

Don't document. Automate!

W. Adam Koszek

Koszek ORG

wojciech@koszek.com

2022-10-05T17:48:17Z

When I go to an article or a GitHub page it sometimes amazes me how good stuff people produce. I'm building an sensor platform recently, and a lot of problems I solve by using software others wrote and made it free. This is great. But there's one pattern which I need to criticize: people write way to much.

What do I mean by that?

The normal flow of getting software to work is typing a series of steps. For example it may require typing 10 commands in the command line. And these commands for ninety five percent of people will likely be the same. For the rest 10% they'll be slightly customized, but not much. Nothing that couldn't be achieved with 10-line shell script consuming 1-2 parameters.

But people don't do that. They prefer to write elaborate documentation instead. And then I need to go and re-type everything with some silly changes. I still feel grateful for seeing the correct steps, but this makes little sense, wastes my time and I don't like it.

Here's a bad example on my part: my Synology HOWTO You can see what I mean: the tools I've made are useful, and yes, you'll end up with a decent solution for usable NAS system, but to get there, there's some drama on your side for re-pasting all these steps.

What can we do instead?

Automate. Automate every flow you've figured out for yourself and others. If you've spent 15 minutes on making set of commands work, spend 15 minutes more and automate. Here is a little set of rules I made for myself:

- if it's a programming project of any kind, wrap it in `make` or whatever other build tool you use (`rake`, `cmake` etc.)
- if it's a UNIX or OSX project: write a `.sh` script for it.
- if it's Windows, write `.bat` script.
- if there's a chance script must be run remotely, wrap it in Ansible.

I try to follow these for all of my projects. To give you some examples, in the last several months I've automated most of my complex flows. Some of them:

- Gem release flow for my OSX build tool MacB: `makefile`
- Build/release process for my iOS app: `build.sh`
- My middleman based homepage: `makefile`
- Build steps for my DFA simulator: `build.bat`

The very important advantage of these flows is that I see they work, since I can go and wire them to Travis CI or AppVeyor and the green badge in my repositories will give me and others vote of confidence.

Improving and automating my flows is on my TODO. Are all your workflows automated?