Marathon or Sprint?

Recently, I had the opportunity to be stuck in an airplane in Albuquerque, heading towards Atlanta. We boarded the plane a few minutes late, but managed to pull out of the gate and head to the runway within 10 minutes of the scheduled time.

Once we got to the runway, we sat. And sat. And sat… Eventually, the plane engines sped up, and everyone breathed a sign of relief as the plane departed. Unfortunately, all it did was move forward a few hundred feet, take a taxiway, and head back towards the terminal. Passengers were immediately asking "What's happening?" The flight attendants never made an announcement and neither did the flight crew. People started calling on their cell phones, initializing the frantic *can-I-get-rebooked-on-another-flight-or-airline?* calls.

We approached the terminal and stopped short. At that time, the pilot finally came on the intercom and said, "The temperature/humidity conditions are marginal, so we've decided to de-ice the plane before takeoff. We'll be off in 15 minutes, and we're predicting that we'll arrive in Atlanta only 5 to 10 minutes late. Everybody should have no problem making their connections." There were massive sighs of relief as we all smiled and realized that there were no catastrophic schedule changes in store for us. Everybody relaxed, except for me – I thought of how this would be a perfect topic for a BACKTALK.

In software development, we also have passengers. We call them customers. And, just like the passengers on the airplane, they feel they have a right to know what is happening.

The problem arises when it comes down to *How Much Do I Tell My Customer*? At what level do you start sharing (infrequent) good and (frequent) bad news?

I have consulted on software projects that have ranged from full and open communication to *just keep telling them we are on schedule*. And, as expected, these projects have also ranged, in a different dimension, from "Wow – delivered on time and on budget!" to "Well, time to update the old resume." And in all cases, open communication made the difference between (on yet another dimension) "Wow, what a great team of developers!" to "Can Tony Soprano arrange a hit for me?"

Here's a secret: *All* software development projects have a lot of politics involved. And, to misquote a famous author[1], "Politics are like sausages – it is best not to watch them being made." Internal politics make for changes, changes result in bad news, and that requires communication between developers and customers.

On the developer side, people come and go. Good developers often leave, retire, get sick, take a better job, burn out, freak out, sneak out…you get the picture. They hire new personnel, retrain others from other projects (projects that have had their budget cut, were cancelled, etc). Critical folks take leave at the worst time. And every time developer staff changes, productivity metrics changes. Earned value moves around

On the customer side, requirements change. Frequently. Not because we didn't know them, but because of politics. The nine different stakeholders can't agree on interfaces. Small issues become huge (and yet, huge issues seldom become trivial – pity).

And through it all, both sides have to develop a cutoff bar of how much information is shared with the other side. It's not about us vs. them, it's about politics. It's about contract issues. It's about budgeting. But, underneath it all, it's about trying to get the project finished on time, on budget, and with happy end users. Delivering bad news is never a good thing – but it's often a necessary thing.

Being a good consultant, I have a workable answer to the question *how much information do I share?* Put yourselves in their shoes, and ask yourself would their job be easier if they knew more about the politics that I am involved in? Can I tell them additional information about upcoming events without doing damage to myself (or my organization)?

You don't want to be one of the pick-up-the-phone-and-tell-them-everything kind. You need a filter mechanism to filter out garbage and a metric to figure out how much to tell them. It probably requires a centralized communication channel – don't undercut the boss by telling the other side things he or she might not want discussed.

The best advice I have ever seen given to a development team was "Remember, it's a marathon, not a sprint[2]." In case you don't know the origin of the word marathon, it comes from the legend of Pheidippides, a Greek soldier, who was sent from the town of Marathon to Athens to announce that the Persians had been defeated in the Battle of Marathon. It is said that he ran the entire distance without stopping and burst into the senate with the words "Masters! Victory is ours!" before collapsing and dying due to exhaustion.

I don't think the messenger (or the project) should die. Marathon was about a victory, not a defeat! When we win the battle, as we cross the finish line, we might be exhausted in victory, but it will be because of the words we have said up to that point, not what we gasp as we finish. *Please* don't kill the messenger.

— David A. Cook
dcook@aegistg.com
The AEgis Technologies Group, Inc.

Notes

1. Nobody really knows who said "Laws are like sausages – it is best not to see them being made." See <http://en.wikiquote.org/wiki/Otto_von_Bismarck>. It has been attributed to Otto von Bismarck, Winston Churchill, Benjamin Disraeli, Clarence Darrow, Mark Twain, and Kaiser Wilhelm, among others.
2. Oddly enough, the development team was using the Agile methodology Scrum, which has periods of development called "Sprints." Come on, this is funny! And thanks to Mark Mitchell for repeating this advice frequently <http://en.wikipedia.org/wiki/Marathon>.

Can You BACKTALK?

Here is your chance to make your point, even if it is a bit tongue-in-cheek, without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CROSSTALK, we also accept articles for the BACKTALK column. BACKTALK articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BACKTALK article should be entertaining and clever or original in concept, design, or delivery. The length should not exceed 750 words.

For a complete author's packet detailing how to submit your BACKTALK article, visit our Web site at <www.stsc.hill.af.mil>.