Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

Git Use Cases

Subtitle? pft...

Jake Tobak

Engineering and Computer Science Interest Group
URI Student ACM Chapter

6 March 2013

Outline

1 Single Developer Project

2 Multi Developer Project

3 Open Source Project

4 Workflow

Backup Code

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

- The Wrong Way
  Keep your source on Dropbox

Backup Code

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

- The Wrong Way
  Keep your source on Dropbox
- Why?
  For a single developer project, dropbox isn't a terrible alternative to Git for keeping a backup of your current source code, but Git is more than a server to keep a backup of your latest version of code.

Backup Code

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

- The Wrong Way
  Keep your source on Dropbox
- Why?
  For a single developer project, dropbox isn't a terrible alternative to Git for keeping a backup of your current source code, but Git is more than a server to keep a backup of your latest version of code.
- The Git Way
  Use Git!

Backup Code

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  Keep your source on Dropbox
- Why?
  For a single developer project, dropbox isn't a terrible alternative to Git for keeping a backup of your current source code, but Git is more than a server to keep a backup of your latest version of code.
- The Git Way
  Use Git!
- Why?
  Cause Git!

Restore Previous Version

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

You want to revert to a previous version of your code, back when everything actually worked.

- The Wrong Way
  Keep your source on Dropbox

Restore Previous Version

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

You want to revert to a previous version of your code, back when everything actually worked.

- The Wrong Way
  Keep your source on Dropbox
- Why?
  Dropbox does allow you to restore previous versions of your files for up to 30 days, or even longer if you subscribe to their Packrat service, but it's only available to Pro users (starting at $10/month) and costs $40/year on top of that. It also doesn't give you a summary of the changes between each version other than the date and user who made them.

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wronger Way
  Keep multiple versions of your code saved locally

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wronger Way
  Keep multiple versions of your code saved locally
- Why?
  This can get real messy and real confusing real quick. There's also no explanation of what was changed in each version or a clear chain of progression.

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wronger Way
  Keep multiple versions of your code saved locally
- Why?
  This can get real messy and real confusing real quick. There's also no explanation of what was changed in each version or a clear chain of progression.
- The Wrongest Way
  Reopen the source in your editor and hold CTRL+Z until the buffer runs out.

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wronger Way
  Keep multiple versions of your code saved locally
- Why?
  This can get real messy and real confusing real quick. There's also no explanation of what was changed in each version or a clear chain of progression.
- The Wrongest Way
  Reopen the source in your editor and hold CTRL+Z until the buffer runs out.
- Why?
  How far back does your undo buffer go? Will it go back to a working state or stop short? How do you know when you got to that spot where everything is working and not a couple places before/after you changed that = to a ==?

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  git commit and **git push** every time you make a significant change. Then **git log** and **git checkout** to revert to a previous version.

Restore Previous Version (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  git commit and **git push** every time you make a significant change. Then **git log** and **git checkout** to revert to a previous version.
- Why?
  git commit stages all the changes you made to your code since the last update and lets you give a brief explanation of what has changed since then. **git push** uploads the staged changes to your git server.
  git log allows you to view your previous commits with an explanation of what you did in each one. You can use **git checkout** to download a copy of a specific version to your computer.

Working on different files

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

- The Wrong Way
  Everyone on the team e-mails each other their latest source files and tells everyone what they'll be working on.

Working on different files

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  Everyone on the team e-mails each other their latest source files and tells everyone what they'll be working on.
- Why?
  This will take forever if you have to wait for someone to e-mail you a piece of code you need and it's very easy to have branches that are extremely divergent to each other.

Working on different files

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  Everyone on the team e-mails each other their latest source files and tells everyone what they'll be working on.
- Why?
  This will take forever if you have to wait for someone to e-mail you a piece of code you need and it's very easy to have branches that are extremely divergent to each other.
- The Also Wrong Way
  Share a Dropbox account

Working on different files

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  Everyone on the team e-mails each other their latest source files and tells everyone what they'll be working on.
- Why?
  This will take forever if you have to wait for someone to e-mail you a piece of code you need and it's very easy to have branches that are extremely divergent to each other.
- The Also Wrong Way
  Share a Dropbox account
- Why?
  This will work, but if you're both working at the same time and another developer saves a change to test it, you might sync a broken file and start getting errors for something someone else did. Have fun debugging that.

Working on different files (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  Just do it!

Working on different files (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  Just do it!
  When you **push** to the git server, you're uploading your changes, but you can also **pull** the changes from the server that have been made since the last time you sync'd with it.

Working on different files (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  Just do it!
  When you **push** to the git server, you're uploading your changes, but you can also **pull** the changes from the server that have been made since the last time you sync'd with it.
- Why?
  If you're working on the same **branch**, git will **merge** your changes and any changes other people have made since your last **push/pull**. Git will attempt to automatically combine everyone's changes in a way that won't break anything. Git is really clever.

Working on the SAME files

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

- The Wrong Way
  I don't even know, just don't.

Working on the SAME files

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  I don't even know, just don't.
- The Git Way
  Do it, you won't do it!

Working on the SAME files

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  I don't even know, just don't.
- The Git Way
  Do it, you won't do it!
- Why?
  Seriously, Git is REALLY clever when it comes to merging. I'm sure you could break it if you wanted to, but if you don't **push** broken code or really redonkulous changes, everything will probably be fine.

Making Radical Changes

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  I would hope that you would know better than to not use Git by now, but there are wrong ways to use Git too.
  Just continue with business as usual using Git.

Making Radical Changes

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Wrong Way
  I would hope that you would know better than to not use Git by now, but there are wrong ways to use Git too. Just continue with business as usual using Git.
- Why?
  If you're adding a new feature or something big, you might need to **commit** and **push** multiple times before your code is working perfectly again. If anyone does a **pull** before you're done, they'll have incomplete code merged into their source and it'll cause a lot of headaches. It's also possible that you only **push** working code, but maybe someone had a better solution that your team will go with instead and you'll just be competing against each other every time you **push**.

Making Radical Changes

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

- The Git Way
  branch! You can use **git branch** to create a new **branch**. A **branch** is a chain of **commits** that diverges from the current history. When you **push** to a new **branch**, it won't be merged when someone **pulls** from the original. When you're satisfied with your code, you can **pull** your code from the new **branch** into the original and Git will attempt to **merge** everything for you.

Contributing to Open Source

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

Open Source Projects are Multi Developer Projects, but you don't have write access! How can you contribute while having read-only permission?

Contributing to Open Source

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

Open Source Projects are Multi Developer Projects, but you don't have write access! How can you contribute while having read-only permission?
The solution is **git clone**. When you **clone**, you are creating a copy of the **repository** (repo) that includes all the code, history and branches, but this copy belongs to you and you can do whatever you want with it. **commits** you make will be completely independent of the original source. **forking** is GitHub's version of **clone**, but it creates the repo under your GitHub account.

Contributing to Open Source (cont.)

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

Okay, so now you can edit the code, but you're still doing it in your own **repo**. The original project is unaware of your changes, so what good are ya?

Contributing to Open Source (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

Okay, so now you can edit the code, but you're still doing it in your own **repo**. The original project is unaware of your changes, so what good are ya?
The solution is **git request-pull**. **request-pull** generates a summary of the changes you made and a URL to find them. You can then post this to a mailing list or send it to the person incharge of the original project. If they like what they see, they can **pull** from your code and **merge** it into the original project. You can also generate a **patch** that is a list of the changes between your version and the original version, which they might prefer.

Contributing to Open Source (cont.)

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow

Another solution is GitHub's **Pull Request**. It's similar to the native **request-pull** and **patch** process, but it's much more interactive and can be managed from a web interface. If an Open Source Project is hosted on GitHub, then this is probably how they want you submit your changes.

Contributing to Open Source (cont.)

Git Use Cases
Jake Tobak
Single Developer Project
Multi Developer Project
Open Source Project
Workflow

Another solution is GitHub's **Pull Request**. It's similar to the native **request-pull** and **patch** process, but it's much more interactive and can be managed from a web interface. If an Open Source Project is hosted on GitHub, then this is probably how they want you submit your changes.

The best way to find out how you can contribute to a project is to join its IRC channel and/or mailing list.

General Workflow

Git Use Cases

Jake Tobak

Single Developer Project

Multi Developer Project

Open Source Project

Workflow