge - Conflicts



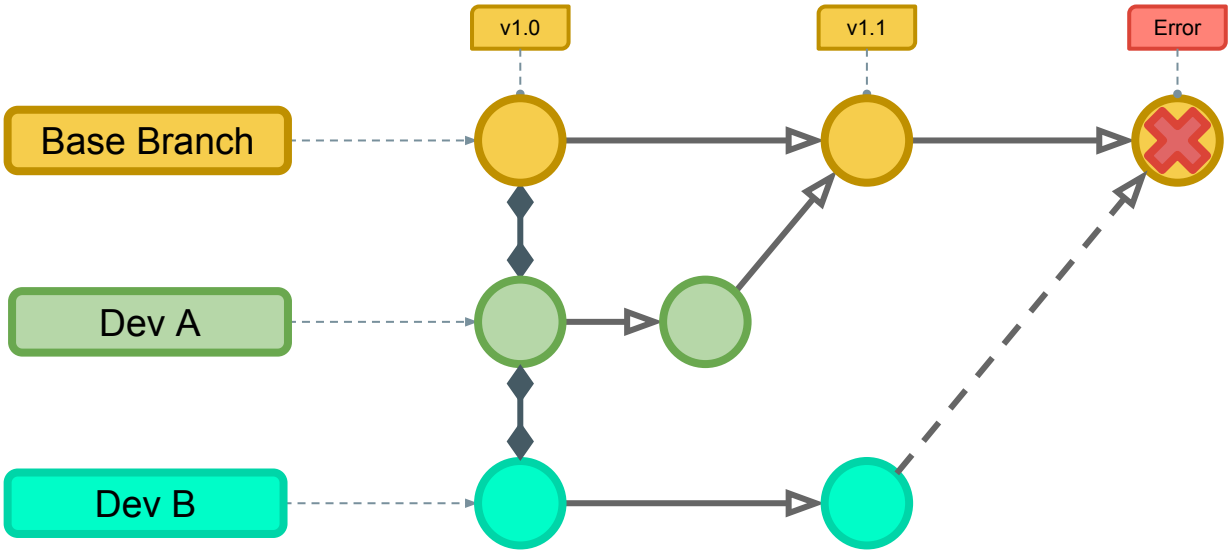
Merge - Conflicts



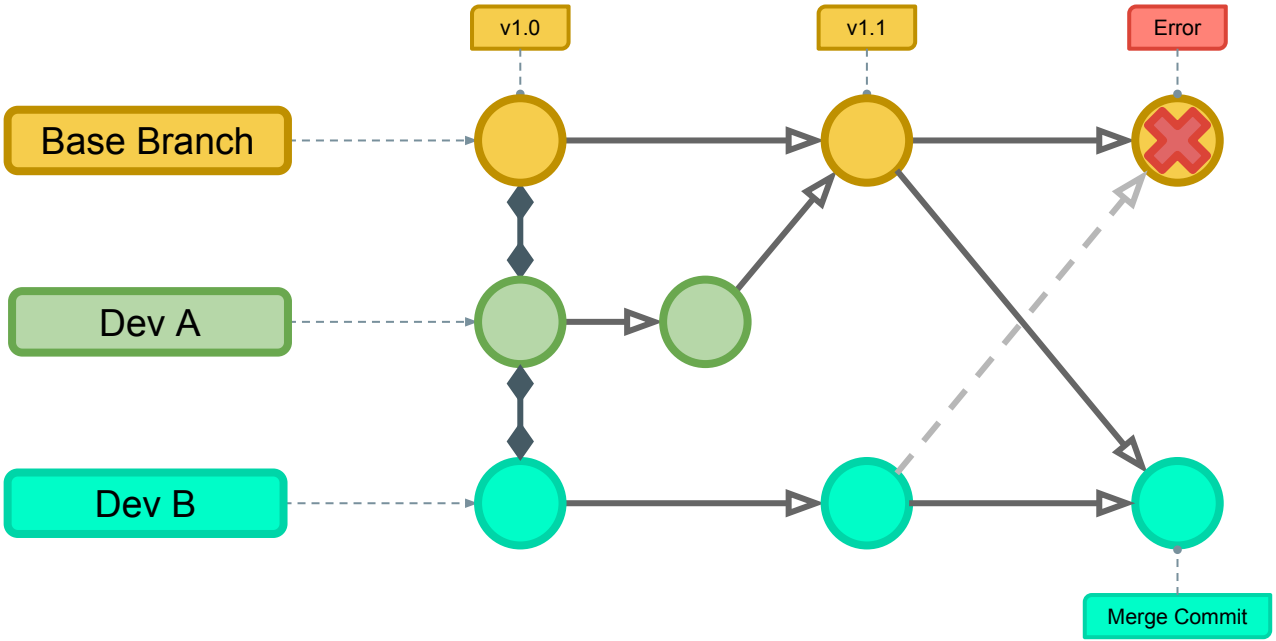
Merge - Conflicts



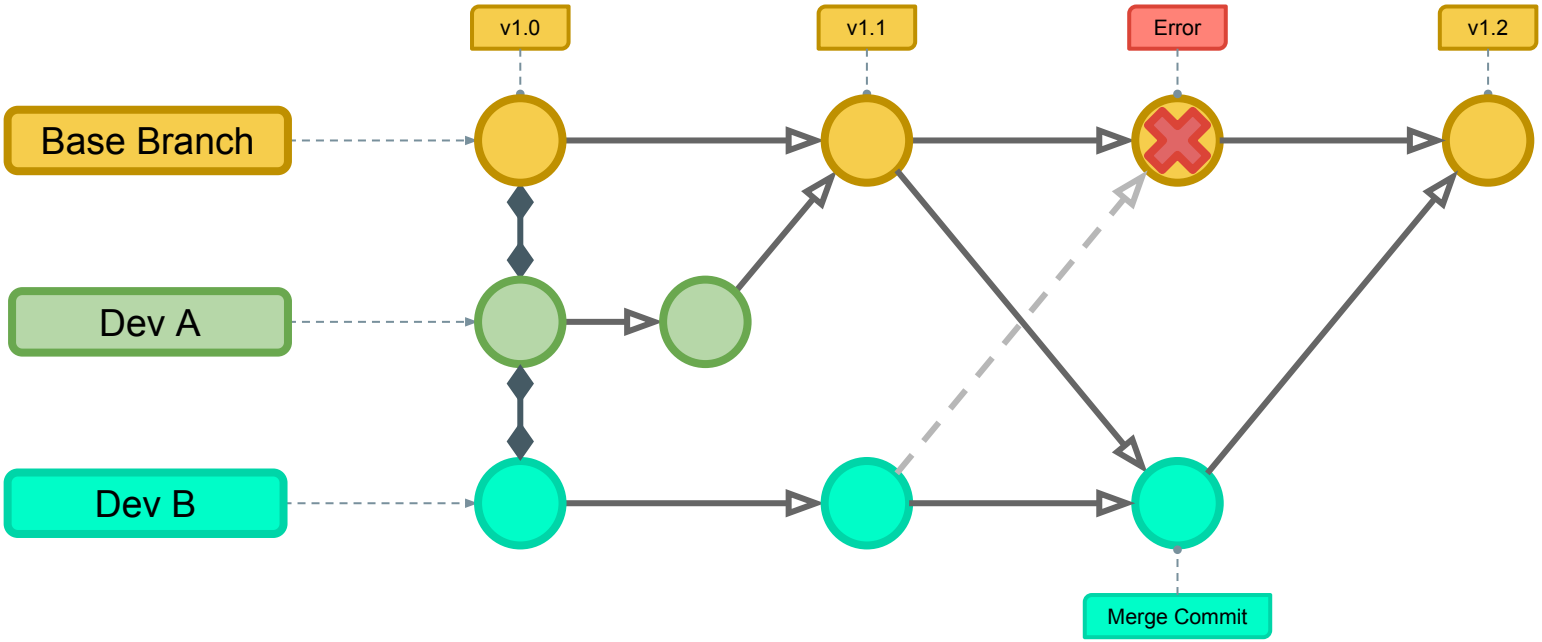
Merge - Conflicts



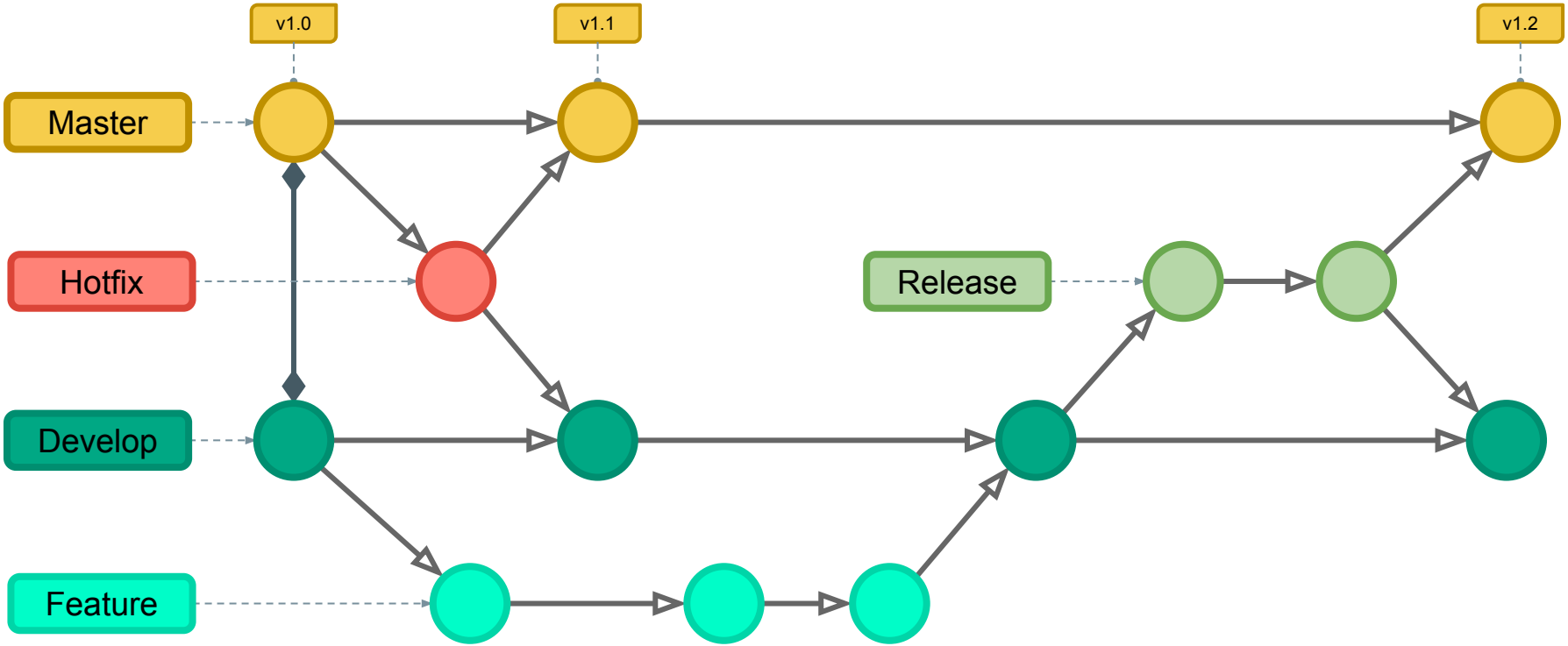
Merge - Conflicts



Merge - Conflicts



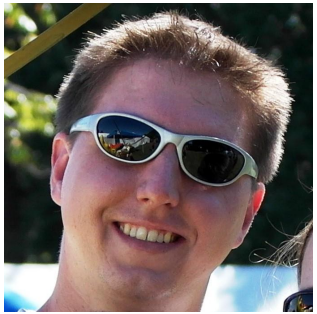
GIT FLOW WORKFLOW



Appendix

- https://www.atlassian.com
- http://www.mnu.edu/business/software-skills-demand
- https://xkcd.com/1296/
- https://xkcd.com/1597/
- http://danielkummer.github.io/git-flow-cheatsheet/
- https://training.github.com/classes/essentials/
- https://www.gitlab.com/
- https://github.com/
- https://www.sourcetreeapp.com/
- https://blog.axosoft.com/
- http://meldmerge.org/
- http://kdiff3.sourceforge.net/
- https://git-scm.com
- https://arstechnica.com/information-technology/2017/05/90-of-windows-devs-now-using-git-creating-1760-windows-builds-per-day/

About Me



I've been dabbling with code since the late 90's - currently work with Microsoft Business Intelligence products and Amazon Web Services. When given the chance, I write single page applications in JavaScript. I'm also a stickler for automating as much of the day to day tasks as I can. When I'm not coding I enjoy camping and hanging out with my wife and sons.

Email: Dan@ShraderLand.com