I’m a fairly happy Git user :-)  It’s and excellent product and a great improvement

on CVS, SVN, SourceSafe, PVCS and other systems I have used in the past.

In my day to day working I’ve got into a pattern of using very simple commands like clone, commit, update, pull, push etc.

But then I got into the situation where my code was broken but I couldn’t pinpoint exactly what had gone wrong or when this happened. What I wanted to do was to roll the code back until I reached the time when the code was last working which would allow me to determine the exact cause of the failure.

The question was how to go back in history using just the date:

Turns out that this is really easy: Just use t and retrospective tagginghe command below:

|  |  |
| --- | --- |
| 1 | git checkout master@{2009-07-27 13:37} |

If this fails because the reflog doesn’t go back to that point in time then you need the command below:

|  |  |
| --- | --- |
| 1 | git checkout `git rev-list -n 1 --before="2009-07-27 13:37" master` |

Another useful thing to do is to be able to tag a build retrospectively. So for example in our scenario, let’s say you’ve found the last working version you might want to tag it.

That’s also pretty easy.

Use the command:

|  |  |
| --- | --- |
| 1 | git log |

to show a history of all the commits on the project.  Once you have found the commit at which you want to tag just use this command:

====== Jonathan Dutra

|  |  |
| --- | --- |
| 1 | git tag -a v0.9 **7fcea889f** -m "Retrospective tag" |

**7fcea889f** corresponde ao início da identificação do commit

Após isso você pode enviar usando :

|  |  |
| --- | --- |
|  | git push --tags origin master |

Com isso você tem uma tag que irá facilmente permitir que qualquer pessoa para revise as versão do código.