

I am not sure how many of you work in technology, but those that do have most likely experienced at least one major outage. For those that have not experienced a major outage, I can give a brief example of what it means at Bear Stearns. At Bear, when any situation is known to the central IT operations staff to be impacting the productivity of a business unit, a major outage is declared. The entire IT management team in support of the given technology is paged with status. Application and engineering escalations are initiated, and everyone that can help gets on a conference call to ensure their efforts are coordinated with the rest of the team. The sense of urgency with which individuals address major outages is caused by the knowledge of its impact. If it is a trading application, the team knows the business can't trade. If it is an email server, the team knows the business can't communicate with its clients. The key to successfully managing technology and technologists to serve the business is to have that clear definition of requirements and that urgency to coordinate efforts every day.

As a technology manager, it is extremely hard to create a sense of urgency around a task when accountability for the whole is widely distributed. Though it can happen in response to outages, on a daily basis, technology deliverables are broken down into tasks by project managers. This makes technologists focus on pieces of projects rather than the success of the projects themselves. If you have not read the book *Leading Geeks* by Paul Glen, I highly recommend it for its depiction of this phenomenon. Technologists who are working on small pieces of a whole for which they are not responsible tend to focus on honing their skill set so their piecework can be marketable in case the project manager in charge of the larger effort screws up and leaves them without a job. This is expected behavior.

From where I sit in Information Security, most of the reason technology and technologists get distracted from serving the business is that they do not know how the business works and have little interest in how technologies interact except in the case of their own deliverables. I always say that developers see infrastructure as a vast jungle through which they have to hack away with a machete in order to deploy their applications, while administrators see the infrastructure as a neat garden with well-defined network paths and pruned services that application developers trample all over. Where there is no one person who understands completely how things work even in their interactions, there is little urgency around coordinating each process to maximize business value. There is also no way to operate the technology in a way that it will perform as expected. You cannot deliver something to the business if you cannot control its end-to-end operation.

For those of you schooled in the philosophy, "you don't have to be technical to manage technology," I have bad news for you. The common tactic of trying to get away with managers who are non-technical exacerbates the situation because they cannot tell when they are being schmoozed (schmoozed used in this sense, for you non-technologists, is a technical term for misleading someone as to the true nature of a problematic situation to make it look like you are not accountable while technically telling the truth). In information security review, I have seen numerous examples of technology teams insisting that a security requirement was impossible to work into a project because they did not know how some component with which they interacted worked. It is impossible for a non-technical manager to know in these cases whether someone is honestly ignorant of the technology or just trying to cut back on the amount of work they have to do. Disregard for technical skill in management, combined with technologists' disinterest in technology outside of their scope, creates a community of technologists that are immune to motivation to advance. Technologists spend their time honing the technical skills they think will make them marketable, while project managers are left to figure out how to deliver products to the business.

For example, I was speaking with an application developer who bemoaned how idealistic her users were because they wanted to see her application recognize the caller ID of an incoming call and display the associated customer record on her existing screen. She knew it was technically possible to use software integrating the PBX system with caller-ID database lookup but the thought of having it as a project requirement was daunting. For those of you who think this type of integration is fungible technology, I have some more bad news. It takes an extremely technical project manager to put together the right tasks to ensure coordination of specifications across multiple teams in voice, network, and software technology, especially when you consider that you cannot count on major vendors to program to protocol specifications. Add the fact that one is expected to define the work at a level such that it can be outsourced, and it is no wonder that most technology projects are over time and budget.

So you are embarking on a career in IT management and you know to succeed, you need business alignment. What should you do?

1. Be clear on business requirements. Don't sacrifice them without an all-out thinking out of the box exercises that involves expertise from a wide variety of technology areas.
2. Do your best to ensure technologists understand those requirements and the urgency of the business goals that are expected to be met via their implementation. Do your best to ensure that the success of the business process will result in success for them personally as well.
3. Don't try to deliver something you do not understand. It will be impossible for you as a manager to control, and impossible for your staff to support.

The key to managing technology and technologists to serve the business is to be able to understand that there are an extremely wide variety of choices with which to meet business requirements and to choose the easiest to implement, control, and operate.