How do I create Value with my Testing !

by Jonathan Kohl

I had written an article for EuroStar about creating value with testing. In the article, I talk about getting feedback from stakeholders, but that isn't always easy or possible. One of the most important stakeholders on any project is *you*, so how do you go about satisfying yourself with your testing value?

The easiest way to get feedback is from other stakeholders. What does your manager think about your testing? How about the programmers, business analysts and customers (users) of your software?

The hard part with that answer is you may not be able to talk to all of those stakeholders. Or, they may not know what good testing looks like so they won't have answers that satisfy you. In some cases, the stakeholders around you may have such low expectations that their feedback might not help you at all. They may expect you to provide testing work that you might consider shoddy and negligent. In that case, you have to show them what great testing looks like. When that happens it's like graduating from a cheap box of wine to the good stuff. Once they've tasted the good stuff, it's hard for them to go back to expecting poor testing.

Even if you have good direction from other stakeholders, I recommend asking yourself some questions to help determine if you are creating value or not. This is hard to do, and will result in work for you over the long-term, much like personal growth endeavours.

Don't expect quick fixes, but if you work in these areas over time, you will see changes in your testing. Here are some things to think about:

- Is my testing work defensible? (Cem Kaner talks a lot about this.) Think of a court case. What would a jury think if you testified and described what you did as a tester and why. How did you determine priority? Why did you test some things and not test others? (100% complete testing is impossible, so you have to make decisions to optimize your work. Are those decisions well thought out, or more subconscious? What sorts of things might you be missing that you haven't thought of?)

- James Bach talks about how important it is to have thought out and varied *approaches* to testing. What kind of approach do I have to testing? Do I consciously choose to have a varied approach using as many models of coverage as I can to discover important information about the product? Do I make the best use of tools, testing techniques and management approaches that I can? Or do I just do what the programmers or someone else tells me to do?

- In the absence of getting real feedback from real people on my project, what would happen if a well-known consultant came to visit me? Could I answer their questions about why I chose to test this way? What kinds of holes might they spot in my thinking? Would they see weak spots? More importantly, would I be proud to have Cem Kaner or someone else I look up to see what I actually do? Have I used ideas from testing thought leaders in my work and found out what works well for me and what might not work so well? Could I communicate my work to an expert outsider clearly and thoughtfully? If so, what might they think?

- Do I adapt my test plans and strategies from project to project based on the risks and rewards our project environment has at a particular point in time, or do I just copy and paste what I did last time, and repeat the same thing over and over?

- Do I track down and find repeatable cases for important intermittent bugs, or do I just file them and forget about them?

- Do I feel energetic, creative and proud of my work as a tester, or do I just feel like I am doing the same boring things over and over and filling in paper work and forms to please a manager?

- Can I look at a released product and identify ways in which my testing has improved the product experience for our end users?

- Do others on my team feel better with me around? Do they miss me and my creative input when I am away, or do they welcome the break from my negativity? Do they request that I work with them on other projects?

- Is my testing service in demand? Am I the person team members come to when they need help solving a particular problem that I am really good at helping solve?

- Am I aware of other approaches to testing that challenge my favorites? Do I understand approaches that I may not favor or I may even dislike, or do I just dismiss anything unfamiliar and threatening out of hand? Do I have an open-mind and look to challenge my ideas in testing to help improve?

- Am I learning about different ways I could improve my work? Am I aware of recent changes in testing techniques and tools? Do I know where to find information to learn from?

- Do I consistently try to do better than I did last time?

These are the kinds of questions I ask myself regularly.

I don't always have the best answers to my own questions, but as time goes on, I feel much more confident about both my own answers to those questions, and more importantly, the value I know my testing work provides.

Biography

Jonathan Kohl is the founder and principal software consultant of Kohl Concepts, Inc.

Noted software testing thinker and strategist, Jonathan is a natural investigator on software projects.

In addition to assisting teams with testing, Jonathan helps companies define and implement their product vision; coaches practitioners as they develop software on teams, and works with leaders helping them define and implement their strategic vision.

Jonathan supports and contributes to Open Source testing tools. He is the founder and project lead on the Session Tester project, and was an original commiter and contributor to the Watir project.

Jonathan is also a popular author and speaker. His blog on software development and testing issues is one of the most well-read testing blogs in the industry.

Jonathan can be contacted at http://www.kohl.ca/.