



Less 'Oh Sh*t' With GIT

How having a GIT-based workflow can make your life infinitely easier

What with a who-now?



- Who the hell are you?
- What the hell are you talking about?
- What the hell is a 'Workflow' ?
- What the hell is 'GIT'?
- Oh. Why the hell am I not using this thing?

Who the hell are you?



- Richard Tape
- Sticky, sello, masking, video, worm. You're welcome
- Working with WP professionally for >3 years. Thanks Mike!
- @RichardTape
- <http://rich.is/> (coming soon) but have site at <http://iamfriendly.com/>

!important



- Here's what I'm **not** saying:
 - ★ That my WorkFlow is the One True Way ©
 - ★ That what I do is 'right'
 - ★ That you should copy what I do without having a basic understanding of what you're doing
 - ★ Manchester City deserved to win the Premiership Title this year

What the hell **are** you talking about, then?

- What is a Workflow and why do I need one?
- Richard, what were you doing this time last year?
- Haha. That's ridiculous. You're an idiot. What are you doing now, though?
- But I already have my way of doing things and it works great. When do you stop talking? You're boring.
- OK, OK, as it'd apparently be 'rude' if I got up and left, you might as well tell me about this 'GIT' thing. ENTERTAIN ME.

What is a Workflow?

You're going to crack out a dictionary definition aren't you?
You're **so** predictable.

1. The scheduling of independent jobs on a computer.
2. The set of relationships between all the activities in a project, from start to finish. Activities are related by different types of trigger relation. Activities may be triggered by external events or by other activities.
3. The movement of documents around an organisation for purposes including sign-off, evaluation, performing activities in a process and co-writing.

And in English, please?

How you go about your day-to-day tasks and the systems you put in place to allow you to do those tasks more easily

Richard, what were you doing this time last year?

- Work locally using a MAMP setup.
- WordPress Multisite installation, each ‘project’ was a new site.
- When happy with work, sFTP to live site. Cross fingers and sacrifice a goat in the hope that it wouldn’t break.
- For edits: either edit file locally and re-upload or, if I was in a rush, edit the file on the live server. Yes, I’m going straight to hell.
- For database: local and live sites were entirely separate.

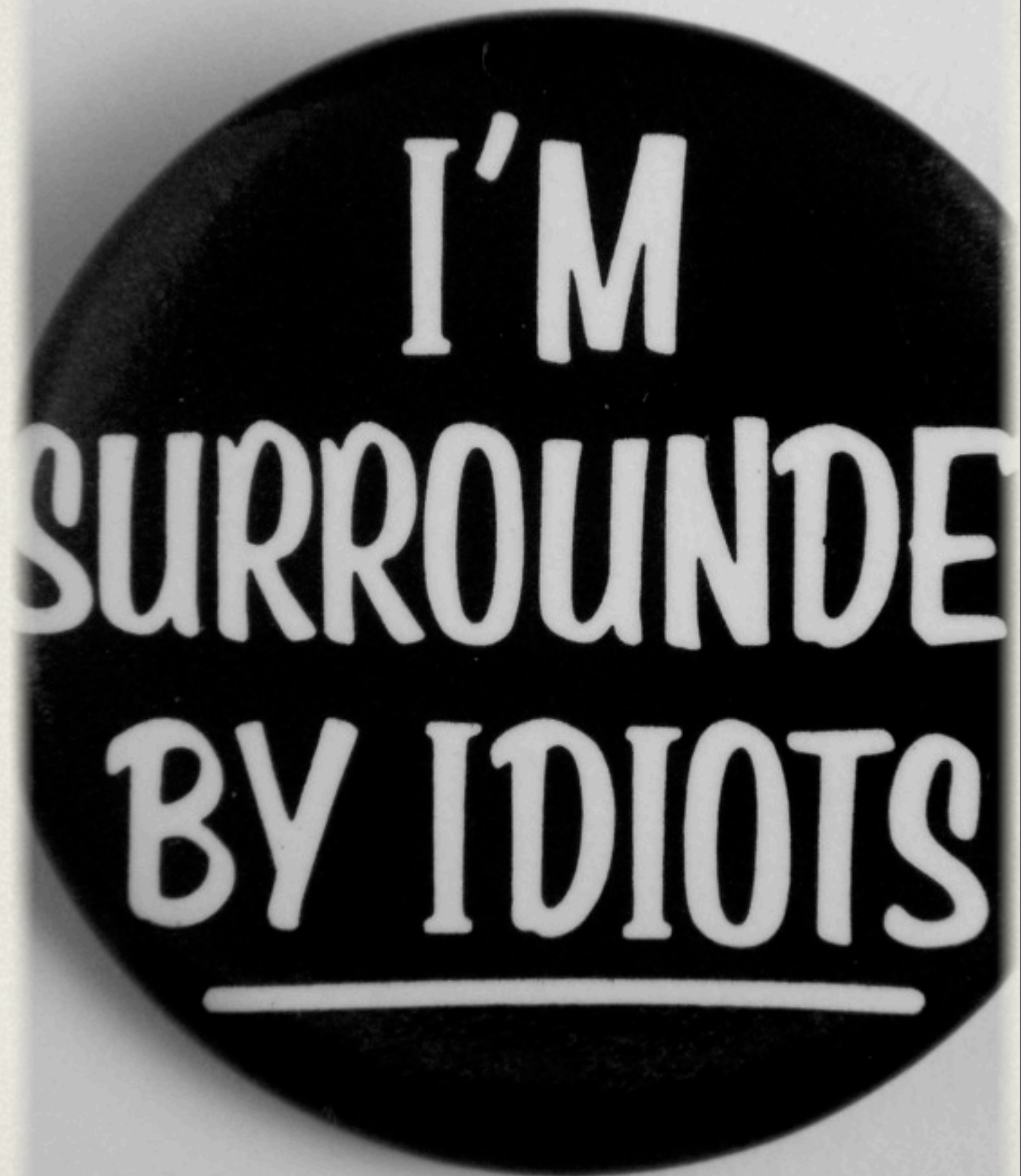


Live file editing?

Hahaha. HAHAHAHA.
AAAAHAHAHAHAHA.

You're an **idiot**.

(Secretly, I sometimes do this too. But don't tell anyone)



What are you doing now?

- **All** source files (including WP core) are version controlled in GIT - stored in local repos and on github
- I make **all** edits locally (even though I am sysadmin on my live server, I have intentionally revoked my rights to edit files directly)
- Commit to local git repo. Push to github.
- Post Commit hooks on github automatically pulls all edits onto staging server
- If happy on staging, deploy changes to live

GIT? Repo? Staging? Deploy?

- GIT is a VCS - a **v**ersion **c**ontrol **s**ystem (a 'distributed' one at that) and source code management (SCM)
 - ★ It keeps track of all changes you make to your files
 - ★ You can think of it like the Revisions System in WordPress
- Repo is short for Repository.
 - ★ It's sort of like a database
 - ★ When you make changes to your files, you 'commit' them to your local repo
 - ★ If you have made a mistake, you can simply 'roll back' to a previous state
 - ★ Sort of like an 'undo' feature for all of your files

GIT? Repo? Staging? Deploy?

- Staging

- ★ An intermediary between your local environment and the live site
- ★ Ideally, the staging server and the live server have identical setups
- ★ Means you can see exactly what your changes will look like on the live site without the panic of 'guessing'

- Deploy

- ★ The equivalent of FTP'ing all your changes elsewhere
- ★ When you 'deploy' something you're effectively "making it live"

Hang on. So you don't FTP? Anything?

- Nope. Not a sausage.
- If I need to install a plugin I do it locally. Then commit and push to github.
- This then automatically grabs all files to the staging server (which means FTP is disabled on my server so (a little) more secure.
- Likewise with themes.



Iiinteresting. Veeeeery
interesting.

OK smart-arse, what about content?

- Content is a **little** more complicated
- Text is stored in a database, so no 'flat files' as such (there is, kinda, but it's complicated and not for now)
- Can do a 'database dump' change references to 'local' environment and replace with links to live site
- Or...
- RAMP Plugin by Crowd Favorite
<http://crowdfavorite.com/wordpress/ramp/>



RAMP



Holy Frack. How Much?!

- Yeah, it's not cheap. But it does a great job.
- Also, it has a developer API, so all of the themes and plugins that I develop are 'RAMP-enabled' which mean that any user of said themes or plugins can benefit from RAMP's content management
- Free Alternative: DeployMint by Mark Maunder
<http://code.google.com/p/deploymint/>

Care to show me an example?


This can only go well



OH GOD! THE CONSOLE

- You don't **have** to use the console, there are several apps you can use
- Tower or github for mac
- GIT Gui / TortoiseGIT for Windows

```
1234567862
[agarvin@cerberus ~]$ date +%s
1234567862
[agarvin@cerberus ~]$ date +%s
1234567863
[agarvin@cerberus ~]$ date +%s
1234567863
[agarvin@cerberus ~]$ date +%s
1234567863
[agarvin@cerberus ~]$ date +%s
1234567864
[agarvin@cerberus ~]$ date +%s
1234567864
[agarvin@cerberus ~]$ date +%s
1234567864
[agarvin@cerberus ~]$ date +%s
1234567864
[agarvin@cerberus ~]$ date +%s
1234567872
[agarvin@cerberus ~]$ date +%s
1234567873
[agarvin@cerberus ~]$ date +%s
1234567874
[agarvin@cerberus ~]$ date +%s
1234567875
[agarvin@cerberus ~]$ date +%s
1234567878
[agarvin@cerberus ~]$ date +%s
1234567879
[agarvin@cerberus ~]$ date +%s
1234567879
[agarvin@cerberus ~]$ date +%s
1234567881
[agarvin@cerberus ~]$ date +%s
1234567882
[agarvin@cerberus ~]$ date +%s
1234567883
[agarvin@cerberus ~]$ date +%s
1234567884
[agarvin@cerberus ~]$ date +%s
1234567885
[agarvin@cerberus ~]$ date +%s
1234567887
[agarvin@cerberus ~]$ date +%s
1234567888
[agarvin@cerberus ~]$ date +%s
1234567890
[agarvin@cerberus ~]$ echo YAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAY
YAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAY
[agarvin@cerberus ~]$
```

But I work in a team?

- On some projects, so do I!
- Different people have their own 'local' repository and work on different branches
- These are then 'merged' as and when

What about backups?

- Well, because your **code** is all stored on GitHub, you already have 1 level of backup. RAMP handles media, too.
- For the database, I use a plugin called BackupPress (<http://www.getbackuppress.com>) - it 'just works' and allows you to restore your database back to a previous state if things go awry
- Alternative: WP-DBManager (<https://wordpress.org/extend/plugins/wp-dbmanager/>)
- I also have a cron job on my server which does a dump twice a day.

I've forgotten absolutely everything you've just said

- Having a workflow you are both comfortable with and that you stick to is more important than what the workflow actually is
- Use a Version Control System; they will - after a small learning curve - become a God-send
- Never, ever edit files on a live server. Ever. Forever, ever.
- Have a fallback plan
- Don't just jump straight into something - do some reading about it first, understand it, then take small steps

You've been generous.

THANK YOU



R

A collection of six WordPress logos arranged in a 2x3 grid. The logos are in various color combinations: orange with a blue 'W', blue with an orange 'W', white with an orange 'W', orange with a white 'W', white with a blue 'W', and blue with a white 'W'. They are set against a background of numerous small, light gray rectangular tiles, each containing a different word or phrase in a sans-serif font, creating a word puzzle or Scrabble-like effect.

-