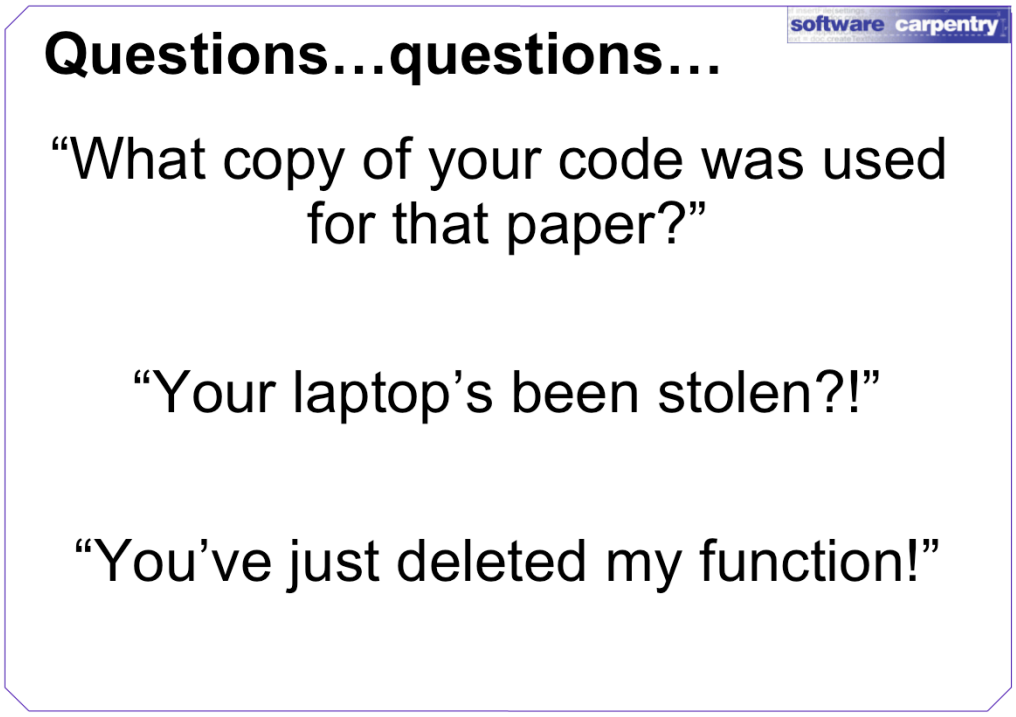
Version control

Chris Cannam
SoundSoftware
Mike Jackson
The Software Sustainability Institute



Version control

Imagine if, during your viva, you are asked “so, this graph here, can you show me the software you wrote to create the data that is presented here?” Could you do it?

You may store different versions in directories with different names, but…
If your laptop was stolen would you lose 6 months of work?

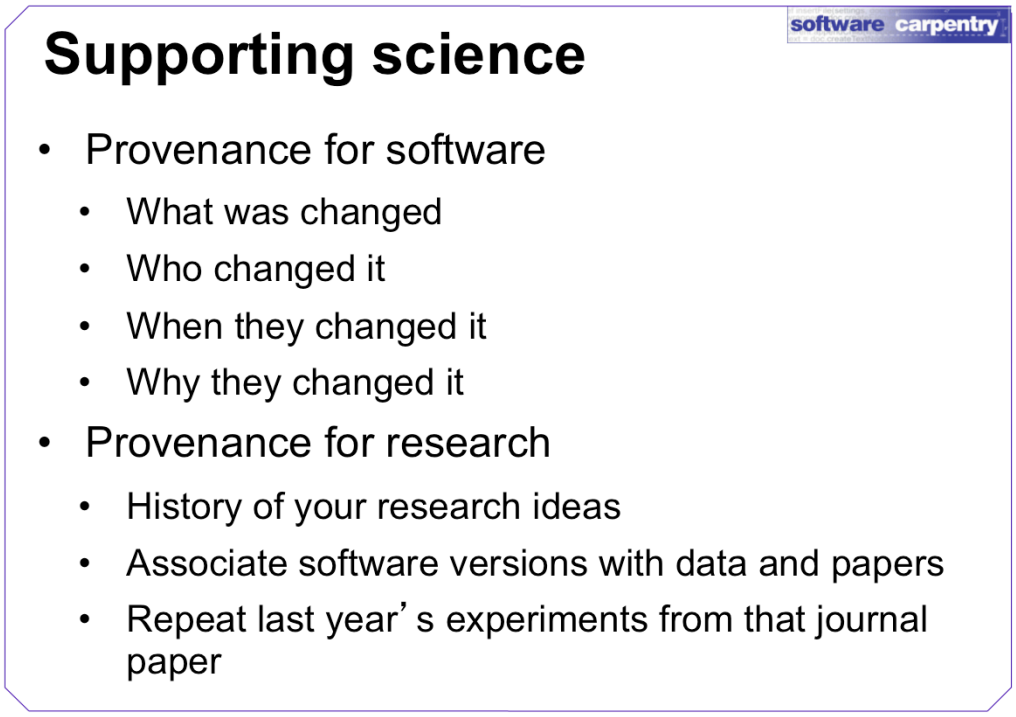
If you’re working on the same code with others in your research group could you do so without losing forever someone else’s work?

Using shared directories still risks losing work even if you’ve backed up the directories.

One tool allows you to,
Manage the evolution of your software.
Not lose your software.
Work collaboratively on software.

Version control

- Software to help keep track of changes made to files
  - Source code
  - Configuration files
  - Documents



Version control.

Source code control.

Change control.

Like a log book for software – ideas, evolution, fixes, false paths.
Contributes to reproducibility e.g. for paper or thesis if someone asks "how did you produce this exact graph?"

Supporting security

- Repository is a master copy
  - Directories
  - Source code
  - Configuration files
  - Documents
- You work with a working copy
- Retrieve changes from any point in time
  - “The one yesterday…that was working”
  - “Oh, that way I did it before was better after all”

Version control

Anything created manually – programs, original observations, paper sources in LaTeX etc.

Anything that a compiler or a tool can produce don’t need to go in as they can be recreated.
Avoid dead laptop-lost code-lost research.

“Undo” for developers.

Supporting collaboration

- Support collaboration…with others
  - Work on the same files
  - Find out when each other made changes
  - Merge changes without making a mess
- Record who, what, when and why
  - Nothing is lost!
- Support collaboration…with yourself
  - Work on the same code on your laptop, desktop, lab machine…

Made changes e.g. introduced a bug!

What you do…

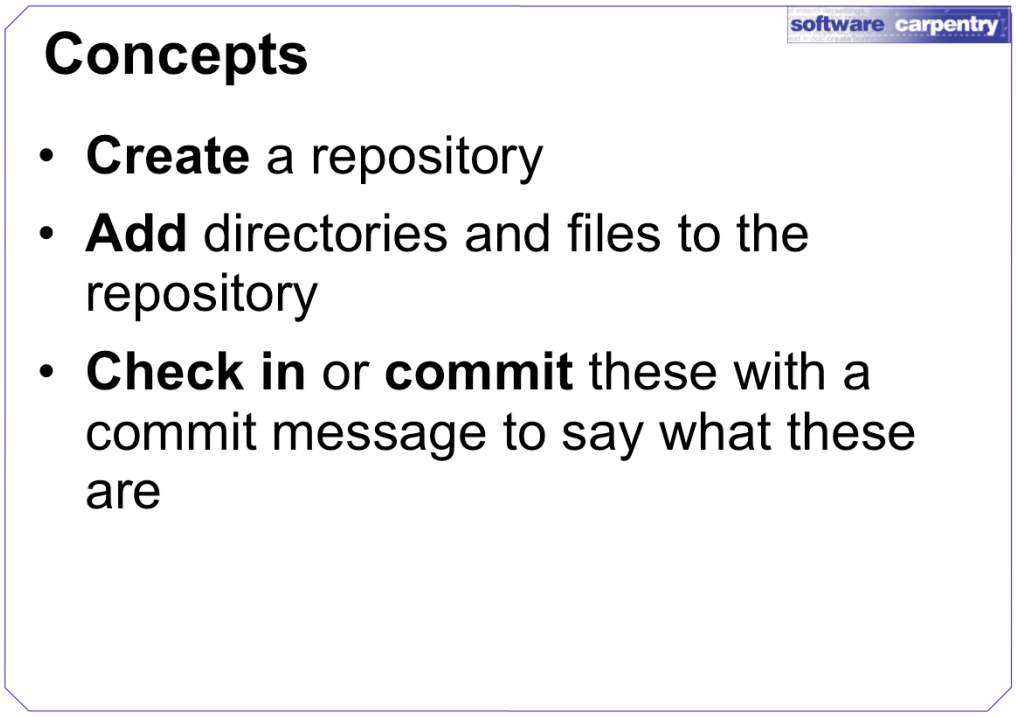
- Version control needs you to tell it…
  - What files it needs to manage
  - When you make a change
  - How to resolve conflicts
- Your colleagues need you to tell it…
  - Why you made changes!

Popular products

- Centralised
  - One shared repository
  - CVS
  - Subversion
- Distributed
  - Every developer has their own repository
  - Push/pull to synchronise repositories
  - Git
  - Mercurial

Version control

Other useful tools support limited forms of revision control. Wiki, DropBox, GoogleDocs all preserve histories of pages and documents and support roll-back to previous versions.



Products differ in commands the concepts are the same.
Other useful tools support limited forms of revision control.
Wiki, DropBox, GoogleDocs all preserve histories of pages and documents and support roll-back to previous versions.

Concepts

- **Check out** a working copy
- Cycle
  - **Update** your working copy or **pull** changes from the repository
  - **Merge** your copy with copy in repository and **resolve** any conflicts
  - Edit files in your working copy
  - **Check in** or **commit** changes with a commit message to say **why** you made changes
  - **Push** your changes to the repository

Commit for every logical change.

Concepts

- **Revert** to the repository copy to throw away your changes
- View the **difference** between your working copy and the repository
- View the **history** of changes to a file

software carpentry

"If you are not using …… …… then, whatever else you may be doing with a computer, you are not doing science"

Greg Wilson

Version control

software carpentry

"If you are not using version control then, whatever else you may be doing with a computer, you are not doing science"

Greg Wilson