

No More Whining

Reframing the “Not-Enough” problem

by Johanna Rothman

“**T**est is always at the end of the schedule. We never test for as long as we need to. We get whipped around by whatever the developers do. It just isn’t fair.”

“We don’t have enough testers.”

“My management makes the ship decision. I don’t get to say no.”

“We didn’t test enough, and now I don’t feel like I’ve done a professional job.”



Johanna Rothman urges us to embrace reality and deal with the “Not-Enough” problem.

When I meet people in my work, they are genuinely concerned about their perceived inability to do a good job for their companies. They know they could do a great job if they had enough time, or enough testers, or if they had real input into the ship decision. These people feel browbeaten, as if they are victimized by their circumstances.

If you feel like this, you’re not alone. But you can choose to *not* feel like a victim; there is another way. Consider re-

framing your perspective on your organization’s quality investment.

Reframe #1:

Shipping your product is a business decision, not a quality decision.

Many organizations schedule testing toward the end of the project and refuse to change the ship date—even when it’s obvious that you won’t be

able to do the work you wanted to do. When they do this, they’re not asking you to find all defects, even though they may say so—they’re really asking you to bless the software. So go ahead: Do whatever testing you can in the time you have, and then report on what you’ve tested and weren’t able to test. Concentrate your report on the risks of shipping the product. Remember, *Shipping the product is a business decision*. You will be most effective if you can describe the risks of shipping to the rest of the organization, rather than by exercising your authority to stop shipment.

You don’t have to stop product shipment to be useful to your organization. The fewer times you say, “We can’t ship this now,” the better off you will be. Instead, explain the risk of shipping in terms of money and time. You can say things like, “I expect all/plenty/some of our customers to find this defect. Without a workaround or a fix, we can expect a ton of/lots of calls. Our developers will end up spending time we can’t afford responding to those calls.”

Develop release criteria with your developers and with Management to assess the risk of shipping the product. Do it early in the project, as part of a project team, to help the rest of product development—and with an understanding of three important things:

- The risks you see while testing the product
- Where more people would help
- When the product is ready to ship

As a tester or test manager, you don’t have enough information to see the whole business picture—so it’s inappropriate for you to stop product shipment. On the other hand, it’s
(Continued on page 79)

TONY SCARPELTA

(Continued from page 80)

quite appropriate for you to explain that you can't assess the state of the product because you haven't tested the product yet. If your management wants to live with the consequences of shipping, that's *their* prerogative.

Reframe #2:

Testing is a mechanism to assess risk of product shipment, not a mechanism to find "all" of the defects.

When I first started working as a tester, I was in a QA group. My job, according to my manager, was to test and to find "all of the defects." After the first week, when I realized the developers were writing self-modifying code—and they weren't going to stop—I returned to my manager and said, "Nope, my job *can't* be to find all of the defects. It's hopeless. I never will. So I'm going to look for the high-risk defects, okay?" He wasn't happy about it, but when I explained the situation to him he decided it wasn't the end of the world.

He and I talked with the rest of our QA group, and decided that we'd all first assess risk in the product. Then, we'd focus our testing in the risky areas, rather than try to find all of the defects. We changed our approach to testing. Before, we'd assumed we had to develop automated tests, starting at the beginning of the component and going through to the end. Now we looked at the component's architecture and design to decide where the areas of risk were. We asked the developers what parts they'd had trouble with, and what kinds of pre-QA testing they'd done. We spent a little more time investigating and planning, and that paid off in deciding where to test.

Here's a short self-test. Check all that apply, and no cheating:

- I think it's my job to find all of the defects.
- I think you can really only test at the end of the development project.

If you checked either of these statements, then it's time to rethink your idea of what testing is.

You *can't* find all of the defects. You didn't create them; the developers created them.

And you can't wait to test at the *end* of the project. There's never enough time then, no matter what business you're in. You have to test starting at the beginning. Indeed, Tim Lister says the best place to start testing a project is with the schedule. Is there a schedule, and does it make sense? If there's no schedule, don't bother testing anything—you already know the product will be awful. Stop working on *that* project now, and start working on the next one.

Reframe #3:

They're paying you to do the job they want done.

Some of us don't feel as if our work is really at a professional level because we know there's more we could do. We can see many more opportunities to test the product, and to improve its quality. It's when we don't get to do those things that we may feel as if we haven't done our best. But remember that Management pays you for what they find valuable about your work. If you want to do more, explain

to them the value of doing more, and what it will take to get more done.

One of my colleagues, Steve, was concerned that he wasn't performing at his highest level of capability. He had successfully completed his first testing pass, and now he had many ideas for how he wanted to further test the product. If he'd had more time to develop and run more tests, he knew he could have found more defects. Management decided that they were satisfied with Steve's initial level of testing and weren't ready to fund more. Steve reluctantly agreed, but decided to speak with his manager after the product's release. Steve explained the trade-off of more test development time and finding test results. Steve really was being a professional by recognizing the potential to do more, discussing the results with his manager, and accepting his manager's judgement.

Stop Being a Victim

You have more capability to influence attitudes, behaviors, and actions in your organization than you know. If you feel like a second-class citizen, reframe the situation. Rethink your job and how you do it, and realize the importance of the contribution—finite, but powerful—you can make toward your organization's product quality. **STQE**

Johanna Rothman observes and consults on managing high-technology product development, looking for the leverage points that will make her clients more successful. You can reach her at jr@jrothman.com or by visiting www.jrothman.com.

SOFTWARE
Testing & Quality
E N G I N E E R I N G

Your Ad Should Be Here!

If you are seeking software testers, managers, or QA professionals, *Software Testing & Quality Engineering* magazine is the right vehicle for getting your message out. With a circulation of over 20,000 per issue, *Software Testing & Quality Engineering* is the only publication reaching this market exclusively.

For more information on placing your ad in *Software Testing & Quality Engineering*, contact Tom Jones at 630-876-5484 or tjones@sqe.com.