

# Is Software Development Risk Costing You Money?

By [ExecutiveBrief](#) Staff

*Poor software project management often means missed deadlines, cost overruns or even outright failure of the project. How can your company avoid this industry-wide problem? In our brief you'll learn best practices for successfully completing software projects.*

Software development projects are plagued with risk and impending failure. According to The Standish Group's 2006 CHAOS Report, only about one-third (35 percent) of the researched software development projects undertaken in the previous two years were considered "successful"—that is, they met all requirements and were completed on time and within budget. Almost half (46 percent) were reported to be "challenged," meaning that they were completed and operational, but were delivered late, went over budget, or did not support all of the features originally required. And 19 percent were considered outright failures or cancelled prior to completion. According to the same report, 41 cents of every dollar spent on software development was considered wasted.

Anyone involved with software development projects knows why bringing them to successful completion can be so difficult to achieve. The most frequently cited factors that contribute to the challenge include:

- **Poor communication.** The legendary communication gap between developers and business clients or stakeholders often leads to poorly defined requirements, which in turn will lead to inadequately designed software that doesn't provide the functionality needed by the customer.
- **Time.** Because market forces today are continually changing—and at such a fast pace, the longer a development project takes to get off the ground and cycle through to completion, the more likely it will be that the software will fail to meet the current needs of users. Requirements and priorities can change multiple times as projects progress through planning, development, testing, and production.
- **Scope.** Poor communication, time creep, and intense market competition lead to scope creep, where requirements are frequently changed and added to the project without a proper evaluation of their true importance and without corresponding increases in resources, budget, or schedule. A project whose scope is too broad will become too complex, exceed allotted costs, overrun schedule, and ultimately fail. A project whose scope has been defined too narrowly may finish on time, but likely will fail to perform according to customers' real requirements.
- **Unorganized development processes.** Despite the availability of established best practices in software development, such as the U.S. Software Engineering Institute's (SEI's) Capability Maturity Model Integration (CMMI), many development departments don't follow a coherent development methodology. A study by the SEI in 2005 found that, when it came to the quality of their development processes, almost 40 percent of companies gave themselves a low rating of 1 or 2 out of a possible 5 on the CMMI scale.
- **Budget, resources, and schedule.** Software development is rife with unrealistically low budgets, inadequate resources, and limited time.
- Success for any IT project, particularly in software development, is very much dependent on effective management of risks, both early on in the project and throughout its lifespan. Not long ago, risk was defined almost exclusively in negative terms, but in the past seven years or so, risk management philosophy has expanded to include unexpected opportunities. "Risk can be defined as uncertainty that matters," says David Hillson, director of Risk Doctor and Partners, a well-known global consulting firm specializing in project risk management. "Uncertainties, when they happen, can damage a project, but some uncertainties can help."

For example, new hires can bring new expertise that takes a project in a positive new direction. A company could discover reusable code that it didn't know it had. Certain parts of a project could finish early, opening up possibilities that seemed impossible at first. "You can either rely on good luck or be proactive and seek out those opportunities."

There is more than one approach to risk management. The most widely known is probably the one detailed in A Guide to the Project Management Body of Knowledge (PMBOK Guide) published by the Project Management Institute (PMI), considered by many to be the project management bible. Like any religious text, the PMBOK Guide has its legions of followers as well as its fervent

detractors, and alternative methodologies abound. But most approaches to risk management follow steps similar to those outlined below.

### **Step #1: Identify**

Identification of project risks is usually accomplished via a brainstorming session that includes the development team and the stakeholders. Including stakeholders in this process is essential for fostering good communication and gaining a true understanding of the business risks associated with the project. For example, the company's financial calendar may influence when certain accounting or other financial-related systems must be in place. Including stakeholder representatives is also a great way to gauge their skills, availability, and authority to make decisions for their part of the project. In the case of software development, many of the risks analyzed will stem from the factors outlined above (i.e., communications, time, scope...).

Risk experts cite two principles for the success of this step. The first is to avoid an unfocused, free-for-all discussion. "The meeting really must be facilitated and structured," says Hillson. The PMBOK Guide includes a risk breakdown structure (RBS) that can help the team to map out risks in a very detailed, hierarchical fashion. Another useful tool for gauging internal and external treats and opportunities is the SWOT (strengths, weaknesses, opportunities, and threats) analysis. It's also helpful to generate and repurpose checklists of risks from similar past projects to help map out risks for the current one.

The second principle for success in the identification step is to ensure that the meeting doesn't become an empty exercise done solely to comply with company policy. "My problem with this process is that people often go through the steps without any real thinking," says Gartner analyst Donna Fitzgerald. "They end up filling out the forms, but missing the real substance." Fitzgerald points out that company politics play a big role in project risk and that risk mitigation often entails measures that lie outside the original project. These factors should not be ignored.

Hillson agrees. "The whole idea of this process is to be innovative and think outside the box. Real risk management is about exceptions to the normal day job."

### **Step #2: Quantify and Prioritize**

Once the identified risks are mapped out, the team should prioritize them based on probability and expected impact. Certain risks can kill a project all by themselves; even if the probability of such risks is low, they should be assigned a high priority. Risks that have low impact and low probability often can be ignored. This is also a good time to map out specific early warning signs for each of these risks.

### **Step #3: Address**

The next step is to develop strategies for addressing each risk. Some of the classic strategies include:

- Avoid it. Sometimes it's possible to find a strategy for avoiding a risk completely. For example, if there's a risk that a certain supplier will deliver an essential item too late to meet your deadline, it may be possible to use a different supplier.
- Minimize it. Some ways to minimize risks include lining up contingency plans and funding, bringing in team members who have the expertise to take over a key task if other members quit, or nurturing a positive relationship with people outside of the team whose cooperation is essential for success. "No one person should be so indispensable that losing this resource would mean a significant delay for the project," says Fitzgerald. In one of her projects, Fitzgerald had to add \$80,000 to the budget in order to hire a consultant who could take over if an unhappy critical team member quit.

"In every organization, there's an off-org chart network of people who get things done," says Fitzgerald. "Build your team so that you are tied into that network from the start." Fitzgerald also points out the importance of putting together a team of people who are known to be flexible and eager to adjust to new circumstances.

- Transfer it. Sometimes you can hand over responsibility for certain risks to a vendor, an outside consultant, or even a stakeholder.
- Accept it. If the impact and probability are minimal, it may not be worth addressing the risk at all.

Again, it's always a good idea to keep records of successful strategies from past projects that could be useful for addressing the risks of a current one. This is also the time to assign each risk to an owner who will be responsible for detecting and addressing it.

#### **Step # 4: Monitor**

Murphy's Law reigns in just about every project. Frequent risk reviews should be conducted to continually monitor for early warning signs of risks, assess progress in addressing identified risks, close out some risks, and discover new risks that have cropped up. Ongoing communication with stakeholders about a project's progress and shifting risk profile is important as well.

This is also the time to assess whether the project may be in real trouble. Fitzgerald points out some key early warning signs, including a high number of unresolved issues, persistent disagreements about what should be done, and early slippage in the schedule. Any of these circumstances may indicate a need to reassess your entire risk management and project implementation strategy.

#### **Step #5: Record**

Post-mortem reviews can be extremely valuable in planning for successful future projects. Conduct an end-to-end evaluation of the project, from conception to delivery. Compare the risks that you originally identified with the real risks that you ultimately needed to address. Which strategies worked, which didn't, and what would you do differently next time? Record this information and use it to create checklists that can be referenced during your next initiative.

#### **Consider Agile Development Techniques**

In recent years, agile development has risen in popularity as a way to minimize risks related to time, scope, and communication. Rather than undertaking massive software projects in their entirety, agile development methods promote development of one small complete piece at a time. "Software is evolutionary by nature, which is why projects that last longer than six months have a very high failure rate," says Fitzgerald. "Get in, and get something done in less than six months. Then, when the state of affairs changes, do something else."

#### **Get Real**

A structured risk-management process is vital to success, but it's also important to align the structure to the size and type of project and to your company philosophy. "Your risk process should be scalable," says Hillson. "A small project probably requires a light touch, perhaps an hour-long meeting every couple of weeks." On the other hand, a large project at a large company with many players and stakeholders very well could require a highly structured, detailed approach to managing risk. Perhaps more important, however, is to cultivate an organizational culture in which everyone understands and lives the concepts of risk management instinctively. Fitzgerald offers this suggestion: "Your company should adopt an attitude that likens risk management to breathing."

#### **© ExecutiveBrief 2008**

**ExecutiveBrief**, the technology management resource for business leaders, offers proven tips, techniques, and action plans that companies can use to better manage people, processes and tools – the keys to improving their business performance.

To learn more, visit us at: [www.executivebrief.com](http://www.executivebrief.com)

