



Managing Modules Without Going Crazy

brian d foy
brian.d.foy@gmail.com

At the Nordic Perl Workshop, I gave a new talk I'm calling "Managing Modules Without Going Crazy". It goes through how to be a prolific CPAN author without doing too much work.

I have many modules on CPAN, and even if most of them aren't widely used, that doesn't mean they take any less time to maintain. A bug is a bug, and uploading a distribution takes the same amount of time no matter how many people use the module.

So how can I keep up with all the maintenance without losing all of my free time? The trick is to figure out what I really want, then to automate those things.

Value Your Time

My first principle in not going crazy is to respect my own time. Paraphrasing Yoda: "Wasted time leads to stress, stress leads to insanity, insanity leads to bug reports."

Before I decide to spend time on something, I need to figure out if I really want to spend my time doing it. Sometimes it's not even a conscious decision, so I have to be aware of that too. Am I working on a module or creating a new one because I'm avoiding doing real work?

I'm not going to bother going through all of the time management techniques and advice (although you might want to check that out separately). I just want to figure out if the time I trade to do something with Perl is worth it to me. Now, I certainly like programming in Perl and doing new things, but what do I get out of it? What happens if I don't do it?

For instance, sitting on my desktop, I have a proto-module dealing with poker hands, but it will probably never make it to CPAN. I'd really like to finish it so I can use it while I play poker online (hey, that's what I like to do, and I know that), but the work I'd put into it isn't going to get me anything. Sure, it could calculate hand probabilities and other really cool

things, but I don't think it's going to help me win any more hands.

Not only will my half-baked poker module not help me win poker, but if I upload it to CPAN, I'll have less time to play poker. I'll have to maintain the module as people send in bug reports and request new features. Despite the idea that anyone can participate in open source, they mostly seem to participate with requests instead of patches. Do I really want to create more work for myself?

Tolerate Shaggy Yaks

Let's say that I decide I really do need to make a module and create some work for myself. I think my contribution to the community by finishing the module and uploading it will provide some value.

So, I sit at my computer and start to hack at it. Then I discover that there's a new release of Perl, so I stop working to install that. While I'm doing that, I find out that I need to update my Berkeley DB support, so I update that before I compile Perl. Since I'm already here, I might as well update `mod_perl` and `apache` too. Oh, and look at that, the new version of Perl is binary incompatible to the last one, so I need to recompile a lot of modules.

All of this extra work that doesn't get me any closer to my goal is known as *yak shaving*. For most work, I could have probably used the Perl I already had installed, so I should have just used that. It's not perfect or beautiful, but I'll talk about that later.

My litmus test in these situations is "Am I doing a sysadmin or programmer task right now?" If I think I'm doing sysadmin work, I'm probably yak shaving. Sure, it would be nice to have the latest versions of everything, but most times I don't really need them. I wait to upgrade until I absolutely need to.

Well, that's the goal anyway. I've been fixing my system for my upgrade to DBM: : Deep 1.00, which