Metaprogramming how much is too much?

Asked 15 years, 6 months ago Modified 14 years, 5 months ago

Viewed 872 times

3

As I have become more and more comfortable using metaprogramming techniques I have found more and more applications for metaprogramming as well. I'm working now on a little project in which I am creating classes and instances of these classes on the fly and I'm wondering if I have taken metaprogramming too far? Is there such a thing as too much metaprogramming? If so, where does one draw the line?

ruby metaprogramming

Share

Improve this question

Follow

asked Jun 22, 2009 at 18:39

Bryan Ward

6,683 ● 8 ● 38 ● 48

5 When it doesn't help anymore and it complicates more than it solves? :) – Skurmedel Jun 22, 2009 at 18:40

1 When you're having fun rather than getting things done. – Andrew Grimm Jul 27, 2010 at 5:26

4 Answers

Sorted by: Highest score (default)

10

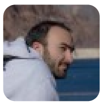
When it becomes too difficult to

- test
- debug
- understand
- maintain

then it's probably too much. Metaprogramming is supposed to simplify your daily development. If you end up spending more and more time maintaning complicated code base, you probably need to simplify it.

Share Improve this answer Follow

answered Jun 22, 2009 at 18:43



Simone Carletti
176k ● 50 ● 367 ● 368

1 I would say if it gets ANY MORE difficult to test, debug, understand and maintain--to anyone on your team, not just yourself (Note that initial coding is NOT in this list! Typing less is never a good excuse to add complexity to the end processes). This should be the rule that all programming features are selected by--what does the job, and then of the things that do the job, what is simplest for the next guy to maintain. Period. – Bill K Feb 26, 2010 at 18:42

metaprogramming is like violence, if it is not working, you're not using enough.

8

In all seriousness, metaprogramming can get excessive very quickly, but as long as the code is readable, it does not impact anything global in spooky ways, and the effects are well contained it is fine. That said, often when you get in the metaprogramming mindset, you forget how to do things the normal programming way. Metaprogramming and OOP have a lot of overlap in functionality, so you should see if there is a simpler way of accomplishing your goal. Yehuda Katz has a funny story about just such an abuse: http://yehudakatz.com/2009/03/06/alias_method_chain-in-models/

Share Improve this answer Follow

answered Jun 22, 2009 at 18:45

Ben Hughes
14.2k ● 1 ● 42 ● 34

5

You can know you are doing too much meta programming when you can never end your original project (the thing you were supposed to be working on), but keep on developing lots of useful reusable components.

Enrique, a partner in a start-up long years ago, was a meta programming fanatic. I remember he started programming a report for the system he was working on, it was a task for a day or two. One week later the report was not there yet but a fantastic general library for building any type of report you want was forming in his mind.

By the way, I learnt more about programming with Enrique that in years of work before, his view was inspiring... just that he used to went too far and we really needed to finish some work...

But is better to run the risk of being too much meta. Maybe you have read the typical "some programmers are more productive by a factor of 10x"... I believe the key factor to become one of those 10x superstars is thinking meta.

Share Improve this answer Follow

edited Jun 22, 2009 at 18:49

answered Jun 22, 2009 at 18:42

tekBlues
5,783 ● 1 ● 30 ● 32

0

if you and your team are feeling confortable with your techniques, and you achieve the goals of the project in time... then you have not taken metaprogramming too far.

Share Improve this answer Follow

answered Jun 22, 2009 at 18:43

Jonathan
12k ● 5 ● 58 ● 91