

Project Termination Doesn't Equal Project Failure

Barry Boehm, University of Southern California

One of the most frequently cited software project statistics comes from the Standish Group's 1995 Chaos report (<http://www.standishgroup.com/visitor/chaos.htm>): "A staggering 31.1 percent of [software] projects will be canceled before they ever get completed." The Chaos report, and numerous documents citing it, label these canceled projects as "failed" and imply that all 31.1 percent of them were canceled because of poor software management.

A FALSE IMPRESSION

This implication is both false and hazardous. It is false because, particularly in an era of rapid change, a lot of software projects are properly started, well managed, and properly terminated before completion because their original assumptions have changed. It is hazardous because it often leaves software managers with the following temptation: "It's becoming clear that continuing this project will waste company resources. I should probably have the project canceled now, but that would make me the manager of a failed project and wreck my career. I'll be better off if I say nothing, keep the project going, and look for a new project to transfer to."

To counter this train of thought, I review the main sources of project termination determined in the Chaos re-

port, and estimate how likely each termination source applies to a well- or poorly managed project. I base these estimates on my personal involvement in five or six digital library projects per year,



Good reasons exist for putting projects on the chopping block before they're finished. Not all stem from poor management.

view of about 20 term papers per year on failed industry projects, and reviews of terminated or failing projects developed with some of the Center for Software Engineering's 30 affiliates.

Project termination sources

Here, I list the top 10 sources of canceled projects reported in the 1995 Chaos survey of IT executive managers. Each source shows a percentage that indicates how many failures it accounts for of all the failures reported in the survey.

Incomplete requirements (13.1 percent). Terminations from this source are more likely to come from mismanaged projects that blunder ahead without a clear idea of the operational stakeholders' needs and priorities. But some ter-

minations come from well-managed projects when project managers recognize that the operational stakeholders are never going to agree on or commit to a clear requirements statement.

Lack of user involvement (12.4 percent).

This project termination source is almost equally split between projects that fail to communicate with users and users that fail to communicate with projects, no matter how hard the project managers try. In the latter case, it is generally good practice for the organization to terminate the project, because it is highly unlikely that its product will satisfy users.

Lack of resources (10.6 percent). Budget cuts, downsizing, and reprioritized projects are common. A project terminated for any of these reasons probably has relatively low potential business value, regardless of how well it is managed, although poor management is more likely to decrease a project's business value.

Unrealistic expectations (9.9 percent).

Projects with unrealistic expectations are also about equally likely to be

- poorly managed projects that fail to validate the feasibility of satisfying user expectations, or
- well-managed projects that try to validate feasibility and find unavoidable factors—such as immature technology, overhyped COTS products, or a saturated market—that justify a project's prompt termination.

Lack of executive support (9.3 percent).

Most often, these are mismanaged projects whose managers made and failed to verify unrealistic assumptions about executive support. For example, project managers might have relied on executives reprioritizing other initiatives to support their project's needs. Frequently, however, organizations can terminate well-

managed projects when they replace an executive with someone who has different agendas and priorities.

Changing requirements (8.7 percent). Most terminations due to changing requirements stem from unrealistic project acceptance of scope changes without accompanying budget and schedule changes. This is still a habit of many unskilled project managers. However, well-managed projects in a climate of rapid change may need termination when the continuing costs of adapting to change outweigh the benefits of having the capabilities. For example, the sponsors of several multimedia projects have withdrawn their commitment to a project because of rapidly changing COTS products.

Lack of planning (8.1 percent). Lack of planning shows up in mismanaged projects, which are terminated when it is discovered that the project managers have no idea of where they are or when they'll finish.

Absence of need (7.5 percent). In an era of rapid change, these terminations are equally likely to occur in well- and poorly managed projects. Good managers are more likely to track trends in needs, and recognize and terminate unneeded projects earlier.

Lack of IT management (6.2 percent). Projects lacking adequate management are by definition mismanaged projects.

Technology illiteracy (4.3 percent). Technological illiteracy of performers or managers can sink projects. These are usually mismanaged, shouldn't-have-started-in-the-first-place projects.

IS 31 PERCENT TERMINATION REALLY "STAGGERING?"

From the preceding analysis, project termination clearly does not always indicate project mismanagement and failure. Most of the top sources of termination apply about equally to well- and poorly managed projects. And good managers can detect an infeasible project earlier and terminate it without wasting as many resources.

So, in many cases, a precompletion termination rate of 31 percent might not be staggering. In fact, in our current and future world of rapid technology, organizational, and marketplace changes, 31

percent might not be high enough.

For example, on the last 10 rapid-development, Web-based, USC digital library projects, four projects were not transitioned into operational use because of rapid changes in technology, organization, or marketplace factors. One was not used because the sponsor organization (for project-external reasons) changed its standard platform from Unix to Windows at

A *Computerworld* article cites common failure or termination rates of 40 percent for new product introductions.

about the time of delivery. Another was not used because its sponsoring manager/user was transferred; her replacement had other priorities and did not want to commit the personal time necessary to make it an operational success. Two more applications were built on top of a potentially strong COTS product that the vendor later de-emphasized. When the USC Library recognized this change, it dropped its commitment to using the COTS product, although one project fielded a partial capability on another platform.

In more stable software applications domains, a 31 percent precompletion termination rate might be considered high. But there is great similarity between leading-edge Web-based software application domains and other market sectors, where a 31 percent termination rate would be considered surprisingly low. Frank Hayes cites common failure or termination rates of 40 percent for new product introductions and about 50 percent for sales leads ("Managing User Expectations," *Computerworld*, Nov. 3, 1997).

Even for in-house software development, a leading-edge application that requires users to change work styles is more like a new product introduction or a product that must be marketed and sold. Expecting such projects to succeed 100 percent of the time does two bad things. First, it sets an unrealistically high standard for software project managers

to achieve. And second, it creates a climate in which neither developers nor sponsors see the need for significant marketing or user-preparation activities as key to project success.

SUCCESSFUL PROJECT TERMINATIONS

What can you do to make software project terminations healthy and cost-effective? Here are five primary steps.

Market your project's products

The main recommendation in Hayes' *Computerworld* column was for software projects to better manage user expectations via marketing practices such as

- conducting user satisfaction and needs surveys;
- holding internal technology fairs;
- ensuring sustained user involvement in product definition, prototype exercise, and project reviews; and
- stating clearly that unanticipated changes might make it advisable to terminate or redirect the project.

Use risk management and incremental commitment

There are three ways to handle project risks. One is *risk avoidance*, in which you reduce the project's scope, at some cost in product performance or competitiveness. Another is *risk transfer* to a trusted or warranted third party. The third is *risk acceptance*, in which you accept that the project may involve risks. You then apply risk assessment and control techniques to manage risks, including the option of early termination if risk exposure becomes unacceptably high. Using a risk acceptance approach, a certain fraction of project terminations becomes an acceptable way of coping with an unpredictable world.

The risk-driven spiral model provides an effective process framework for software risk assessment and control. Its spiral cycles also provide a good framework for *incremental commitment* of project resources one cycle at a time in a way analogous to a poker game: Rather than putting all your chips on the table at the beginning, you assess your hidden cards

and the cards shown on the table to determine whether to drop out (terminate the project) or to incrementally add a few chips to the pot to see your next card (Barry Boehm, "Spiral Development: Experience, Principles, and Refinements," Tech. Report CMU/SEI-2000-SR-008, July 2000; <http://www.sei.cmu.edu/cbs/spiral2000/Boehm>).

It can take some adjustment to realize that terminating projects can be natural and even healthy.

Use architecture review boards and feasibility rationales

The Architecture Review Board (ARB) process is a commercial best practice originated by AT&T ("Best Current Practices: Software Architecture Validation," AT&T, Murray Hill, N.J., 1993). It focuses the organization's best technical, management, and user talent on a thorough review of a project's architectural and business feasibility at two critical stakeholder commitment points. The first review point is a *discovery review*, which checks for project feasibility and stakeholder commitment to develop a definitive product architecture and life cycle plan. Later, the organization conducts an *architecture review*, which assesses project feasibility and the commitment to use and support that stakeholders have for the architecture and plan.

The discovery and architecture reviews correspond to the life cycle objectives and life cycle architecture milestones in the Rational Unified Process (Philippe Kruchten, *The Rational Unified Process: An Introduction*, Addison Wesley Longman, Cambridge, Mass., 1999) and the USC Model-Based (System) Architecting and Software Engineering (MBASE) method. MBASE also includes a Feasibility Rationale Description for providing the best possible justification of a project's technical and business feasibility. The ARB can review this document

to determine project continuance, redirection, or termination. Details are in the spiral-development technical report mentioned earlier.

Monitor business assumptions

Frequently, organizations establish projects on the basis of assumptions that may or may not remain true with time. Examples are, "We'll be first to market with this type of product." Or, "The COTS vendor will remain committed to provide the additional features we need," and "The operations group will be revamping their facilities, equipment, data, workflows, and operator preparation to fit the software system we're developing." If any of these assumptions becomes seriously invalid, it's essential to get project stakeholders together for a continue-or-terminate review. The DMR Benefits Realization Approach (John Thorp, *The Information Paradox*, McGraw-Hill, New York, 1998) provides an excellent framework and set of procedures for monitoring business assumptions.

Don't equate project termination with project failure

You need to identify and terminate infeasible projects early. Sending a message to project managers that project termination threatens their career will tempt them to continue projects that should die.

It can take some adjustment to realize that terminating projects can be natural and even healthy. If you don't try some risky projects, you'll lose your competitive edge. But you shouldn't expect all your risky projects to succeed. *

Barry Boehm is director of the University of Southern California's Center for Software Engineering. Contact him at boehm@sunset.usc.edu.

Editor: Barry Boehm, Computer Science Department, University of Southern California, Los Angeles, CA 90089; boehm@sunset.usc.edu