Hey, let's report errors only when nothing is at stake!

devblogs.microsoft.com/oldnewthing/20110729-00

July 29, 2011

Raymond Chen

Only an idiot would have parameter validation, and only an idiot would not have it. In an attempt to resolve this paradox, commenter Gabe suggested, "When running for your QA department, it should crash immediately; when running for your customer, it should silently keep going." A similar opinion was expressed by commenter Koro and some others. This replaces one paradox with another. Under the new regime, your program reports errors only when nothing is at stake. "Report problems when running in test mode, and ignore problems when running on live data." Isn't this backwards? Shouldn't we be *more* sensitive to problems with live data than problems with test data? Who cares if test data gets corrupted? That's why it's test data. But live data—we should get really concerned when there's a problem with live data. Allowing execution to continue means that you're attempting to reason about a total breakdown of normal functioning.

Now, if your program is mission-critical, you probably have some recovery code that attempts to reset your data structures to a "last known good" state or which attempts to salvage what information it can, like how those space probes have a *safe mode*. And that's great. But silently ignoring the condition means that your program is going to skip happily along, unaware that what it's doing is probably taking a bad situation and subtly making it slightly worse. Eventually, things will get so bad that something catastrophic happens, and when you go to debug the catastrophic failure, you'll have no idea how it got that way.

Raymond Chen

Follow