Content vs. Pathnames

coursera.org/learn/git-distributed-development/supplement/DD2qs/content-vs-pathnames

Let's see what this means in practice.

git commits tend to be all the changes for logical grouping (a feature, a bug fix, etc) including changes in all necessary files. For example, with RCS (Revision Control System) you might do:

```
$ rcs co product_manual.doc

$ rcs co src_slct.c

$ rcs co src_slct.dlg

$ rcs co src_slct.hlp
```

to check out the files to revise. Then, you would edit them to get the bug fixed or feature implemented. Finally, you check them back in with:

```
$ rcs ci srv_slct.c

$ rcs ci srv_slct.dlg

$ rcs ci srv_slct.hlp

$ rcs ci product_manual.doc
```

Tags must be used in RCS or other similar version control systems to group these changes together. If someone is in the middle of the check-in process and a system build begins, the build may fail. Backing out changes require finding all the files with the same tag. If someone mis-tags one of the updates, backing out the changes is nearly impossible.

With git, by comparison, you edit the files to get the bug fixed or feature implemented and then, you do:

```
$ git add srv_slct.c
$ git add srv_slct.dlg
$ git add srv_slct.hlp
$ git add product_manual.doc
$ git commit
```

The git commit step contains the new versions of all four files. Backing out the changes if a problem was discovered only involves removing the commit.