Fetch then prune.

When you clone, you clone everything, branches, features. All in the repo.

Pull will get everything on repo.

Pull will overwrite files.

Fetch will update files. Will notify you only.

Need to kill source after merging. Delete the whole branch.

Usually people just prune and pull.

Commit is saving to your local.

Can’t clone it if it exists. You don’t want to clone in a repo that you cloned before because its disorganized.

Git add <filename>

or

Git add .

Or

Git add \*.txt

Means all txt files.

. Means all files

Git add will be in the temporary container

Git commit -m

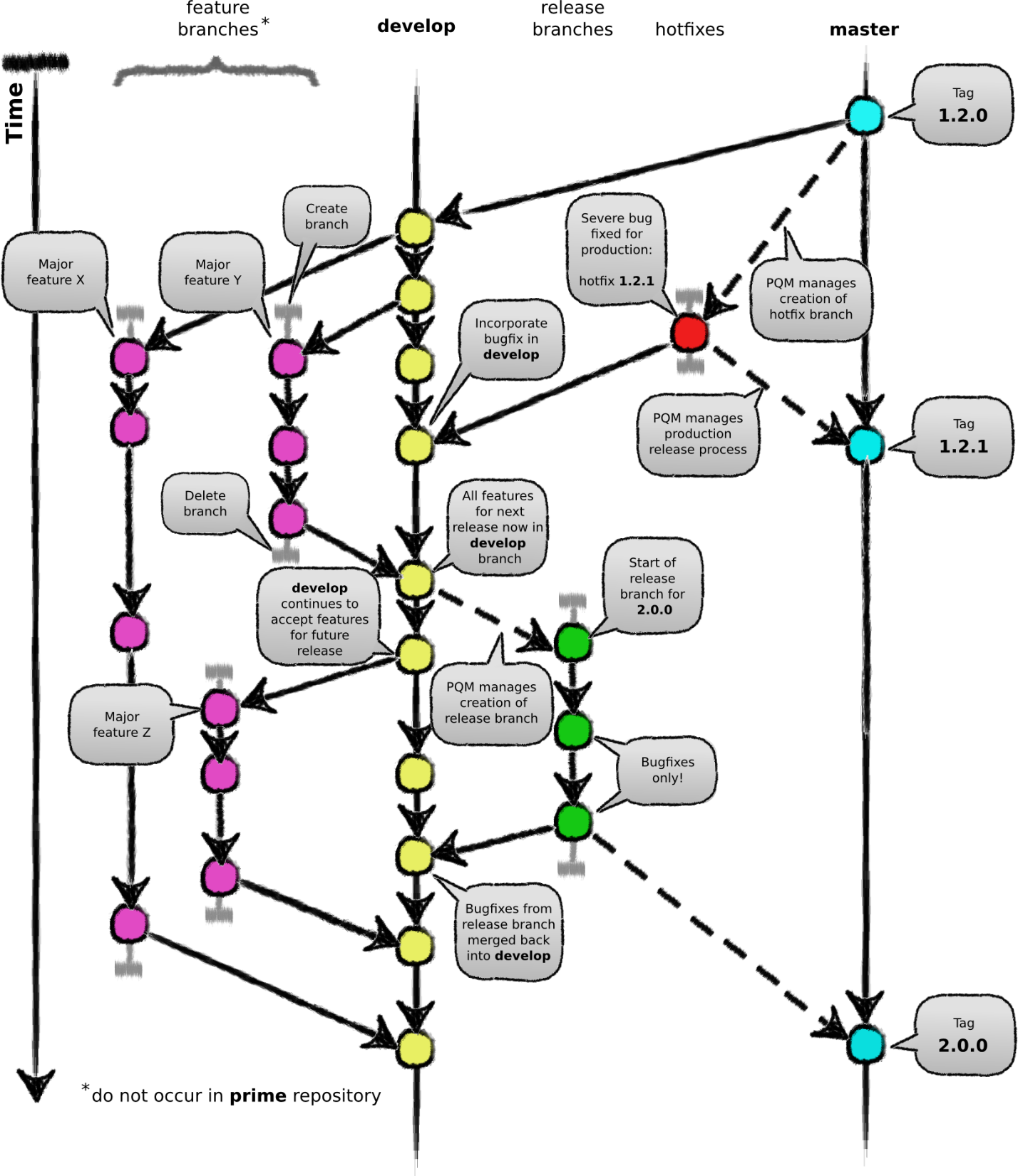
Saving

Then

Git push

Or

Git push origin <branchname>



View by patch or tree

Don’t do tree

Always do patch – can let you see changes.

Git commit -m ‘finished test’

Git reset --hard

Means all files that are changes/modified will be deleted.

Goes back to whatever the head is

git checkout -b <branchname>

to create new branch

git checkout master

go to master

git checkout is getting the branch

git pull --all

for organization sake

go to branchname then pull first.

changing the comment is hard when it is pushed

you can go back 1 commit

git reset --soft HEAD~1

~ (tailda)

on this commit, I want to go back 1 commit.

HEAD~1 it means 1 head below. i.e. UNDO

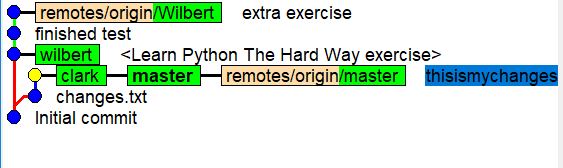
Normally you only do head 1 and not head 2

In real world, you can’t push to master.

Usually push to dev then merge to master.

If HARD

Everything you haven’t committed, gets erased.



Yellow and bold font is where you are currently at.

Remotes/origin is in the repo

Red line means it branched out