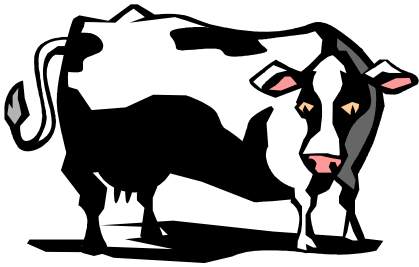


# Acquisition Archetypes

Changing Counterproductive Behaviors in Real Acquisitions

## Feeding the Sacred Cow



### Toil and Trouble

Most programs gain momentum as time passes. Some, though, take on a life of their own—after a number of milestones pass, and teams expend time and effort, they seem self-propelled, unstoppable. They're woven into an organization's existence. They seem privileged, too, despite the fact they often have yet to show any value to customers or stakeholders. These few "precious" programs become sacred cows: they are fed, protected from harm, and are often revered. They are beyond reproach.

*"Sometimes ... well-established programs ... are not tolerant of even healthy criticism."*

But all systems also have issues—and it is healthy to raise risks and problems with a system in development (especially major ones). That way all concerned become aware of the issues, and can commit to finding the best possible resolutions. Unfortunately,

sacred cow projects are neither subject to nor tolerant of even healthy criticism and dissent. In one real-life example, a program was fielding an IT business system to a network of field offices. The project was several years into its timeline and nearing its initial beta test deployment, yet had long before become largely off-limits to any active questioning by the organization.

### Problems? What Problems?

For example, even as stakeholders attempted to raise issues during development, their questions—and their acts of criticism—were rebuffed by project staff and managers. The project team became increasingly defensive:

- User concerns about creating a centralized system architecture with a single point of failure were dismissed.
- Disagreements with choosing the second busiest site in the nation as a beta site were ignored.
- Concerns over rushing to a cut-over date before the system was ready were downplayed.
- Criticism from whistle blowers, the media, and Congress about serious issues after initial rollout began elicited only defensiveness.
- Program team members characterized disgruntled users as incompetent or computer illiterate—warning other critics to back off.

This cow could moo.

### Hey! Keep Feeding Me!

As these and other risks arose, the program office and the contractor repeatedly deflected them. In continuing to receive funding—and continuing to

throw money at the project—they blamed others, or shrugged the problems aside as irrelevant. Any questioning of or disagreement with the program's direction or approach met unresponsiveness or hostility. This single-minded support of the program, even at the expense of the stake-

*"Six months after 'go live' became a disaster, the contractor still denied there were significant technical problems."*

holders' interests, marked a form of defensiveness by the PMO and contractor. It affected the objectivity of the decisions being made, and the program proceeded on its obdurate path. Decisions believed by many to be fatally flawed went unchallenged, yielding only further development investment to implement those decisions.

### It's The User's Fault

Six months after "go live" became a disaster, the contractor still denied there were any significant technical problems with the system—that it was entirely a case of user incompetence. Of course, the sacred cow still was fed—substantial time and effort continued to be invested in system development.

(Continued on page 2)

## The Bigger Picture

The general phenomenon of *escalation in decision making*—of which this archetype is an example—is widely recognized, and described as “persistence with a venture beyond an economically defensible point” [Drummond 1996]. Decision making on large, high-visibility programs becomes less technically objective and more politically defensible as time passes. Various factors may come into play in producing the effect, including uncertainty regarding the outcome of the program, poor visibility into program progress and status, sunk cost, prior decisions, personal self-interest, and ego- and face-saving.

As the diagram illustrates, various *System Issues* continue to arise during development. These are dealt with through a series of *Effective System Investments*, maintaining equilibrium within a balancing loop. However, increasing *System Issues* produce *Criticism of [the] System*, which then drives up the level of *Personal Investment [and] Defensiveness*, reducing *Objectivity* on the part of the decision-makers and reducing the *Quality of Investment Decisions*. The lower *Quality of Investment Decisions* in turn reduces the value of the *Effective System Investment*. This creates a reinforcing loop that surrounds and can ultimately overwhelm the original balancing loop by increasing *System Issues* in a continuing cycle.

A key aspect of this dynamic is the loss of objectivity on the part of the decision makers. They have become too close to the project to be impartial, making them unable to assess the true feasibility of the system. It may be due to ego or stubbornness [Flowers 1996]. Regardless, the results are likely to include overly optimistic status reports and a “desire to commit more resources to improve things.”

## Breaking The Pattern

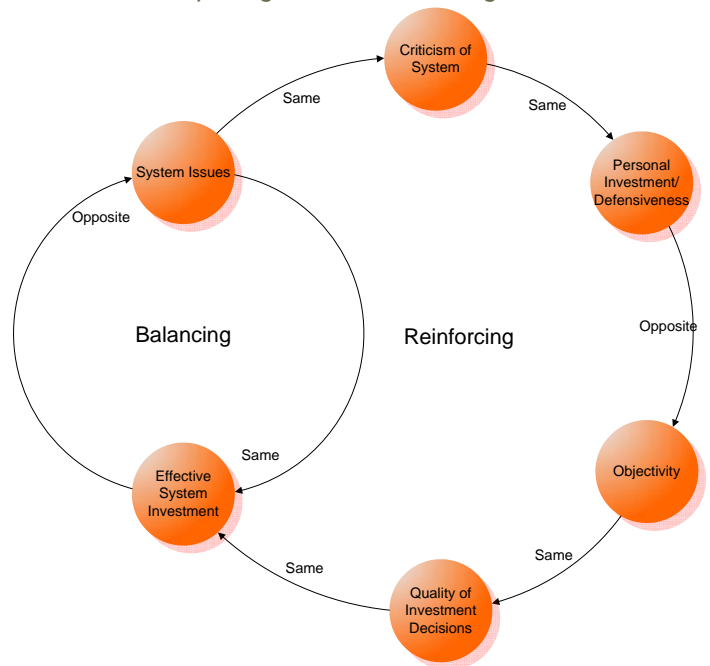
Recovering from “Feeding the Sacred Cow” requires recognition that the counterproductive behavior is taking place—recognition by the very people who are embedded in the dynamic and have lost the ability to make objective, rational program decisions. If “Feeding the Sacred Cow” has taken hold of the program, a significant change in management may be necessary to “reset” personal factors such as self-interest, ego, and face-saving.

Another key step in recovery is to conduct a series of formal, objective reviews with external technical experts to identify and address issues. This on-going review process will bring the original program goals and assumptions

[Drummond 1996] Drummond, Helga. “The politics of risk: trials and tribulations of the Taurus project.” *Journal of Information Technology II* (1996): 347-357.

[Flowers 1996] Flowers, Stephen. *Software Failure: Management Failure*. New York, N.Y.: John Wiley & Sons, 1996.

A Causal Loop Diagram of the Feeding the Sacred Cow



System variables (nodes) affect one another (shown by arrows): Same means variables move in the same direction; opposite means the variables move in opposite directions. Balancing loops converge on a stable value; Reinforcing loops are always increasing or always decreasing. Delay denotes actual time delays.

back into focus, test them for continued feasibility, and help decision makers make rational choices.

To help prevent this escalation behavior from taking hold, several steps should be taken:

- Actively encourage dissenting opinions; don’t shoot the messenger. Honest, objective resistance to the program can help solve problems early, when the chance of resolution is greatest. To leverage dissent, establish a formal process to raise, review, negotiate, and resolve issues in a way that stakeholders can agree is fair.
- Let technical rationality rather than political considerations guide decision making. Planning regular, technical program reviews (such as the one described earlier) is one good step to take.
- At a minimum, regularly review and question the original assumptions behind the decision to develop the system. Are they still true? This is a great preventive measure. Determine if it’s still possible to move forward, if a change in direction is needed, or if the original rationale has changed such that the program is no longer relevant.