

Customers and how they Contribute to Estimating Angst

Posted At : July 10, 2017 5:04 PM | Posted By : Mark Kruger

Related Categories: Business Of Development

This is the moment the developers have been waiting for. After having thrown both the developer and his application under the bus, we are going to make room under there for the customer. Now if you are customer reading this, take heart. While I'm going to say some things that will hurt, in the end I'm going to rub salve on the wound and we will all sing Kumbaya together. Here we go.

The Muse finally has a number - a project that we've estimated at 350 hours. Of course, this estimate is not correct. Why? Because no estimate is *ever* correct. Please refer back to the first post in this series. The majority of estimates are wild guesses. We've done our best to find out all we can about the system, requirements, priorities and every little nitpicky detail we can imagine. Our estimate is based on that discovery. If *all of our dozens of assumptions are correct* then our estimate is rock solid. But of course, there are still things we don't know and many of our assumptions are simply off. Much of what we need to know we simply *can't know* until we dig in and start doing the work. For that reason we will give Bob a range of hours - say 275 to 375 hours - and dozens of caveats that he will epically fail to hear and absorb. What he will hear is the number **275** and *that* will be the basis of his expectations.

Let's check in with the Muse and Bob as they finalize the project.

- **Muse:** Well Bob, as you can see by the painstakingly detailed 13 page document in front of you, we estimate 275 to 375 hours. Have you read the document?
- **Bob:** I read... I got to page 2... uh... I did see the estimate at the end. So, 275 hours?
- **Muse:** [*caveat 1*] 275 hours to 375 hours. You can expect it to be somewhere in the middle.
- **Bob:** 275 eh? Wow that's a lot. But I get everything I asked for?
- **Muse:** [*caveat 2*] To be clear you get everything we have carefully outlined and detailed in the proposal. If it's not in the proposal it would be "out of scope" - not included in this estimate.
- **Bob:** Well, I can probably swing 275 as long as I can have it in 6 weeks.
- **Muse:** [*caveat 3*] Well the timeline depends on resources and other factors. This is probably a one-woman job. It will take her around 4-5 weeks to do the core programming, then QA, revisions etc. You are probably looking at 8 to 10 weeks. [*Caveat 1 again*] It's 275 to 375 hours not including out of scope items that tend to arise.
- **Bob:** Hmmm.... ok, I think we can do it. But just one thing. Did we discuss adding reporting to this application?
- **Muse:** ...
- **Bob:** Muse... you there? Did we lose our connection?
- **Muse:** I'm here Bob... I was just asking my assistant to quickly remove sharp objects from my immediate vicinity. Now about your reporting...

Most clients are not quite as bad as Bob. Still, every contractor or development shop owner has a story remarkably similar to the one above. I know most Muse readers are developers. I can see you now in my mind's eye with righteous indignation and clenched fist saying, "You tell them muse, clients are the worst!" But actually, *this is*

not their fault. Your customer has his or her own domain of knowledge. I have a customer who is a commodity trader. I expect him to know the difference between a bull call and a butterfly spread, but I do not expect him to know the difference between MySQL and MS SQL, how the cloud works, or why it takes 10 hours to program something. The basic issue is threefold:

- Communication
- Expectations
- Cart and Horse problems

Today let's tackle the first two.

Communication: The Zen of Seeing

The Muse is working on advanced degrees (I'm a lifelong student) at our state university here in NE. I am privileged to interact with a wide swath of people from different countries. Sometimes, when they find out I am a businessman these students ply me with questions about how to make money in America. I enjoy this much more than when they find out I'm a programmer and they ask me to fix their laptop or printer. To make money, I usually suggest they apply for a job at the mint, but that sometimes gets lost in translation. Recently I had an interaction with a fellow from the DRC (Democratic Republic of Congo). We'll call him Samuel. Sam sent me an email and asked to meet me with a business proposition. I agreed. The following is a loose transcript (and yes this is a real conversation).;

- **Muse:** So, you say you have a business proposal.
- **Sam:** Yes - in my country, in the Congo, there are many opportunities to make money.
- **Muse:** Excellent. Let's hear what you have.
- **Sam:** How about import-export.
- **Muse:** ... ok, what are we exporting or importing?
- **Sam:** Oh many things! Congo is full of artists and we grow many crops.
- **Muse:** Ah... well, crop importing sounds a little out of my area of expertise.
- **Sam:** What about fishing? We could buy a boat and...
- **Muse:** Let me stop you there Sam. Have you ever heard of a business plan?

I spent the next hour going over a sample business plan and showing him the information - including having a very specific idea where he can demonstrate expertise - that he would need to have a shot at attracting any American business to work with him.

Sam's problem was a basic misunderstanding or interpretation of how this all works. He had been in America, used American products and taken note of American wealth and prosperity. He assumed parts of that world would translate into his own experience in broad strokes, never realizing the level of knowledge and detail it actually takes. Moreover, he had a fuzzy idea that he could, "Start a business with American capital if he found an American partner." This is actually true, but it is plainly superficial.

Customers often arrive with the same surface knowledge of what they are attempting to do. But remember, a customer's understanding of computing comes from using his laptop and phone. He is not automatically aware of the vast network of servers and application end points that makes simple things possible. How many average non-IT folks know that your phone taps dozens of other computers out there "on the Internet" throughout the day for various things? To them it is connected to Verizon and Apple -

and the image is not a cloud of vCPU in a ginormous warehouse datacenter. It's a tower chassis. If they have imagination it might be 10 feet tall, but it is still a "thing" not a system.

The problem is compounded by the simple fact that most customers don't really know what they want until they *see something*. This is not a unique problem. Have you ever purchased a paint color only to bail once you "see it on the wall"? When you purchase a car do you have a complex, detailed and exacting list of features and components to match precisely or do you go and test drive a few cars and "discover" you can't live without that automatic hair clipper and eyebrow weaver? Customers too need to see things. Wire frames, story boards, screen shots, artwork - all of these so more than simply show what *you* can do. They advance your customer's understanding of his or her own desire. Yet all of these cost money to produce. That is a real dilemma for shops like mine. How do we produce proposals that sell knowing we will eat the cost if it *doesn't* sell? But we will save that for another time.

The point here is that customers need visual clues to understand the whole process. Even after you begin a project one of the best things for your customer is to create things they can see. It makes customers nervous when you spend the first 200 hours of a 300-hour project on everything *but* the front-end UI. Order your work to keep the customer engaged. In the end, he or she will feel better about it.

Expectations: Those Little Cobbler Elves

The work of programming is a black box to many customers. Like the helpful elves in the tale of the cobbler (it's a shoe thing - nothing to do with apples) they expect to approve the project, go to sleep, and wake up 8 weeks later with a houseful of shoes. There is no point of reference for many of them to the actual work involved. Developers know it's like a cross between exploration, art, and a math examination. But ask a customer what is it like and - is it like working with a spreadsheet? Nope. Is it like writing an essay? Nope. Is it like building a house or repairing a car? Nope and nope. It does not match their point of reference. To illustrate, indulge me by considering this Muse parable from 2009.

An amazing ancient document was unearthed in Peoria last month. It contained the account in ancient writing of a man who lived before the time of the wheel and running water. This man - let's call him Grog - had a vision of something in the future and he described it in great detail. It starts with a vast hall...

Behold I saw a great hall. It was many cubits long and many cubits wide (the ancients were all about cubits), but the ceiling of the hall was low. The ceiling and the walls of the hall glowed with strange unearthly light. Indeed, the hall was filled with wondrous magic all around, and with such sites and sounds as I have never seen. There were many people in the hall. Some were laughing and some were angry but all of them were talking noisily. The great hall was filled with thunder like the coming of a storm and the people were performing a ritual that was strange, wonderful and terrible to behold.

Near each group was a beast with a giant maw as large as a man's head. The mouth was dark and foreboding. There were 21 such beasts. One by one each person approached the mouth of the nearest beast and prayed to it with one hand hovering, palm down, near its mouth. As they prayed the beast would vomit up a strange and wonderful object. From the depths of the beast came a stone orb - but

not like any stone I have ever seen. It was smooth as if it had been washed in a brook by the water of many years. It seemed wondrously round like the moon or the sun. It glowed and shimmered in the light. The stones came in many colors - blue, red, yellow, green and often black. Some of the stones seemed inlaid with quartz or jewels or pearl, yet somehow still smooth. When the beast delivered up a stone orb a worshiper would pick it up and hold it beneath his eyes in prayer - being careful not to look at it. The worshipers the eyes would focused on the other end of the hall.

At the other end of the hall there were groups of tiny white trees placed as a sacrifice. A very straight and smooth path to the trees had been crafted from wood, but it was wonderful wood indeed! It was smooth like winter ice and it shined with a gleam like new fallen dew. The path was not for the worshipers. It was made to receive the orb. The worshipers would rush up to the edge of the path and fling the orb down the path toward the trees. The orb would roll on the path like a ripe fruit rolling down a hill, and it would crash into the trees. This crashing was the thundering sound that filled the hall. Sometimes the worshipers would fling the orb but miss the path. When they missed the path many of them would fall to their knees immediately at the edge of the path or even grovel prostrate on the ground. However, if a worshiper's sacrifice was successful and the orb crashed into the trees and threw all of them down, the worshiper and his companions would clap their hands together and raise a cry to heaven in rejoicing.

Now if you are the savvy muse readers I think you are you know that Grog is describing a bowling alley. Why is it "strange and wondrous" and why does Grog use odd metaphors that don't seem to fit? Because he lacks point of reference. He has nothing concrete tying his vision to his experience - he has no bridge to help his understanding.

For this reason - the point of reference problem - customers have very little idea what something *should* cost. So what do they use? Often, they base their expectations on what it *seems* like it should cost. If they have what they think is a *simple idea* they expect a low cost. If they have something they feel is exceedingly innovative, they may expect a high cost. But as any developer can tell you these feelings do not predict reality.

Expectations can also be an outgrowth of budget and wishful thinking. Customers may arrive with a monetary number in their head. For them the process may seem like a negotiation or bargaining - "How do I get these guys to agree to do this work in *n* number of hours". For these customers, the process is very frustrating. Neither hours nor timeline will yield to your budget or expectations. Only the rate is left negotiable. This leads to frustration and often a bad project launch.

Conclusion

Finally, when it comes to estimating keep in mind that you are the bridge between IT and the regular business world. Work to communicate, be patient, respect and honor the knowledge that your customer brings to the table in his or her own realm. Don't fall into the trap of being snarky about customer's lack of understanding. Remember how clueless you were at the auto repair counter or the last time you tried to pick out "a nice cabernet" from a lengthy wine list. Everyone brings different things to the table. Make sure your "things" help rather than hinder the process.

In my final post on this topic I will discuss ranged quotes and agile.