



► Managing Projects that Customers Like

by Sharon J. Cichelli

Customers can be kind of difficult, can't they?

As a Development Lead at a major computer manufacturer, I lead projects to develop software that will be used by other employees to help us make computers. My direct customers are co-workers, which does not make them any less demanding. (In fact...) My role includes meeting with my customers to learn and document their requirements for the software, designing the solution to meet those requirements, creating a project plan with estimated durations and a delivery date, and guiding the developers to deliver the product on that date.

Some of my cohorts grumble about our customers, but I have learned the key to building a successful relationship that delivers on-time software to a happy customer: Empowerment. This article describes three ways I empower my customer to have a successful project.

The first key to empowerment is, if you'll pardon the blasphemy, the Holy Triangle. The Holy Triangle is an immutable law of nature, and if you explain it to your customers, they will respect your sagacity and know that you are telling them a great truth. The Holy Triangle is thus: A project is constrained by scope (how many features), resources (how many people, or how much money), and dates (when it is delivered). For any project, two of these can be fixed, but the third must slide. If scope is fixed, you can have all the features you want, as long as you're willing to pay for the needed programmers or let it take as long as it takes. If the price tag can not slide, or you need it on a certain date or not at all, then let's decide which of these features you can live without.

Prioritizing features is the second key to empowerment. Customers want a lot of things, and the Holy Triangle usually means you can't deliver all of them.

Software Consulting Services, LLC
630 Selvaggio Drive, Suite 420
Nazareth, PA 18064
Sales: 1-800-568-8006
Fax: 610-746-7900
E-mail: sales@newspapersystems.com
www.newspapersystems.com

Rather than gnash your teeth over how

to produce the impossible, put the decision in the hands of your customers. After you've explained the Holy Triangle, they will understand the reason to prioritize the list of features.

This works when feature requests crop up during the project, too. When they want to add a new feature, ask them which other features should drop off. There are tradeoffs, but the people paying for and receiving the product are deciding which tradeoffs are important. Because, honestly, as a developer, you don't care which features you work on, as long as your to-do list can realistically be accomplished in the time you have to do it. Let the customers decide.

Prioritization is difficult, though. Do you let the one feature that is really important but really difficult wipe out the five other features that are pretty important and dead easy? To the rescue: The Prioritization Matrix. In describing it, the matrix sounds like a lot of arithmetic, but once you set it up in a spreadsheet, you'll be prioritizing with ease.

With your customer, before the project begins, create the list of features, such as integrating with the existing systems or generating a custom report. Then, together, decide on the attributes by which you want to rank those features. Some examples are: can't publish the newspaper without it; affects a lot of users; easy to code; important for security. State those attributes in the positive, so that a high score in that area means it's a higher priority item. List your features down the left; list your attributes across the top.

Give each attribute a weight. "Can't publish without it" is more important than "easy to code," so give the first one a weight of 10 and the second a weight of 1. Now, give each feature a score in each attribute. Keep the scores simple, like "high," "medium," and "low," and make those correspond with 10, 5, and 1. You want scores and weights that are far enough apart to make an appreciable spread in the results. You and your development team should score the



attributes you're knowledgeable about—how easy it is to code, how important it is for scalability or security—and your customers should score the columns that are important to them.

Here comes the math. For each feature, multiply the score for an attribute times the weight of the attribute. So a feature that is very important for security gets a high score (10), times the weight for security (also 10, since security is important), for a rank of 100. Add up the ranks for each attribute to get the priority of that feature. Sort your list by priority, with the highest number first, and you have your prioritized list, with no bloodshed.

Here's an example, showing the formulae, for one of my website projects:

Criteria	Attracts Visitors	visitors weight	Enhances Accessibility	accessibility weight	Easy to Implement	ease weight	Priority
Features		10		5		1	
Use stylesheets for layout	1	=B3*C\$2	10	=D3*E\$2	5	=F3*G\$2	=C3+E3+G3
Provide searchable content	5	=B4*C\$2	5	=D4*E\$2	1	=F4*G\$2	=C4+E4+G4
Post product reviews monthly	10	=B5*C\$2	1	=D5*E\$2	10	=F5*G\$2	=C5+E5+G5

The results of that exercise:

- Post product reviews monthly (115)
- Provide searchable content (76)
- Use stylesheets for layout (65)

After you've prioritized your list, if the order of some of the items surprises you, then perhaps they were scored wrong, or the weights of the attributes need to be adjusted. Resist the temptation to simply change the order, because that flouts your process, and brings you back to the ambiguity you were trying to escape.

I know what you're thinking. The features that Management really wants are different from the

features that users know are important. This brings us to the third key to empowerment: Get the right stakeholders. This applies to decisions about both the Holy Triangle and the prioritization matrix. You need representatives from the groups who are spending the money, using the product, and managing its long-term lifespan. And you need to convince your customers that all of those voices are important. If you build exactly what Management asks for but the users hate it, the project is still a failure.

When I led a project that had users from all the global segments of our company, my customer liaison, the person who represented the users and defined the features, traveled to each of those global segments to collect their requirements. He then tempered those with a global view, weighing priorities in light of the whole organization. (He was a great customer, I've gotta say.) I knew we had a strong working relationship when I heard him answer one of his customers with, "Yes, the software can do that. Which of these features should drop off instead?"

My customer understood the Holy Triangle, he knew the importance of prioritizing the list of features, and he met with all the stakeholders while keeping sight of what's best for the company. I did not have to stress over resolving impossibly conflicting demands; I let the people who should make the decisions make the decisions. When I came across problems, like resource constraints or features being more difficult and time-consuming than we had thought, I brought the information to my customer promptly, and let him reprioritize as needed. He was an empowered customer who helped me successfully deliver the right product.