Column: Nov 24, 2003

The Only Winning Move

By Dave Gelperin

Summary: Organizations that don't expect and encourage about 20 percent of their projects to be stopped may either be wasting resources or implementing only the safest bets. Denying reality or playing it safe may be the riskiest strategy of all.

"*A strange game. The only winning move is not to play.*"
— Joshua, an AI system in the 1983 movie War Games

At the end of War Games, Joshua has gained full control of America's nuclear strike capability. He has just finished an in-depth analysis of the likely outcomes of various attack scenarios. The "strange game" Joshua refers to is Global Thermonuclear War.

Joshua's insight also applies to some software projects, where the only winning move is not to continue to play i.e., to stop (suspend or cancel) the project. In "Project Termination Doesn't Equal Project Failure" [IEEE Computer Sept. 2000 or http://www.cs.unc.edu/~welch/class/comp145/media/docs/Boehm_Term_NE_Fail.pdf], Barry Boehm argues that cancellation of some projects should be a normal and expected occurrence in well-managed software development organizations. He says that, in a world of rapid change, a 30 percent project cancellation rate may not be high enough.

There are many reasons to stop a project, including:

- Environment -- *New government initiatives or controls may change priorities*.
- Business -- *The introduction of new products by competitors may redefine goals*.
- Organization -- *Business reengineering or integration with other organizations may change priorities*.
- Technology -- *New technologies may make design strategies obsolete*.
- Personnel -- *Critical stakeholders may not be sufficiently involved or may have serious disagreements with each other, or developer capability may have been misjudged*.

Innovation is bred in organizations that encourage risk-taking, that expect some stopped projects, and that implement a management strategy of incremental commitment. Incremental commitment anticipates change and learning.

Projects are marked by multiple review-points where project status and justification are reevaluated. Projects can be reauthorized, suspended, or cancelled at any review-point. The ability to stop projects before completion is a mechanism for managing quality while taking risks.

As with any human decision, we can get it wrong by stopping work on a project that should continue. More often however, for an array of psychological, social, and cultural reasons, projects continue that should be stopped. We need stakeholders to buy-in to stopping, just as we need them to buy-in to the project. Some projects, like some government programs, seem to develop a life of their own.

Failure to stop a deserving project can result in:

- insufficient resources for priority projects
- projects that deliver nothing or little-used systems
- turnover of key staff, because working on failed projects is a major cause of extreme frustration and burn-out

Organizations that don't expect and encourage about 20 percent of their projects to be stopped may not be taking enough chances or may be wasting significant resources. High-payoff projects are often risky, but "playing it safe" may mean a significant loss of opportunity (e.g., market share to risk-savvy competitors) and may be the riskiest strategy of all.

Reader Comments

At a major New York insurance company a few years back when we were setting up the processes, we included a "get out of project free" card. After design and before coding begins, the participants at that point - PM, architect, customers, QA, etc. - meet to discuss whether to continue with the project or cancel it and take the losses incurred to date. Because the concept was sold to mgt, it was a positive thing to come away with a cancelation - no one lost face or reputation. Instead of blaming someone for starting the project in the first place, people were congratulated for saving the company from more losses. Since we worked in an increasingly "re-usable everything" environment, not everything was a loss, and the lessons learned were priceless. -steve blais....(11/25/03) *Steven Blais*

Author's Response: You all understood that the point of the column was to advocate incremental commitment and "successful stopping", but some expressed dissatisfaction with "% of cancelled projects" as a useful indicator and caused me to think more deeply about measuring project risk and termination.

As a result, I propose that development directors ask for the following information on a yearly basis: (1) What % of our total yearly project value falls into each of the 5 project risk categories? This assumes that each project has an associated measure of its value to the enterprise and contains a risk assessment task as part of its planning process. This question provides a profile of risk undertaken. (2) What % of the project value in each category completed successfully, completed unsuccessfully, stopped successfully, stopped unsuccessfully? This distinguishes between failure and success and between completion and stopping before the end. Ed provides two examples of what I think of as "successful stopping" -- one at the end of requirements and one at the end of design. This question provides a profile of outcomes by risk category. (3) For each project, I would want (honest) answers to two questions: a. Should we have started this project based on what we knew at the time? b. Should we have stopped sooner? This assumes that some form of project retrospective will provide quality indicators for our project management judgments. I hope you find this approach more useful

Author's Response: After project initiation, the three most important justification review points follow needs assessment, system specification, and architectural design. Each of these activities may provide significant new insight into project feasibility and lead to "successful stopping".

Any later activity may uncover "nasty" surprises that require a justification review, but this should be an exceptional occurrence. You mentioned that you have rarely experienced successful stopping, but spoke of termination caused by unrealistic goals and company politics. These causes of project failure sit atop Barry Boehm's list of top 10 software project risks. (1) Personnel shortfalls (2) Unrealistic schedules and budgets (3) Developing the wrong functions and properties (4) Developing the wrong user interface (5)

Gold-plating (6) Continuing stream of requirements changes (7) Shortfalls in externally furnished components (8) Shortfall in externally performed tasks (9) Real-time performance shortfalls (10) Straining computer-science capabilities. This list complements my list of reasons for successful stopping.