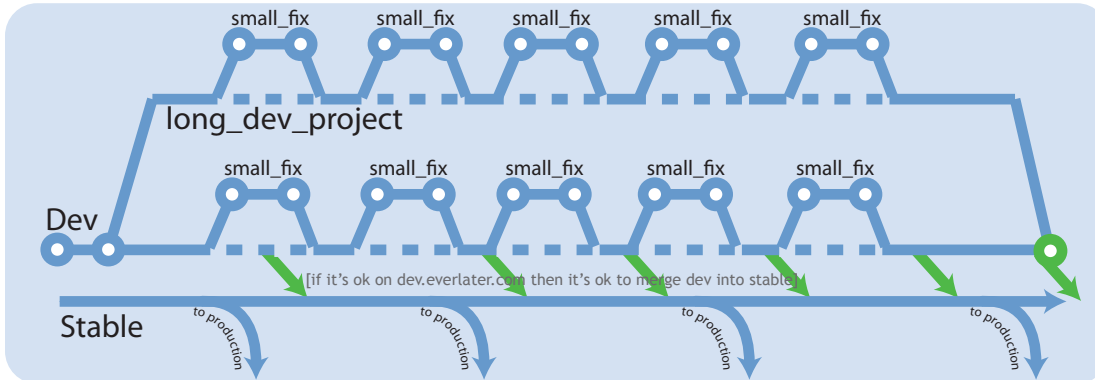


Everlater + git

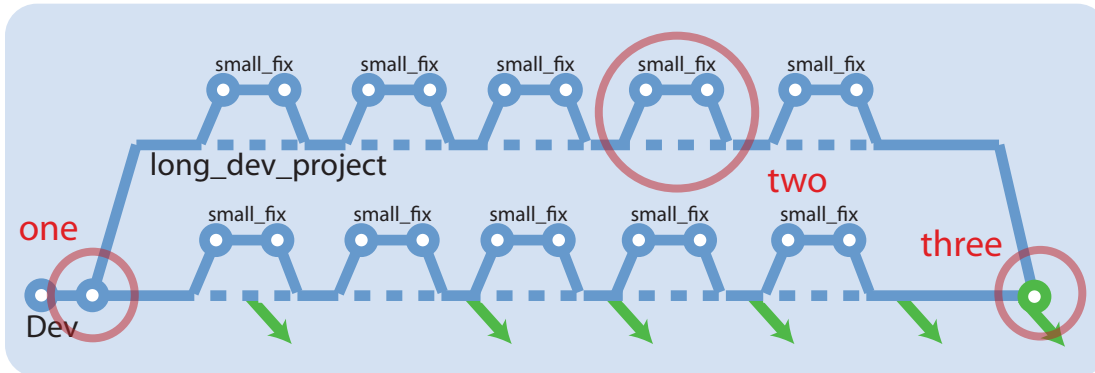
cheat sheet!

Theoretical Workflow



*Circles represent commits; Green denotes merge commits

Main Operations



Mirroring a new remote branch

```
$ git remote update
$ git checkout nate/long_dev_project
$ git checkout -b long_dev_project
$ git push origin long_dev_project
```

[Use this set of commands to mirror a new remote long-term development branch that someone else had created, in this example Nate created long_dev_branch using the instructions from ONE, and this user made a copy of that branch and pushed that copy to their remote repository. When long_dev_branch is merged into dev (eg. by Nate), then this user would run steps 5 and 6 of THREE but in step 5 use the -D flag instead of -d. This assumes that this user did not have any un-pushed changes on their long_dev_project branch.]

**NEVER REBASE SOMETHING
THAT EXISTS OUTSIDE
YOUR LOCAL REPO**

ONE [make a new branch and push it to your remote]

1. `$ git checkout -b long_dev_project`
[Creates the branch on your local machine, ENSURE YOU ARE IN DEV BRANCH to run this]
2. `$ git push origin long_dev_project`
[Creates the branch on your remote repository]

TWO [create a temporary work branch and then rebase it into the parent branch]

1. `$ git checkout -b small_fix`
[creates local branch; ensure you execute this within the correct branch (long_dev_project)]
2. `$ [make some commits]`
[Test and ensure that the code is working...of course!]
3. `$ git checkout long_dev_branch`
[Changes branch to long_dev_branch]
4. `$ git remote update`
[Check to see if anyone has made changes remotely to any branches]
5. `$ git merge _____` (dependent on 4)
[merge any changes that were fetched in 4 eg. `$ git merge nate/long_dev_branch`]
6. `$ git rebase long_dev_branch`
[Rebases changes from long_dev_branch to small_fix; DO THIS FROM small_fix branch]
7. `$ git branch -d small_fix`
[Deletes the temporary work branch small_fix]
8. `$ git push`
[pushes all updates to your remote repository on all branches]

THREE [merge and delete a branch and push these changes to your remote]

1. `$ git checkout dev`
[Changes to dev branch, make sure you have a clean git status before doing this]
2. `$ git remote update`
[Check to see if anyone has made changes remotely to any branches]
3. `$ git merge _____` (dependent on 2)
[merge any changes that were fetched in 4 eg. `$ git merge nate/dev`]
4. `$ git merge long_dev_project`
[Merges long_dev_project into dev]
5. `$ git branch -d long_dev_project`
[Deletes the branch locally]
6. `$ git push origin :heads/long_dev_project`
[Deletes the branch on github]
7. `$ git push`
[pushes all updates to your remote repository on all branches]