

5 Bugs Minutes

Mike Kelly teaches how to build credibility with developers and win a free lunch in the process

I have a game I play with programmers. I call it five bugs in five minutes. It's been 1 while since I've done it (you have to be sitting close to the people doing the programming), but I still get to play it every now and then.

The game works like this:

1. A programmer finishes coding some testable unit of work.
2. The programmer comes over and throws down the testing gauntlet.
3. You go over to the programmer's desk and they launch the software you will be testing, navigating to the code that is supposedly "done," and the programmer establishes the limits to the software they are presenting.
4. You accept or you decline the challenge (noting that declining lowers your project "street cred" and will eventually lead to programmer ridicule and sitting alone at lunch).
5. If you accept, you test for five minutes with the programmer looking over your shoulder.
6. Every time you find a bug, you laugh sadistically and tell the programmer the bug you found.
7. The programmer (and this is important) confirms or denies that what you found counts as a bug. Don't get caught arguing your bug right now, get the feedback and move on. You can discuss the relative merits of your bug after the time limit has expired.
8. After five minutes you stop.
9. If you found five bugs (that the programmer

confirms are bugs), the programmer pays you five dollars (or buys you lunch).

10. If you fail to find five bugs (that the programmer confirms are bugs), you pay the programmer five dollars (or buy them lunch).

I find that this game has the following side effects:

- Over time, I tend to get more free lunches than I give.
- Over time, I develop better communication with the programmers; they become more willing to help me, and they become more interested in fixing my bugs.
- Over time, the programmers get better at preventing the types of bugs I find - forcing me to change my test techniques and raise the bar of my testing.
- Over longer periods of time with the same developers, I start to give more free lunches than I get.

I also find that over time I tend to learn a lot about what the programmer thinks the software is suppose to do, how they designed it, and how it's suppose to integrate with the rest of the system. As I'm learning about the software, they are learning about what kind of problems I'm concerned about and how I find them. I find that I'll rarely find the same bugs more than a handful of times. Programmers are quick learners – and they like free lunches. This game allows us to both grow in a fun little competitive way.