Too big to fail?

By Hugh W. Ryan

Don't bet on it. A hefty price tag is no guarantee that your large-scale information system will be completed on time and within budget—or even function properly. Indeed, chances are that it won't, which is why IT project risk management has become so critical. Here are a number of field-tested approaches to doing it right.

Imagine if the construction industry had to confess that it could complete only 8 out of every 100 buildings successfully each year. Imagine if you overheard carpenters working on your own house casually discussing the fact that they were spending almost three hours out of every workday just fixing things that some other carpenter had done wrong. Imagine embarking on a major construction project only to be told by the person running it that the $100,000 cost you had agreed upon could just as easily end up at $200,000 or $300,000.

Someone else's nightmare? Don't count on it. Substitute "information systems industry" for "construction industry," and you have the true story of how a number of companies Accenture talked to around the world have found their major information technology development projects going not just wrong, but *seriously* wrong.

The actual statistics are certainly enough to keep you awake. According to industry figures:

- Only 8 percent of large-scale applications projects (those that cost between $6 million and $10 million) succeed.
- Among all IT development projects, only 16 percent occur within acceptable constraints of cost, time and quality.
- Cost overruns of anywhere from 100 percent to 200 percent are common in software projects.
- The price tag for annual cost overruns for IT projects has been estimated at $59 billion in the United States alone. Another study puts the figure at $100 billion.
- IT workers spend more than 34 percent of their time just fixing software bugs.

It's no wonder, then, given these numbers, that information systems risk management has become a critical issue for executives around the world.

Annual cost overruns for IT projects have been estimated at $59 billion in the United States alone.

The cost of IT failure has skyrocketed in recent years as development projects have become more and more complex. If companies do not find new ways to manage IT projects and control risks, they will be unable to embrace important new initiatives that can give them an edge in the marketplace. Clearly, the environment for large projects is not going to get simpler; the only thing to do is ensure that the tools and approaches for dealing with risk can move beyond traditional tracking to a more sophisticated and comprehensive approach.

Twin goals

These large IT projects have both a tactical goal and a strategic goal. The tactical goal is to deliver what the business users want, on time and at the cost they expect. The strategic goal is to make sure that a company gets all the business benefits these projects offer.

Think of managing IT risk as something that applies to both of those goals. If you don't, you may be like the fisherman in *The Old Man and the Sea:* You may land the big fish, but in the end it might not be worth anything.

We call the projects we're talking about here "large complex systems." That phrase can refer not only to what is being built or developed—which usually includes an enterprise-wide, mission-critical information system—but also to the project itself, which exhibits all the quirky behaviors of what the science of "systems thinking" calls "complex adaptive systems" (see sidebar, page 66).

Large complex systems generally have the following characteristics:

1. *A longer time frame.* Because of the scope of the work, a project may last from three to five years—far longer than traditional systems development projects. This longer project horizon creates additional challenges because business, the marketplace and the technological environment will all change during the life of the project. Even the executives sponsoring the project are likely to move on to other jobs at some point during the project. Hence, in addition to a long-range goal, the project must also have the shorter-term capacity to adapt quickly to change.

2. *Wider impact.* Today's large projects are not just systems projects, affecting only a limited group of users. They are far more likely to be closely tied to the larger strategy and processes of the enterprise. Complexity also has increased dramatically as these systems link an enterprise with its customers, suppliers and other business partners. Finally, the human factors within the project become just as critical as the technological ones because success today depends, to a very great extent, on changing the behaviors of everyone involved—the workforce, management, suppliers and customers.

More than common sense

Dealing with risk on a large complex systems project depends first on simply acknowledging that things are different. Some of the approaches for dealing with risk on a smaller scale still apply, with some adjustments, to this new environment.

Many, however, do not. One CIO recently argued that effective management of these projects, including risk management, "is more good old-fashioned common sense than anything else." Maybe that was true at one point, but it's not anymore. There are too many things to track, too many chains of events whereby a seemingly insignificant fault can grow into a problem with the potential for derailing the entire project.

Risk management can be tedious and administrative. But a highly visible program of risk management that has the commitment of everyone, especially in the boardroom, is highly correlated with the successful completion of a project. What is success? Success means that something is delivered on time, is within budget guidelines and does what the business community said it wanted the project to do.

What we have found, above all, is that effective risk management is not any one thing. It's a suite of things that includes project management skills, planning skills, some new human resources approaches, a great deal of experience, different ways of managing the entire program and the testing process, and different approaches to rollout. Finally, it requires new kinds of tools that go beyond the traditional linear understanding of a project and can deal with complex, multiple layers of cause and effect.

Fieldwork

Over the course of the past several years, we have developed a number of new tools and techniques to manage IT risk more effectively and to improve our ability to do costing, estimating, metrics and project tracking. But we have also gone into the field to companies that have undertaken large complex systems development projects. We have had in-depth discussions with them not only about where things went well but also about where things didn't go as well. Although these companies have been assured of confidentiality, we can share some common themes coming out of these discussions.

1. *A business vision.* Time and again we heard that successful large complex systems projects begin with a vision of a potential new way of doing business. Moreover, the entire project is guided by whether or not a new initiative or a change in scope is faithful to that vision.

How complicated does the vision need to be? Often, not very. In fact, the simpler the better. At one international stock exchange, for example, the business vision was simply a crisp articulation of the eight essential capabilities that the final system was to provide. But that "simple" vision carried a great deal of weight, primarily because the project team spent considerable time up front getting senior executive support for the vision.

Eventually the vision was integrated into the project training and was displayed in all of the project's essential documents. More important, all projects within the larger program of initiatives had to be linked to the vision and justified according to how they specifically would help make the vision a reality. The main issue here is that the vision has to be not only clearly expressed but adhered to with a great deal of tenacity.

2. *Senior management support.* Strong management support is vital to the success of large projects. As a project increases in size, it requires support from different management levels.

For somewhat smaller projects—say, involving up to 200 people—executive sponsorship has to come from someone who reports directly to the CEO, such as the COO. As the project gets bigger—perhaps involving more than 300 people—you had better make sure it has direct, sustained support from the CEO. Changes in leadership are not un-usual during the extended life of a large complex systems project. One project at a major communications and high-tech company was living proof of this risk: There were five CIOs during the life of the program.

The human factors within the project become just as critical as the technological ones.

Dynamic tools to manage risk

Organizations still like to think in linear, mechanical ways: A project starts on this date and follows a path to the end. Along the way are various checkpoints to make sure the project is going well.

Most automated project management tools simply are more sophisticated versions of this linear approach. But a large complex systems project behaves in a manner that makes linear tracking less than optimal (see story).

The science of "systems thinking" tells us that these projects are more like the system of the human body, made up of many components at many different levels. If we wish to influence or manage such a system, we have to be aware that any action taken will be felt at every level and hence will have multiple consequences.

The problem is that the human brain, for all its marvels, can't effectively track the myriad combinations of cause and effect in such a system. That's where computer modeling comes in. In 1999 we developed a new tool—Accenture Risk Manager—that uses computer simulation techniques to create a computerized, customized model of a particular project environment.

The model can be manipulated so that it corresponds to the particular project you're working on. The tool takes into account traditional project management data—staffing, time lines, budgets and so forth. But it also includes a number of frequently overlooked factors that our research has shown can often present the most significant risks—intangibles like vision, burnout, learning curve and sponsorship.

When the model has been customized, you then run a simulation of the project over its life cycle. Output screens give you a sense of how your project may perform in terms of schedule and budget. Based on that output, you can then return to the model and play "what if," altering various factors (in essence, simulating different decisions you might make as a project manager).

What's the primary benefit of the tool? Most current approaches to managing risk are primarily good at dealing with a problem once it occurs. Computer simulation allows us to anticipate a risk *before* it becomes a serious problem. Risk Manager is also an extremely important training tool for new projects.

Risk Manager doesn't predict exactly what will happen. It doesn't make your decisions for you. But we find it to be an extremely effective tool for generating the kinds of discussions among project teams and executives that are vital to bringing a project home on time and on budget.

A second tool we have developed—the Accenture Engineering Management Facility—addresses another major mistake companies make with their large complex systems projects: underestimating the amount of work necessary in testing.

"We're behind, but we'll make it up in the testing phase" is an often overheard phrase, and it couldn't be further from the truth. Testing is probably the most underestimated and most often overlooked activity in large complex systems initiatives. In fact, testing in these environments can account for 50 percent to 80 percent of the entire effort.

Why is testing so difficult? Again, the answer is because of the intricate cause-and-effect relationships in a large complex system. You need to understand how each part of the system relates to other parts.

Traditional manual methods employed in testing are being stretched to the breaking point. A manual method often has the testing requirements in a spreadsheet or document, the test conditions in another spreadsheet and the source code in a source code management repository. Ultimately, it is difficult to bring all these documents together to gain a full understanding of the interrelationships.

The Accenture Engineering Management Facility automates and tracks many of the areas related to testing, including requirements management, test plan management, defect tracking, configuration management and reporting. The tool itself is a custom integration of proven third-party software packages.

The advantages of being more rigorous around these testing activities is that it creates more efficient processes, more predictable results and a clearer understanding of the interrelationships among parts of the system. Certainly, this automated tracking capability ultimately translates into cost savings. But it also translates, more broadly, into a large complex system that delivers on its promise, with reduced risk and broader business benefits.

How can you mitigate the risk of losing your sponsor? One effective solution that we've observed at several companies is to make sure your work gets buy-in, not just from top executives but also from those who report to them. The reason? If the project sponsor moves on, the new person taking the spot will have been briefed already on the importance of the project. Even if someone from the outside is brought in, all the direct reports will be prepared to explain their support for the initiative.

3. *Skilled personnel and experienced leadership.* As one resources executive from our field studies says bluntly, "You can't take people who have done only 'regular' projects before and make a large complex systems project work. You need people experienced with doing this kind of project, true team players, who are willing to innovate."

Our field studies suggest that ideally, about one-fourth of the senior-level program executives will have had previous experience with large jobs. At one communications company we reviewed, very few of the workers from the project's lower ranks had previous experience with large jobs. As a consequence, the amount of work that needed to be redone (called *rework*) during the middle phases of the project was higher than it should have been.

Staffing increased each month from the first day of the project up until three months before the system went live. The budget overrun was directly attributable, company executives believe, to a shortage of appropriately experienced personnel.

4. *Stable architectures.* All the significant efforts surveyed involved the use of common or shared architectures. A successful effort, however, depended on the early definition and delivery of a stable common architecture. Significant changes to the data, application or technical architectures had severe negative effects on the timeliness of project deliverables—and on the reliability of what was delivered.

Sometimes the ability to work toward this stability can be simply serendipitous. At two of the companies we studied, a stock exchange and a social services agency, there was an unanticipated lag time while senior management decided whether to proceed with the project. Because of that, various teams working on architecture definition and design had several months' grace to define, design and begin the implementation of the required architectures. Although the architectures evolved with the business and technology needs, they remained largely consistent with the initial design. This consistency proved to be essential to the timely delivery of the applications.

Not every project will be able to achieve a stable architecture that quickly. However, the ability to achieve this stability decreases the risk of problems and rework.

5. *Phased release plan.* Successful large complex systems projects generally are rolled out in a series of releases, as opposed to a "big bang" strategy where the whole thing comes up live over a weekend. Only two of the cases we studied attempted a big bang release. One of them encountered some difficulties. The other was successful but required exceptional planning and cooperation by the company as the conversion approached. The company has since come to believe that a phased release approach would have been more desirable.

Why is phased delivery so vital to the management of risk? This approach serves a number of specific functions

Effective risk management requires new kinds of tools that go beyond the traditional linear understanding of a project.

Successful projects have found that risk management must be a formal function within the larger program.

in addition to its general benefit of reducing risk.

- *Early verification:* A phased approach permits early verification by the business users of essential components as they work with the system in their businesses.

- *Ability to make mid-course corrections:* Inherent in the phased release approach is the ability to review the overall situation as the releases are rolled out and then to make mid-course changes.

- *Closer user involvement:* Phased releases show users an immediate business impact; user feedback then provides valuable information for future releases.

6. *"Engineering management": Integrating program and testing management.* A large complex systems development effort can best be thought of as a set of parallel projects that share interdependent deliverables. For example, in an insurance company environment, an underwriting project will provide the policies for a separate claims project. Such parallel projects require a dedicated effort to manage them as a whole.

This effort represents a new type of management that is a synthesis of program and testing management—something we call "engineering management." The program management effort is to ensure that the parallel projects are proceeding at the pace required and at the cost expected. The testing management is to ensure that the interdependent deliverables meet quality expectations.

At one telecommunications company we studied, for example, there is a very formal and comprehensive approach to engineering management. This company uses an approach that is so rigorous that it allows the IT function to provide a new release every three months—far faster than is typical. Although this release rate presents change management challenges to the users, the fact remains that by following a very rigorous approach, the company is pushing out releases at a remarkable rate.

7. *Deliverables-based management.* A success factor that has emerged from later field reviews is a management style that combines long-term visionary planning with rigorous definition and reporting on short-term deliverables and milestones.

At one high-tech manufacturer, for example, teams defined deliverables at the beginning of each phase of the project. However, the level of detail and exit criteria (that is, the criteria that tell you when a deliverable is complete) for early design deliverables varied. The result was that too much extra effort was needed later in the project to identify missed requirements and to finish rework on deliverables that were scheduled to have been previously completed.

In contrast, at a travel and transportation company we studied, a program management office infrastructure was established early in the project. The team employed a simple but effective technique to manage deliverables at a very detailed level, which gave management the ability to anticipate complications early on and to keep pace with progress. This project came in right on schedule and budget.

8. *Distributed risk management.* Although every success factor we have identified in large projects has something to do with risk management, successful projects have found that risk management must be a formal function within the larger program.

That is, dealing with risk must begin with a centralized function, but then it must also be "distributed" throughout the entire project.

For example, at a large ERP installation for one manufacturing company, we helped create an executive steering committee made up of company vice presidents, each of whom took ownership for a part of the implementation. The committee met twice each month to track progress. In addition, a project "war room" was established early on, where project team leaders would gather three times a week to raise issues and discuss risks. (As the project neared completion, three times a week became twice a day.)

Whether a company turns to an external service provider for assistance in undertaking a large complex systems project or chooses to fly solo, it will want to distill its knowledge of effective project, program and risk management into a capability that affects how the company recruits, trains and appraises its project leaders.

Effective management is a distinct talent. As with any talent, some people will have a more natural inclination toward it than others.

All people, however, can learn how to be more effective managers of risk. And if those in the information systems industry expect to improve on the current 92 percent failure rate on IT projects, they must. ■

Hugh Ryan, a partner in Chicago, leads the Large Complex Systems program at Accenture. He has worked extensively in the fields of information technology architecture development, client/server technology, groupware, knowledge management technologies and in the emerging area of netcentric computing. Mr. Ryan is an occasional columnist for the *Journal of Information Systems Management.*

hugh.w.ryan@accenture.com