

# Value without Numbers

by James Bach

A READER WRITES: “I CURRENTLY MANAGE a quality assurance group. As with many testing organizations, I am often challenged to justify our existence. One question that senior management often brings up is, ‘What is your group saving me?’ As with everything, I’ve been asked to express it in quantitative terms.”

Justifying our role and measuring the value of our performance are problems many of us struggle with. Though some roles on projects *create* value, testing does not. Instead, we *defend* value. We help value cross the street safely; we carry its groceries to the car. The ultimate reason testers exist is to provide information that others on the project need to create things of value—but I’m skeptical that there will ever be a meaningful way to quantify what money we save the company.

Only a fool would think solely in financial terms, refusing to consider other scales and concepts of value, and most people aren’t fools. Still, it’s important to know what to do when the bean counters come for your beans. So let me walk through some points I find helpful when responding to them.

## **The value of testing, just like the value of most other roles, is hard to quantify.**

Look around you. *The value of most roles is hard to quantify.* Consider saying, “I don’t know how much money testing saves us, but there are a lot of roles—management, to name one—that we might agree are valuable even though we can’t assign them a specific dollar value.”

## **One way to understand the value of testing is to imagine life without it.**

Testers find problems. What is it worth *not* to have those problems in the product? Testers provide confidence. What is it worth to management to sleep at night without worrying that their prod-

ucts and systems are critically flawed? Good testers find *important* problems *sooner*. What is it worth for the project to ship sooner, instead of dragging on and on while nasty bugs slowly trickle to the surface?

Enlist management to help you assess your value. Show them the list of problems that would have made it into production if you and your team had not been on the job. Ask them what *they* think it was worth to avoid those problems. Consider saying, “I don’t know how to quantify what testing saves you. But the real question for you is whether you want our team on the job and looking for problems that could sink the company and hurt shareholder value. If you think such problems could exist, then you need testers. We are here to help you understand risk. It’s up to you to decide how much risk you can handle and what it’s worth to monitor and reduce it.”

## **The value of testing is lumpy.**

I was once stuck in a cab with an avid gold panner. He told me that most of the time he doesn’t find enough gold to make it worth his effort. But it’s not the *predictable* payoff that makes gold panning worth doing. It’s the *surprise windfall*. You never know when you might find a big gold nugget.

It’s the same with testing. You pan for bugs, and sometimes you find a big one. Sometimes what you learn with a test changes the course of an entire project—like the time my test team was being briefed on a new feature about to be delivered, and one of us asked an innocent question that revealed a fundamental design flaw. Other times, test results have no impact on anything, because the product is “Good Enough” and no one pays attention. Thus, testing progress is lumpy. It doesn’t accrue at a steady and predictable rate; it comes on glacially, then it leaps upward, then it pauses, then it leaps again, like a crazy staircase to the end of the project.

Consider the 1994 Pentium division bug, which was caused by bad entries in a table that in turn were caused by a simple error in a script that copied that table to a programmable logic array. It was an inexpensive bug to create, yet it ultimately cost Intel about a half billion dollars. Now *that’s* a lump. If Intel’s testers had found the script error early enough, they would have saved Intel all that money, probably without even realizing it.

## **If you know what one problem costs, use it as a model.**

Imagine going to Intel’s management, before the famous Pentium bug came to light, and asking for \$25 million to improve testing. They would probably ask, “What does that save us?” No matter what spreadsheets you conjured up, it would be a very tough sell. Now imagine coming to them after they announced the \$475 million charge to pay for the Pentium bug fallout—\$25 million now looks like a bargain if it helps avoid future debacles.

When I was at Borland Software, one \$250,000 problem caused by a single bug became the justification for two years of improved project discipline and QA.

## **Find out what’s behind the question.**

Often, questions about justifying and quantifying things are a cover for deeper issues that aren’t so easy to talk about. Maybe management is preparing for a layoff, or maybe they’re unhappy that a particular bug escaped from the development process and made it to the field. It may only be that they don’t know enough about testing to discuss its details. Consider saying, “You ask an important question and I want to give you a useful answer. In order to do so, it would help me to know how you will use this information. Can you tell me about that?”

I hope you see a common thread run-

ning through these responses: smile and be friendly, but don't give in to the temptation to cobble up financial figures just because your management wishes that life were simple. Testing is about dispelling illusions, not inventing them. [STQE](#)

---

*James Bach is the founder of Satisfice, Inc., a test training and consulting company. A pioneer in the emerging disciplines of Good Enough quality and exploratory testing, James specializes in expert testing under chaotic conditions. He can be reached at james@satisfice.com.*

**STQE magazine is produced by STQE Publishing, a division of Software Quality Engineering.**