

# If you measure something, people will change their behavior to address the measurement and not the thing the measurement is intended to measure

 [devblogs.microsoft.com/oldnewthing/20101116-00](http://devblogs.microsoft.com/oldnewthing/20101116-00)

November 16, 2010



Raymond Chen

We all know that once you start measuring something, people will change the way they behave. We hope that the change is for the better, but that's not always the case, and that's especially true if you are using the metrics as a proxy for something else: People will manipulate the metric without necessarily affecting the thing that your metric is trying to measure. I was reminded of this topic when I read a story in The Daily WTF of a manager who equated number of checkins with productivity. One metric that nearly all software products use to gauge productivity and product progress is the number of bugs outstanding and the number of bugs fixed. Of course, not all bugs are created equal: Some are trivial to fix; others are substantial. But if you believe that the difficulty distribution of bugs, while not uniform, is at least unbiased, then the number of bugs is roughly proportional to the amount of work. The bug count is just a rough guide, of course. Everybody works together, with programmers promising not to manipulate the metrics, and managers promising not to misinterpret them. At least that's how it's supposed to work. (All that text up to this point is useless. When you're telling a story, you have to include a lot of useless text in order to motivate or set the scene for the actual story that comes up next or just to make the story sound like an actual story instead of just a sequence of events. What amazes me is that so many people seem to focus on the "literary throat-clearing" and miss the actual story!) A friend of mine told me about a project from many years (and jobs) ago. Things were pretty hectic, people were working late, it was a stressful time. The bug statistics were gathered by an automated process that ran at 4am, and every day, management would use those statistics as one factor in assessing the state of the project. My friend was wrapping up another late night at the office after polishing off a few bugs, and as a final gesture, re-ran the bug query to enjoy the satisfaction of seeing the number of bugs go down. Except it went up. What happened is that another member of the project was also working late, and that other member had a slightly different routine for wrapping up at the end of the day: Run the query and look at the number next to your name. If it is higher than you would like, then take some of your bugs and transfer them to the other members of the team. Choose a victim, add a comment like "I think this is a problem with the XYZ module" (where XYZ is the module the

victim is responsible for), and reassign the bug to your victim. It helps if you choose victims who already have a lot of bugs, so they might not even notice that you slipped them another one. By following this simple nightly routine, you get management off your case for having too many outstanding bugs. In fact, they might even praise you for your diligence, since you never seem to be behind on your work. Of course, management looks at these manipulated numbers and gets a false impression of the state of the project. But if you're not one of those "team player" types, then that won't matter to you.

And if that describes you, then I don't want you working on my project.

Raymond Chen

**Follow**

