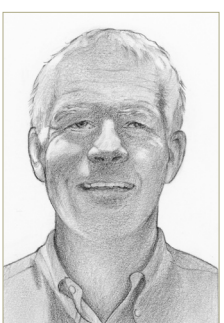
The Extent of Our Ignorance

Learning to manage what we don't know

A recent book, *Risk Intelligence: Learning to Manage What We Don't Know* by David Apgar (Harvard Business School Press, 2006), provides fresh insights on the general business of managing risks.

Apgar divides all risks into two categories, those that are learnable (or knowable) and those that are random (or unknowable). Accidents and natural disasters are random or unknowable risks. So are many financial risks because they depend on factors such as security prices that are determined by what economists call "complete" markets that already incorporate all the relevant available information. Learnable risks, on the other hand, can be reduced through study; the more we know about them, the more effectively we can manage them.



By Jerry Peterson

This provides a useful point of view on how to manage the risks of large-scale enterprise resource planning (ERP) implementations or other big information systems projects, which are notoriously prone to problems. Essentially all of the risks relevant to major systems projects are learnable risks; they result not from random factors beyond our control but from gaps in our information, understanding, or skills. What we don't know can hurt us, so we need to think about where we lack knowledge and how we can get it.

Consider one of the most common problems in ERP programs—a lack of organizational readiness. This may stem from inadequate communication, insufficient incentives, or incomplete training. The extent of these situations is knowable by interviewing or surveying the affected people. Knowing the issues, we can do something about them.

Or take another common problem—poor quality in the legacy data to be loaded into a new ERP system. We can easily measure the completeness, currency, or accuracy of the data through simple audits or by sampling techniques. There is also the problem of slow response time, something that many ERP programs experience at launch and which may stem from capacity shortfalls, design issues, or other sources. Such problems can be predicted by modeling, load testing, and other similar functions.

Of course, there are limits to this approach. Studies take time and resources, and an organization may not have the skills or experience to carry them out. Then, too, some things are too complex to be analyzed in full detail. Experienced IT people know that testing can never be completely comprehensive; there are too many potential paths through the programs, too many combinations of input variables, and too many different operating conditions.

These limits are practical ones; they result from limits on time and resources and therefore imply the need to prioritize. We need to assess carefully where to place our risk management efforts and whether we need to call on outside expertise. This assessment should start with an initial triage of risks, evaluating potential problems both on the likelihood that they will occur and how much harm they will cause if they do. The highest priority should go to addressing risks that are potentially the most severe and about which we know the least. For these high priority risks, we then strive to deepen our knowledge and use it to formulate risk mitigation actions.

Knowing the issues, we can do something about them.

The key to risk management in ERP programs (and many other business programs) is to focus on what we don't know.

As the philosopher Confucius said 2,500 years ago: "Real knowledge is to know the extent of one's ignorance." ■

Jerry Peterson is the director of the Information Systems Executive Forum at University of Michigan's Ross School of Business, a consultant with the Cutter Consortium, and the retired director of information technology services with Ford Motor Company. Reach him at gpetersn@umich.edu.

Read online: www.symantec.com/ciodigest/thinktank/peterson