Connexer Ltd.

Technology Consulting

How `Git` Changed My Life (as a Software Engineer) and How It Can Change Yours

Roberto C. Sánchez

Ohio LinuxFest

Saturday, October 3, 2015

Overview

- Why Am I Giving This Talk?
- About Me
- A Story About How I Came to Understand and Use the Power of Git

Why Am I Giving This Talk?

- I was limited by my own short-sightedness
- I tried reading documentation but “I didn't get it”
- I hope that my experience will help others realize that there is untapped power nearby

About the Presenter

- I am a software engineer and consultant
- I have been programming since 1990
- I run my own one-man consulting firm
- I teach Computer Sci/Eng at Wright State
- I am online here:
  - http://people.connexer.com/~roberto
  - http://www.connexer.com
  - http://www.linkedin.com/in/robertocsanchez/

Stage 1: What is Version Control?

Version control is change management

Stage 2: Why Do I Need It?

It turns out that developing software without managing your changes is really hard

Stage 2b: What Should I Be Using?

In college (ca. 2002), as part of a team project,
I was introduced to CVS

Stage 2c: How Do I Use It?

- Step 1: Check out/update a working copy
- Step 2: Do some work
- Step 3: Check in your changes
- Step 4: If necessary, revert to a previous state (e.g., to work on a bug on an older release, or because you want to abandon your changes)

Stage 3: Tell Me About CVS, SVN,...

- They are centralized (i.e., client-server)
- The larger and more active your team is, the more likely you are to step on each other
- Working from a plane/train/automobile is :-(
- The working copy doesn't have everything
- CVS: can't rename files, remove directories
- SVN: Wastes lots of disk space

Stage 3b: Tell Me More

For all the limitations, centralized version control is better than no version control

Stage 3c: Are There Better Options?

Yes: Git, Mercurial, etc., don't waste space, don't lose history, and don't require network connectivity all the time

Stage 4: Making the Switch

- Step 1: Install Git (or Mercurial, or whatever)
- Step 2: Read quick reference (svn co == git clone; svn up == git pull; svn ci == git commit)
- Step 3: Press on as before

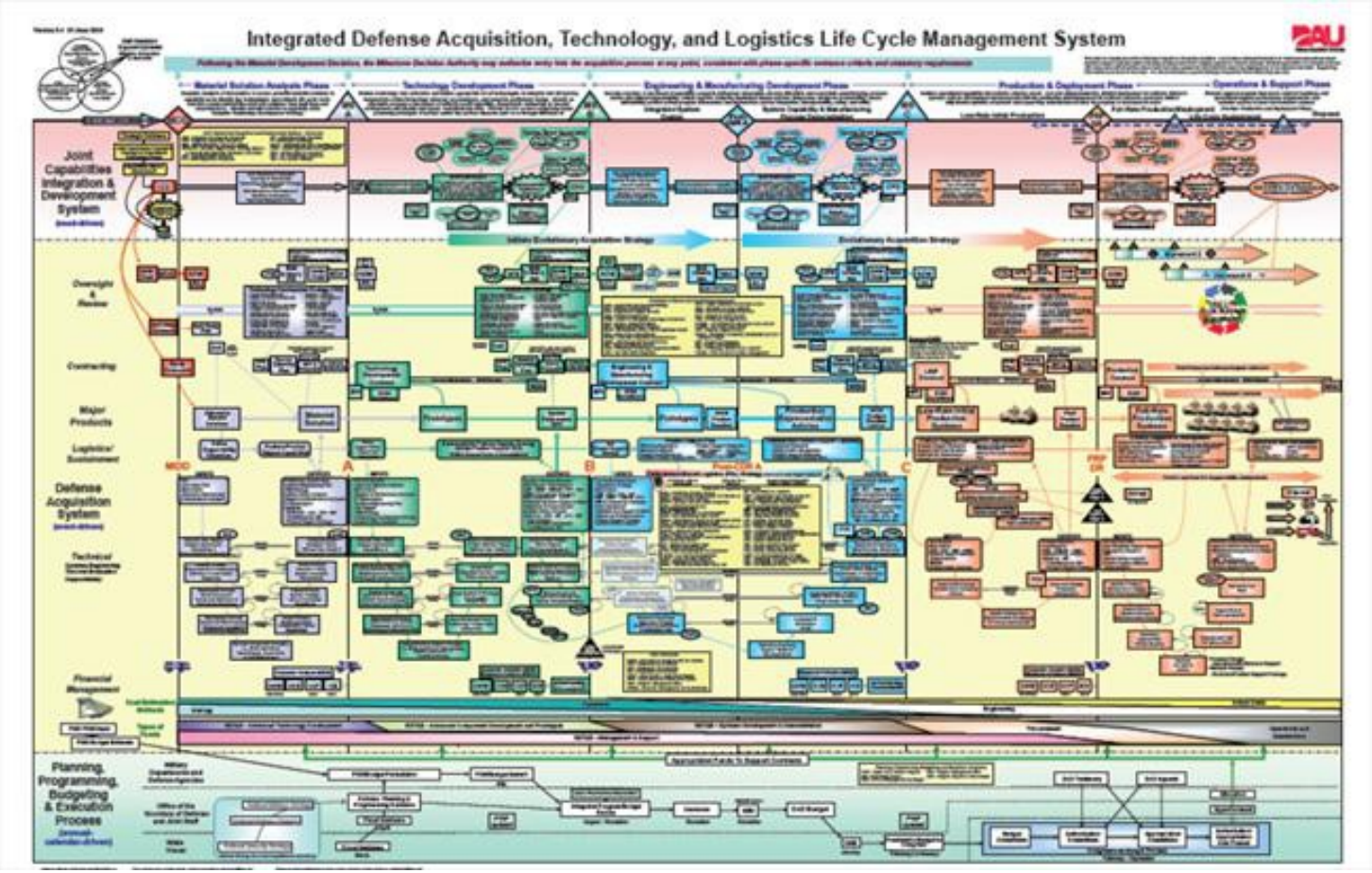
Stage 4b: Something Was “Off”

People around me (especially on open source projects) were doing something called branching and merging; I didn't do it because everybody said “don't do this with CVS/SVN”

Stage 4c: What Changed?

Git & co. had made branching/merging simple

Stage 5: How I Saw “Simple”



Public Domain image from http://acc.dau.mil

Stage 5b: My Reaction

I stuck to my guns and kept using Git exactly the same way I had used CVS and SVN

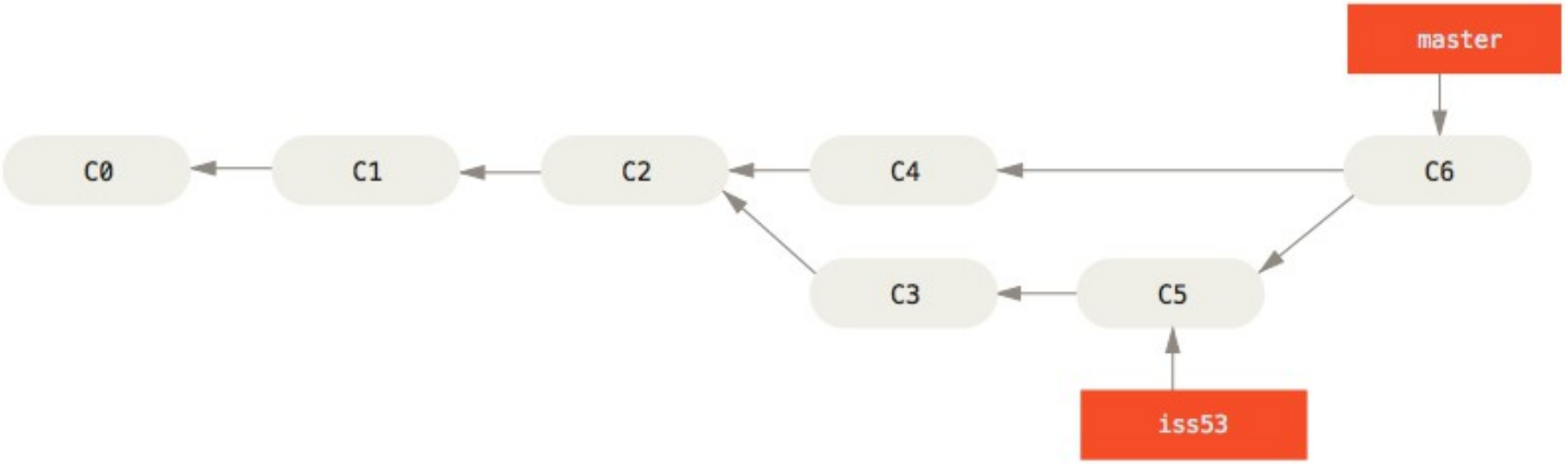
Stage 6: The Road to “Recovery”

As I saw more people switch to Git and more people talking about and doing this branching/merging, I realized that I must be missing out on something

Stage 6b: A New Way of Thinking

Pro Git by Scott Chacon and Ben Straub

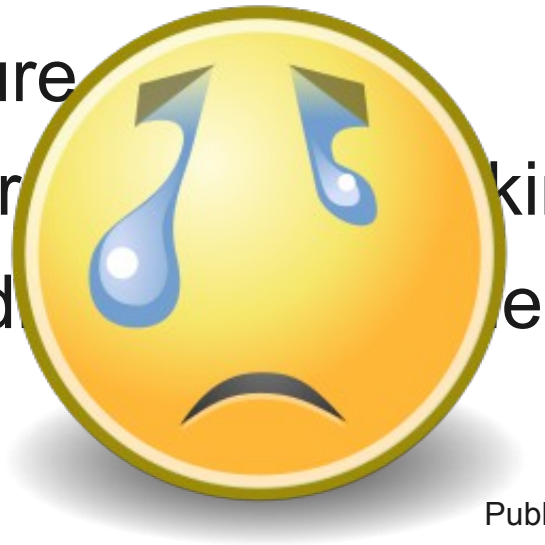
Stage 6c: How Simple Really Looks



cc-by-nc-sa-3.0 image from Pro Git, 2nd Ed.,
by Scott Chacon and Ben Straub (Figure 3-17)

Stage 7: A Changed Life!

- Scenario:
  - Work on a feature
  - Other developer[illegible]king on features
  - Eventually the d[illegible]eed to integrate



Public Domain image from http://openclipart.org

Stage 7b: Flexible Development

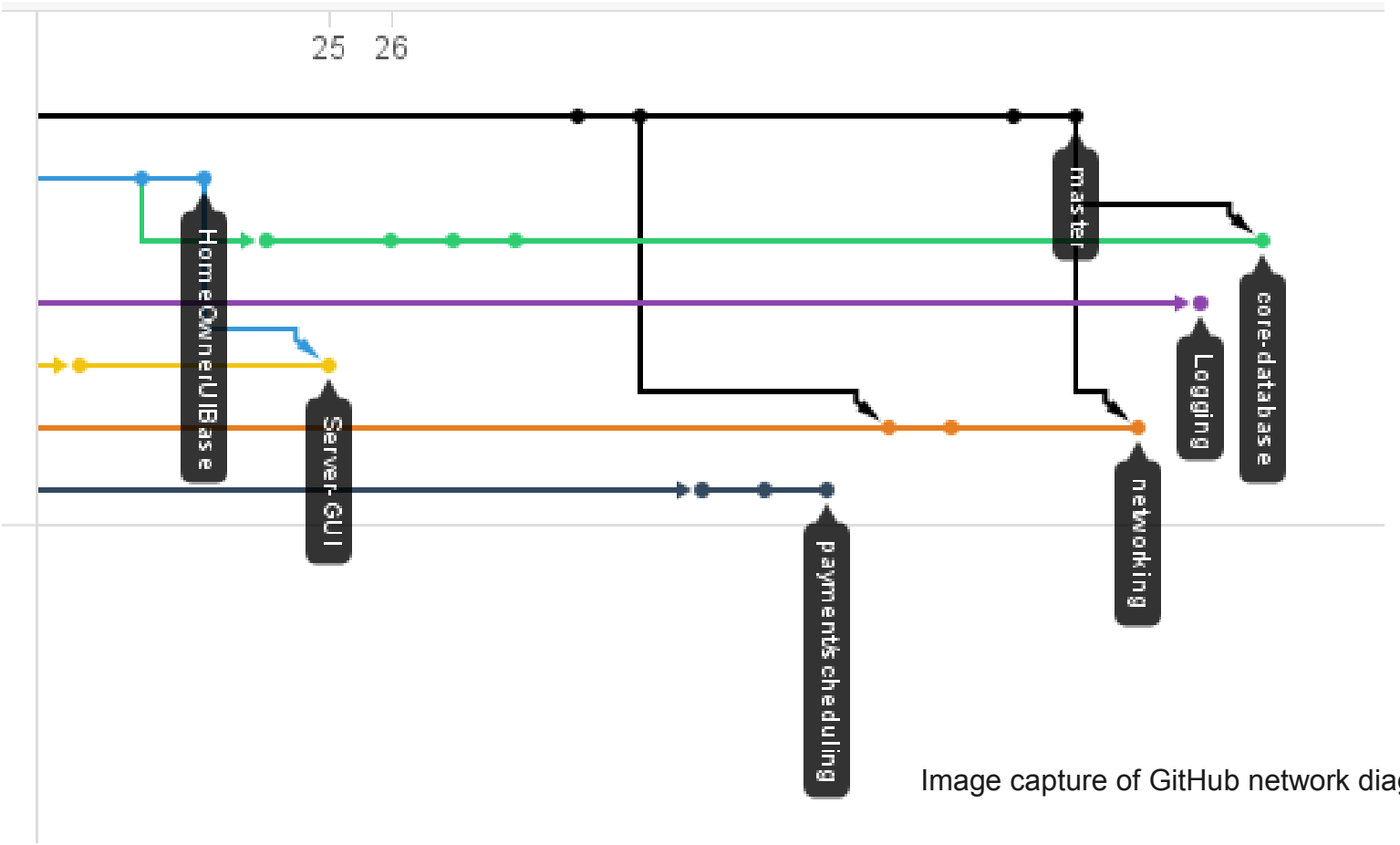


Image capture of GitHub network diagram

Stage 7c: What Really Changed?

- The “central” repo is the one we designate
- Every working copy (i.e., clone) is complete
- Team collaboration is much richer
  - No more copying the working dir to experiment
  - Work can be merged across team boundaries before being mainlined
- Losing history became close to impossible

Stage 7d: How Was This Better?

- I was more productive
  - Experimental work on short-lived branches
  - Working with other developers' experimental changes (not on mainline)
- The teams I was part of were more efficient
- It was easy to answer questions like “was commit abc1234 in the last point release?”

Summary

- Anything (e.g., CVS) is better than nothing
- A powerful tool (e.g., Git) can be misused
  - I can still drive a nail with the butt of a power drill
- Learning and understanding how to properly use a powerful tool is an investment

Questions?