Vershon control

(using git)

Danny Awty-Carroll

May 14, 2018

Sections

This will focus on using git in windows with a UI

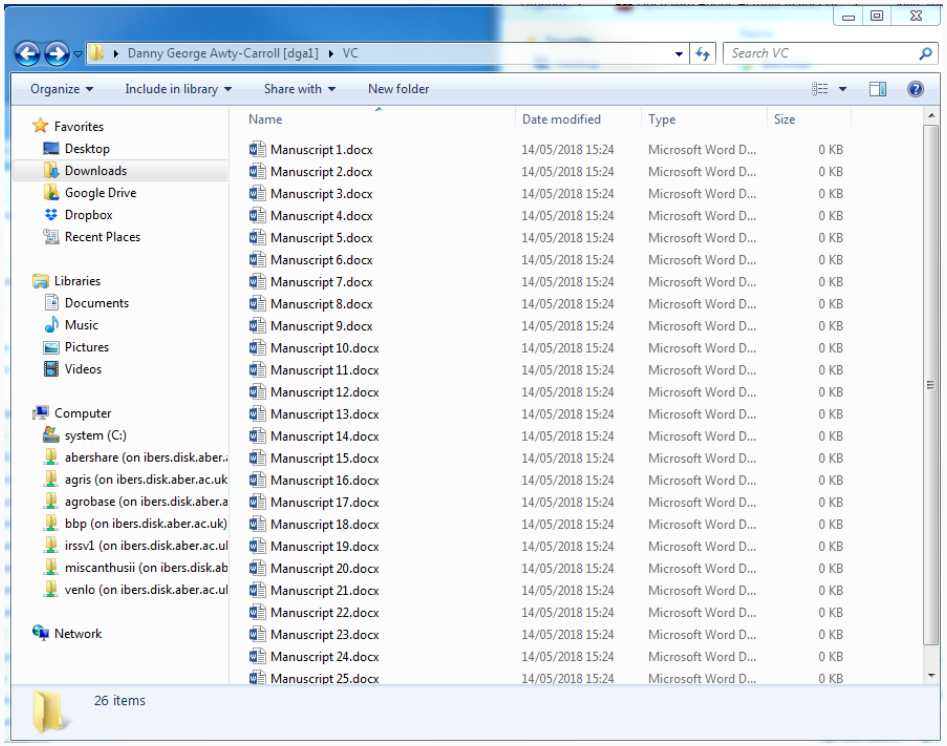
1. Why version control?
2. Version control system
3. Platform
4. Conclusions

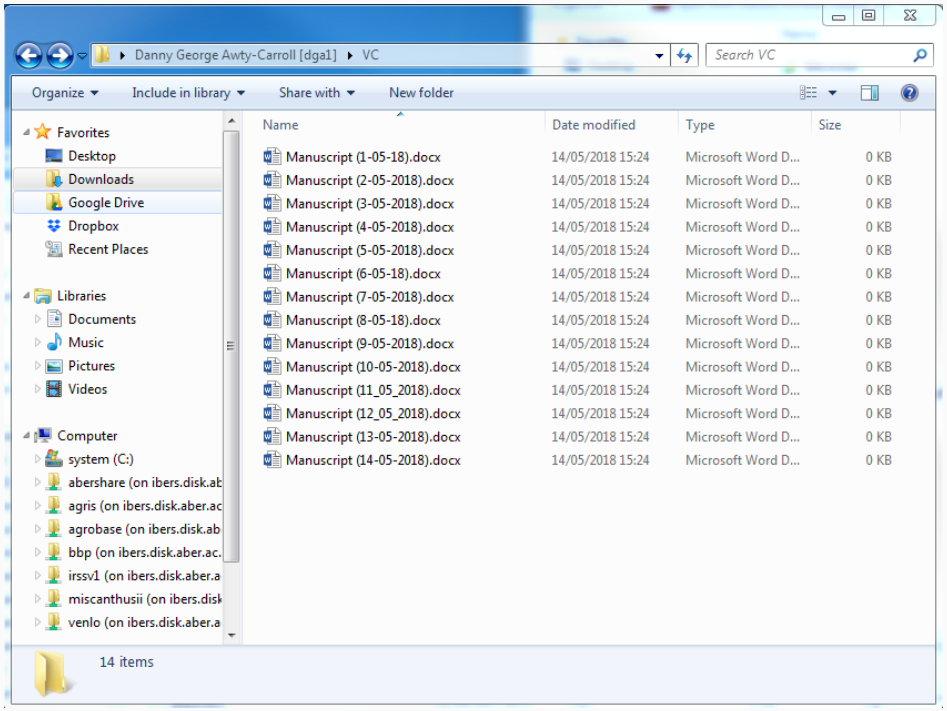
Why version control?

What is version control

This is just the manigment of vershons of a document.

One document through time.





All of us use some version control

Where things get complicated

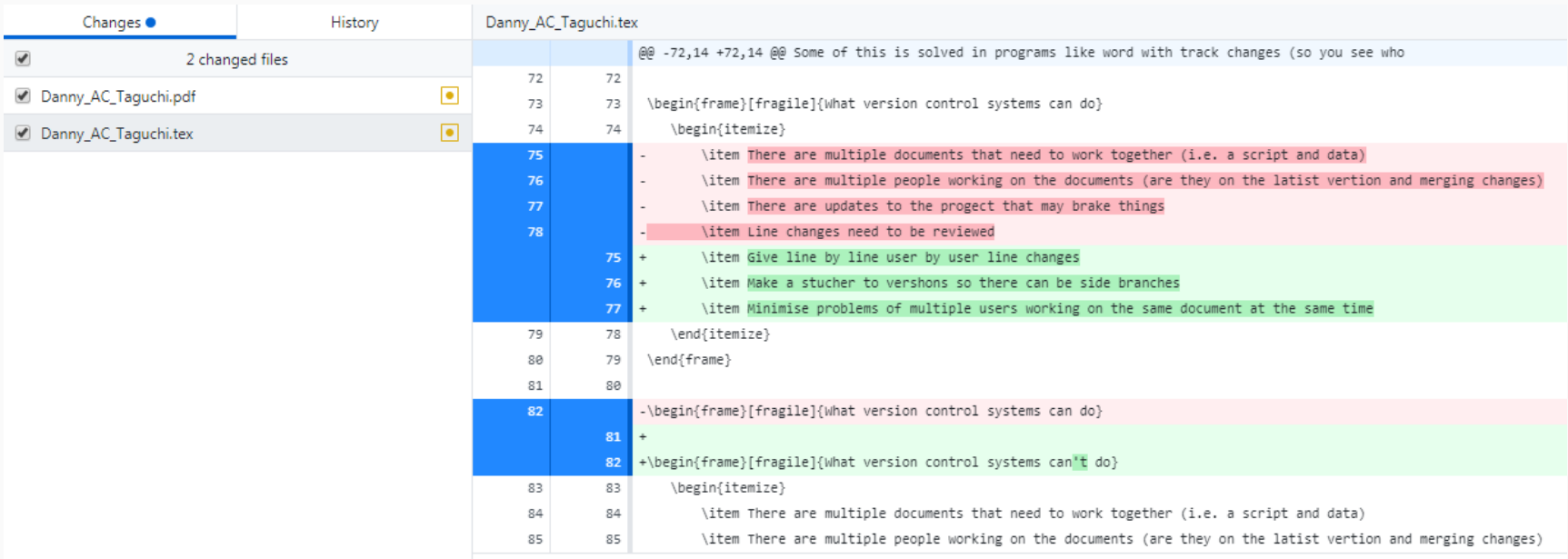
Numbering or dating documents works OK but can fall down when

- There are multiple documents that need to work together (i.e. a script and data)
- There are multiple people working on the documents (are they on the latist vertion and merging changes)
- There are updates to the progect that may brake things
- Line changes need to be reviewed

Some of this is solved in programs like word with track changes (so you see who altered what when)

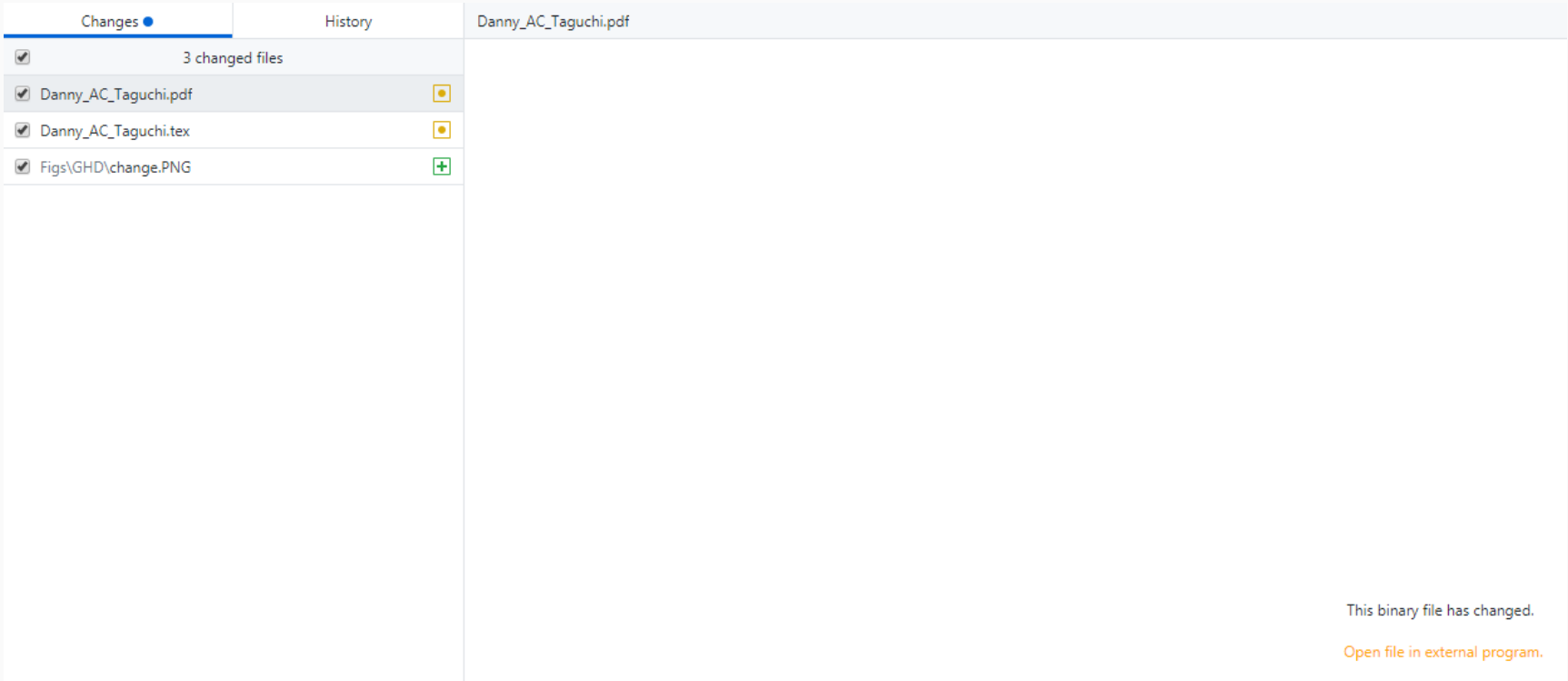
What version control systems can do

- Give line by line user by user line changes
- Make a stucher to vershons so there can be side branches
- Minimise problems of multiple users working on the same document at the same time



What version control systems can't do

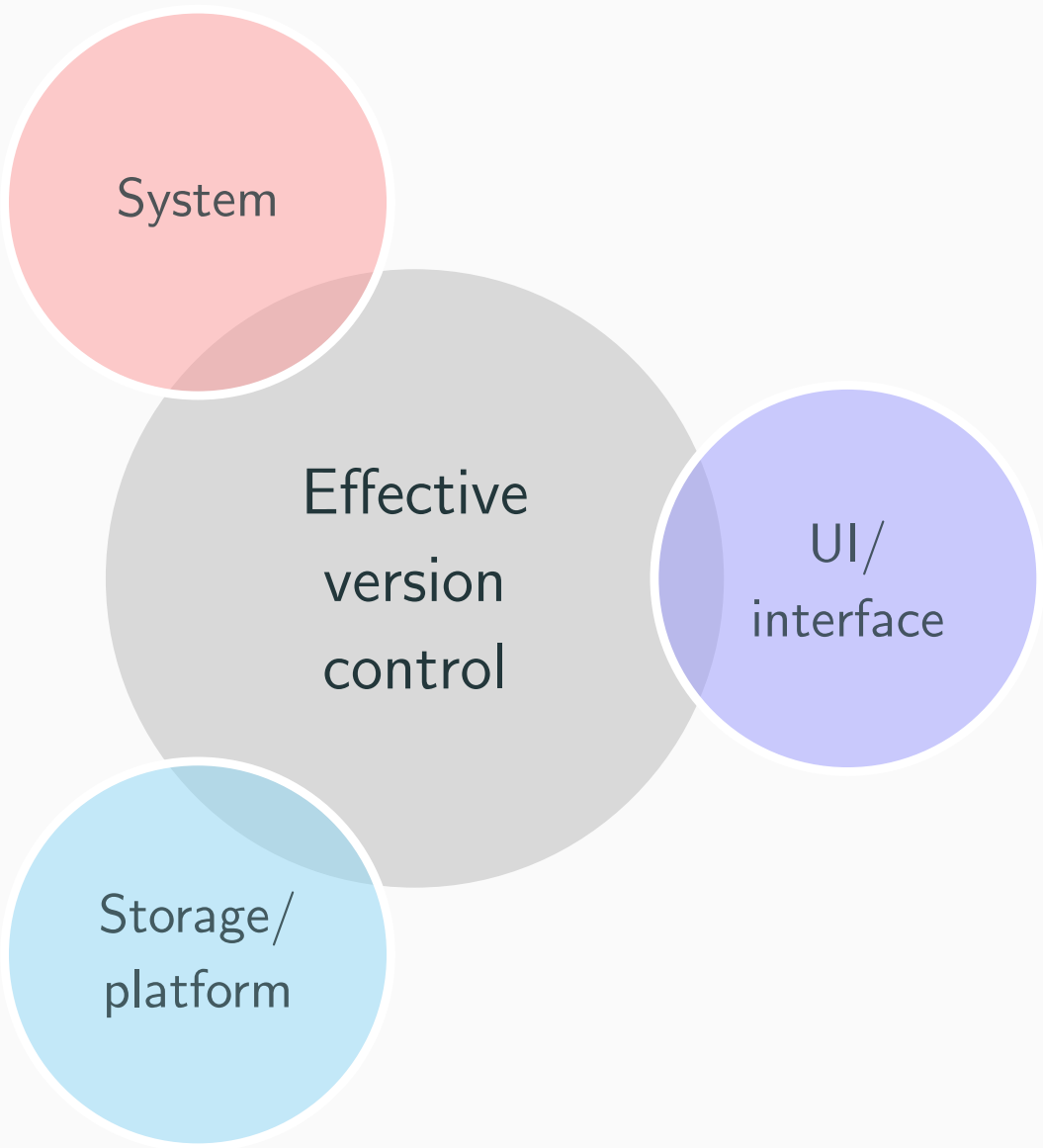
- Add much extra control to binary files (not plan text)
- Back up in real time



Version control system

What version control system?

The first thing about version control systems is which system (we will cover git)



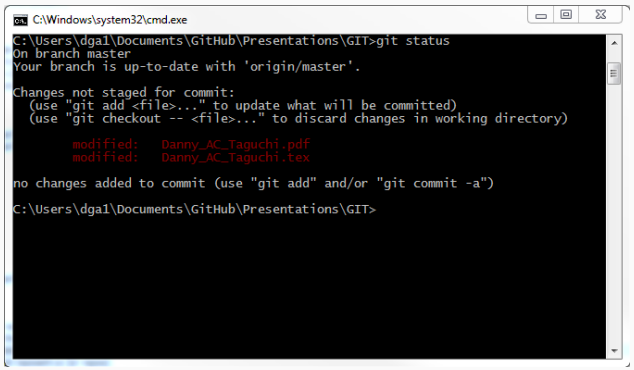
What version control system?

- There are two populer version control systems (git SVN)
- We will cover git as it is the most populer, the one I know and easyist to use
- Git was developed in 2005 by Linus Torvalds

(Principal developer of Linux)

What is git?

- Git is just system of version control



- By default it is used through a terminal
- Git has progects cauld reposetrys (or repos)
- It then has a tree struchure to manige the vershons

Tree structure?

[master] 6c6faa5 My first commit - John Doe

[develop] 3e89ec8 Develop a feature - part 1 - John Doe

[develop] e188fa9 Develop a feature - part 2 - John Doe

[master] 665003d Fast bugfix - John Fixer

[myfeature] eaf618c New cool feature - John Feature

[master] 8f1e0e7 Merge branch `develop` into `master` - John Doe

[master] 6a3dacc Merge branch `myfeature` into `master` - John Doe

[master] abcdef0 Release of version 0.1 - John Releaser

The master branch is in gray with colour branches coming of and then being merged back

Key comards?

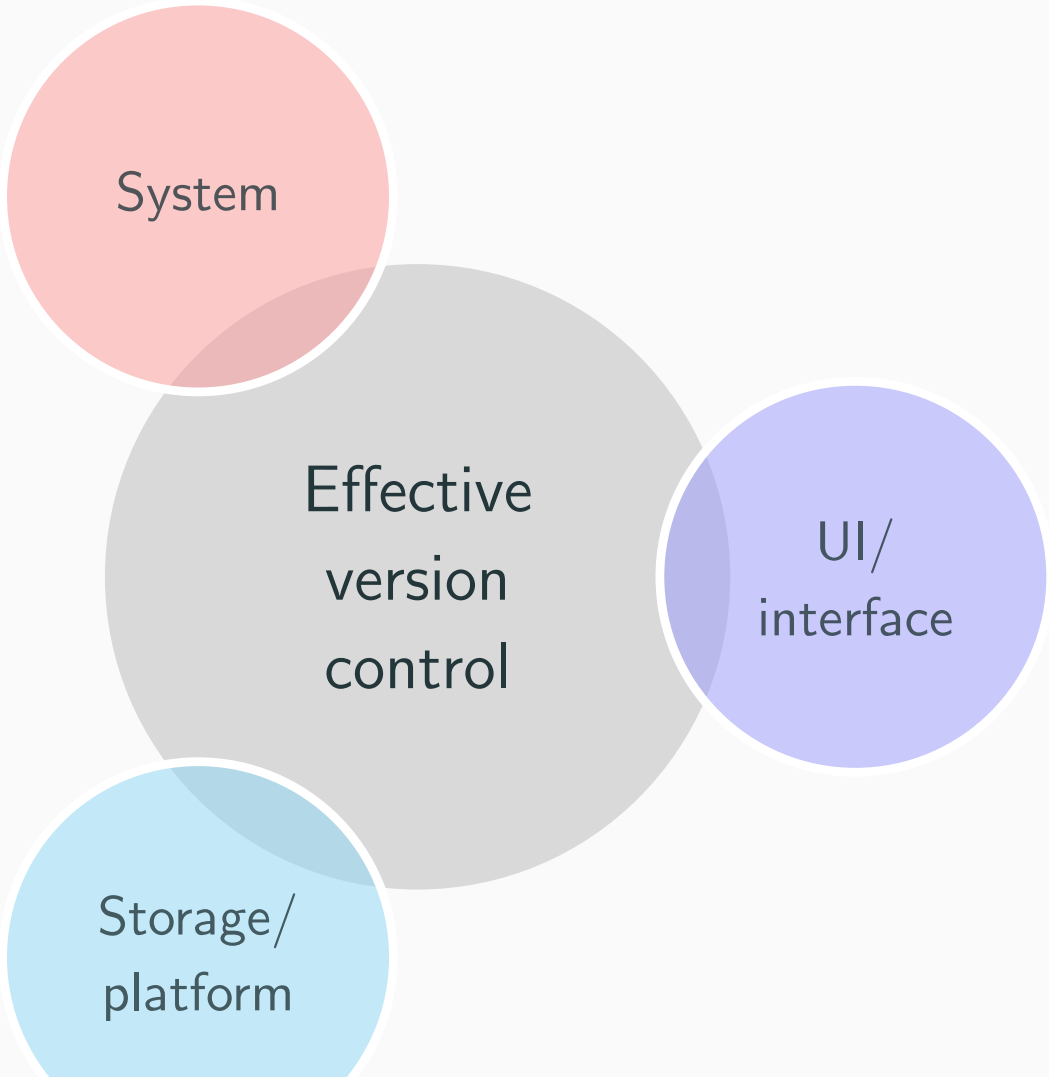
In the terminal interface there are some useful commards:

- *init* - inisholies a repo
- *status* - tells you if you have flies out of sync with the current vertion
- *commit* - adds new files and changes to the current vertion
- *push* - pushes comits to an online repo (only if using a git platform)
- *pull* - pulls commits from an online platform (only if using a git platform)

Platform

What platform?

You can just use git with the treminal on your PC or you can store work on an online reposetry mainging site, the most populer of these are GitHub and Bitbucket



Conclusions


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

(xkcd)

References i

(xkcd)