Estimating Part 3: Explaining the Costs to Customers

Posted At : September 10, 2009 1:56 PM | Posted By : Mark Kruger
Related Categories: Business Of Development

As we have discussed in our earlier posts on the Business of Web Development inexperienced customers (ones who have never done an IT project) are often surprised at the cost associated with a project. This is partially the result of the reputation that the web has for being cheap. Customers look at services like godaddy.com for example, and they see that they can register and host a site for the cost of skipping a couple of frappuccinos a month. While this is true, it is really not the same as professional design and development services and high performing, scalable, redundant, mission critical hosting services.

In fact, if I could digress to hosting for a moment, customers often fail to see the cost benefit of a more complete "managed" hosting setup. They spend thousands on development and then try to save a few hundred dollars a year on hosting. Having settled on hosting "on the cheap" they often have to pay someone a high hourly rate to do things like troubleshoot an underperforming server or handle DNS settings or figure out their mail services for them, or (worst of all) alter their code to conform to a changing server environment - like when a host recently disabled createobject() on a server causing an application to fail for someone who is now *our* customer. Any savings they might have gained is eaten up in support costs and they are actually *losing* money on the deal. In the words of Jesus they "strain out a gnat and swallow a camel" (email me if you don't know exactly what that means - I'll enlighten you).

Of course when it comes to development costs there are other things that mystify customers. As we have discussed before, customers often only account for the visual "up-front" items of a web application. They see forms, lists, charts and displays when the reality is that the bulk of the work on *many* complicated projects goes into coding, revisions, Q/A and Project Management. Here are a few fallacies that range from the hair-brained to flights of fancy:

Data Flows Hither and Yon

A customer came in with an idea for a financial application. He was working with us to outline the data requirements for end users. We went through some complicated scenarios regarding collecting and verifying the data and then the customer said casually that "...naturally this data then flows into the estimating application." I have heard this "data flows" phrase before and I'd just like to say I hope the fleas of a thousand camels infest the armpits of whoever thought up this catchy little phrase (no doubt it's the same guy that thought up "parlay"). Data does not "flow" from one application to another as if all that was required is to walk into the server room and tip the server. A database is not a keg of beer. Data has to be accessed with painstaking development and each and every nitpicky little thing that has to be done must to be outlined, coded for and tested by someone. But thinking in terms of "data flows" makes folks think there is some magic behind it that doesn't require human intervention.

Point-of-Reference Syndrome

A customer comes in with a new idea and begins his explanation with a statement like the following, "...ok, I have something I would like you to do that shouldn't take too long"... or possibly, "I have a really easy one here"... or even ... "I'm hoping you can do this very quickly". Unless the customer is a developer or has a good deal of IT experience, each of these statements represents his *hope* and not *reality*. He or she may even be making an effort to influence the estimate and hold down the hours. The truth is that it requires a ***point of reference*** to determine if something is truly easy or not.

Consider this completely fictional story I wrote for a recent devotional blog (I know, you thought I only blogged for you.... tsk). See if you can figure out what's going on.

>
> *An amazing ancient document was unearthed in Peoria last month. It contained, in ancient writing, the account 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 before or since. There were many people in the hall. Some were laughing and some were angry but all of them were agitated. The great hall was filled with thunder like the coming of a storm. Nearly all of the people were involved in 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. Sometimes the stones seemed inlayed 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 - being careful not to look at it. Instead the eyes were focused on the other end of the hall.
>
> At the other end of the hall there was group of tiny white trees placed as a sacrifice to the orb. 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 what is Grog describing? And more importantly, what is the point? If you said a bowling alley you get a gold star. The short description above is how one of the ancients *might* describe bowling. The description is fanciful and uses points of reference (worship, beasts, stones, trees, paths) that seem laughable to us, right? Why is Grog having difficulty? Well you might think it's because he has never seen a bowling alley. But the problem with Grog is not *just* that he's never seen a bowling alley. The problem is that he has never seen anything remotely *like* a bowling alley. He has no *point of reference* on which to hang his hat (or his animal skin). His only choice is to describe what he has seen based on what he *knows and has experienced*.

Before you dismiss Grog as irrelevant to our discussion, consider that many of your customers lack a similar point of reference. Oh they know what a computer or browser is. They may even know a bit about the web. But they may be ill-prepared to understand a relational database or a web service. That genuine look of mystery on their face is because they are casting about for some reference point with which to anchor their understanding. As a developer or project manager you can often measure success by how well you manage to give them that reference point. This is why your customer relationship requires patient communication and most importantly *respect*. If a customer expects something *should* be inexpensive because his point of reference informs him in that way, it is not an invitation for you to castigate him roundly for his luddite sensibilities (wow... that's a sentence worthy of Hugo).

I am constantly reminding myself that Big Idea people are successful because of all the stuff they know that *I don't know*. Sure, I can dazzle a customer with my erudite explanations of scalable architecture and extensible code paradigms (and I know a lot about baking pies too), but the customer has his own field of expertise that could dazzle me - and often does! Can I just say this with a certain amount of trepidation. We as developers often waste time lamenting and resisting the ignorance of folks who are usually more successful than us. We need to strive to bridge gaps, not maintain them in the interest of ego and superiority.

Ok, that turned into a bit of a rant... let's move on shall we.

Third Party Blind Spots

These days many applications involve *integration* with a third party application. "Amazon Merchant Services" is a good example. They provide a swath of web services for working with their system, querying products, purchasing, shipping etc. A customer will get the impression from Amazon's marketing material that, because Amazon is taking care of so many things for them there is little to do on the development side. It has a magic feel to it. You just drop in a few scripts and off you go. The truth is that the cost of any project boils down to it's *specific requirements*. Integration with

Amazon is a terrific way to sell and they do indeed provide a broad array of services - but consider this:

- Amazon Data may need to be synchronized locally
- Customer data may need to reside on your site as well as Amazon's site.
- When something goes wrong both you and Amazon support will need to coordinate to fix it (and that could be twice as expensive).
- Changes to Amazon services (which are surprisingly frequent - one of the downsides of a very recent offering) will require additional development work.

... and I could go on. There are a host of nuances to third party integration. It *may or may not* save you money in the long run depending on your requirements. Your customer needs to understand exactly what sort of things the third party is replacing, the things it can't replace and the *additional programming that may be required*.

Halitosis Syndrome: Running Out of Scope

When working on a project with defined requirements it is somewhat inevitable that the scope will change mid-stream. Very often (perhaps more than half of the time) the customer's creative juices will start to flow as the application takes shape. They will see your screens and start asking themselves "what if". What if I added text messaging? What if we put a chart right there? What if we included geo-location? Usually the customer will understand that additional features equal additional time and money, but it is important that you prepare them for the eventuality that scope will change and that extra money will be required - no easy task.

Handling Halitosis

There is not easy solution to scope change - especially with big idea folks who have a happy meal wallet and Red Lobster eyes. At CF Webtools we try to do the following.

- Sometimes we offer "Fixed bid" projects based on defined requirements. Internally we are adding some level of margin into the bid so we can accommodate a *reasonable* amount of scope change. But the document and the fixed bid gives us a break point where we can (if necessary) indicate to the customer that a specific feature is "out of scope" and therefore will cost more.
- Including clear references to "In Scope" and "Out of Scope" in the project binder to clearly indicate when an item is "out of scope".
- Discussing scope issues ad nauseum at the beginning of a project so the customer is not surprised when we deem something is out of scope.
- Using an iterative process that leaves scope fuzzy. This is open ended as to cost (or uses a range of phases where the first phase is well defined but outward phases more fuzzy) and often makes the customer uncomfortable, but for some customers it is the only truly workable solution.

In addition to the above, *wisdom* and *communication* is the key. Write detailed requirements and try to anticipate *where* scope change is likely and *how much time and money* such changes might take. Changing scope is one of the things you need to *anticipate* and *plan for*. Set the ground rules for it with your customer and get him or her comfortable with the concept. Get comfortable with it yourself! I get really tired of developers who disparage customers for changing the scope of a project. Take the time to develop a personal relationship with your customer. Prepare them for the inevitability of out of scope items. The better you are at things like reading people, empathy, knowledge acquisition, and vision casting the more likely it is that you will be able to manage scope change without surprising or antagonizing the customer. Be a collaborator, not a mercenary. Try to account for scope change in the estimates and advice you give. The goal here is not to make the customer think like a developer. The goal is for the customer to get what he needs with no surprises.

Conclusion

I think you can see - especially if you have managed to make it through all three posts on the subject - that estimating big idea projects is not for the faint of heart. It requires experience, wisdom, good communication skills, a sense of humor and the ability to wrap your head around complex systems and see all of the possibilities. If you are not a developer but rather a big idea person yourself, I hope you have gleaned some valuable insight into why this process is challenging. In my September series I am going to try and talk a bit more about management issues.