

Of Deck Chairs and Ocean Liners

Stephen R Balzac

“Just give me the numbers!”

Falling firmly into the “I just can’t make this stuff up” category, the preceding statement was made by the head of a certain engineering department. He wanted the performance figures on a series of database lookups so that he could determine if the database code was performing up to specifications. This would be a perfectly reasonable request except for one minor problem: the database code was not producing the correct results in the first place. Performance was sort of irrelevant given that getting the wrong answers quickly is not necessarily all that helpful, although it may be less irritating than having to wait for the wrong answers. It’s rather like driving at 75mph when lost: you may not know where you are or where you are going, but at least you’ll get there quickly. Or something.

In another example, the engineers developing a bioinformatics data analysis package spent all their time arguing about the correct way to set up the GUI elements on each page. The problem was that when they actually ran one of the calculations, the program appeared to hang. In fact, I was assured by everyone, it just “took a long time to run.” How long? The answer was, “maybe a few weeks.”

This may come as a shock to those few people who have never used a PC, but a few weeks is generally longer than either a PC or a Mac will run before crashing. Besides, the complete lack of response from the program regularly convinced users that the program had crashed. The engineers did not want to put in some visual indicator of progress because they



Stephen R Balzac

felt it wouldn’t look good visually. They refused to remove that calculation from the product because “someone might want to try it.” Eventually, they grudgingly agreed to warn the user that it “might take a very long time to run.”

In both of these cases, the team was solving the wrong problem. Although there were definitely complaints about the speed of the database, that was very much a secondary issue so long as the database wasn’t producing correct results. And while the user interface decisions were certainly important, designing an elegant interface for a feature that will convince the user that the product is not working is not particularly useful. At least rearranging the deck chairs on the Titanic was only a waste of time. It didn’t contribute to the ship sinking.

The element that made each of these situations noteworthy is that in both cases there were people

present pointing out that the wrong problems were getting all the attention. The people making the decisions didn't want to hear that. They wanted to solve a certain set of problems and, by golly, they were going to solve them! This is a version of the Hammer syndrome: when all you have is a hammer, everything looks like a nail. Sometimes, though, that nail turns out to be a thumbnail.

So why were these teams so insistent upon solving the wrong problems?

Fundamentally, because they could. Simply put, if you give someone a problem they can solve comfortably, and one that they have no idea how to approach, they will do the former. In addition, they had never established clear metrics for success, never established standards by which they would know if the database was fast enough or the user interface was good enough. As a result, they built their goals and evaluated their performance around those issues. They were not being evaluated on whether they got the right answer, despite the opinions that the customers might have in that regard.

While clear, specific goals are certainly good things, goals also have to make sense. When a company is constantly seeing flaws in its products, it can be a very valuable exercise to look at the goals assigned to each person and each team in the company. Do those goals make sense? What problems or challenges are they addressing? Are the goals complementary, or are there

significant gaps? If the engineering team is being evaluated on how many bugs they can fix and the QA team on how many new bugs they can find, what happens to the step where fixed bugs get verified? If no one is responsible for that happening, it won't get done (and didn't, in several software companies!). If

the team focuses on the wrong problems, they'll spend their time fighting symptoms or revisiting solved problems, and never deal with the real issues.

Therefore, even before you can set goals, you have to know what the problem is that you are trying to solve. That means first separating the symptoms of the problem from the problem itself. The symptoms are only symptoms; frequently, they can point to many possible problems.

It's important to look at the symptoms and brainstorm which problems they could be indicating. When you start developing solutions, you then need to ask what the final product will look like if you go ahead with your solution and you need to know what success looks like. Make sure that your proposed solution will actually solve at least some of the potential problems you've identified, and develop some way of testing to make sure you are solving the correct problem. In other words, have some checkpoints along the way so you can make sure that you're actually improving things. Only then can you start to set goals that will effectively guide you to producing a quality product.

What are you doing to make sure that you are not rearranging deck chairs on the Titanic?

And while the user interface decisions were certainly important, designing an elegant interface for a feature that will convince the user that the product is not working is not particularly useful. At least rearranging the deck chairs on the Titanic was only a waste of time. It didn't contribute to the ship sinking.

Stephen Balzac is a consultant and professional speaker. He is president of 7 Steps Ahead (www.7stepsahead.com), an organizational development firm focused on helping businesses get unstuck. Steve is a contributing author to volume one of "Ethics and Game Design: Teaching Values Through Play." Steve's book, "The 36-Hour Course on Organizational Development," published by McGraw-Hill, is available at Amazon.com, Barnes and Noble, and other book stores. Contact him at 978-298-5189 or steve@7stepsahead.com.