# The Human Impact of Bugs

Last night, Matt Drance took to the Twittersphere with a heart-wrenching experience all of us can relate to:

*iPhoto just demolished 3 months worth of videos from my phone. That trust is not earned back, ever.*

*— Matt Drance (@drance) March 15, 2013*

*Videos of my children, including Son 2's first Christmas. I'm completely shattered.*

*— Matt Drance (@drance) March 15, 2013*

If you've used a computer in the past 30 years, you probably had the same reaction I did — extreme sympathy and a sinking feeling in your gut as you recalled the times you've lost precious data to a computer. Maybe it was a college paper, that email you were almost finished writing, or pictures of your kids. We've all been there before.

Specifics aside, Matt could have been using software written by *any* of us. As software engineers and designers, it's easy to lose sight of the very human impact of the bugs we ship. Whether it's a careless user experience with sharp edges, a broken algorithm that we know works "well enough", or the bug we consciously decided we could afford to fix later, it's important to remember that there are real people on the other end of the terminal. A real person is going to use your software and **expect** it to work. In some cases, they even paid you good money with this expectation. They are trusting you with their data, their time, and in some cases their personal memories.

Shipping is hard. Realistically, we are never going to fix all of the bugs in our software. Anyone who tells you they don't ship until all the bugs are fixed is either lying or naive. The real world is far too diverse, complex, and random to ever produce bug-free software. When faced with the need to ship, compromises inevitably get made.

The key along the way is making the right choices and *caring about what you create*. In our industry's highest form, we all exist as professional craftsmen and craftswomen. When you are dedicated to a craft you take pride in ownership, pride in your tools and methods, and pride in the solid, well-architected underpinings you laid down that no one may ever know exists.

Time is always short, deadlines loom, but we should all take the time to focus each day on writing code or designing UI that is better than what we made the day before. Learn

from your mistakes, fill the gaps in your knowledge, and remember the very normal person who uses your software and **needs** it to work reliably.

*Found them on the filesystem. I'll deal with the rest later.*

*— Matt Drance (@drance) March 15, 2013*

Matt's story has a happy ending, but not everyone is so lucky or has 8,000+ tech-savvy Twitter followers ready to offer technical assistance. The next time you are working a late night, staring down a huge list of bugs to fix, or putting off writing that unit test, remember the real human impact of the bugs you ship.