

Acquisition Archetypes

Changing Counterproductive Behaviors in Real Acquisitions

The Bow Wave Effect

A Never-Ending Project

This is a true story—and one you’ve probably heard before. That’s the point. It’s about a pattern of failure, an archetype.

A government program needed to replace an aging COBOL mainframe financial system—one so old that the costs of maintaining its obsolete hardware multiplied each year. The only people who could maintain it were now retiring, taking their knowledge with them. Yet the replacement project was stuck in low gear: time dragged on, the focus of the program shifted, deadlines were missed. The sponsors became increasingly anxious. It had, as one team member said, “... drifted, moved, and waddled, and done everything but die.”

Finally, with the CIO under increasing political pressure to show IT results, the absolute, final deadline was set—just 18 months away.

How Bow Waves Begin

Could the development team get it done? Yes—but only if they kept to schedule and stuck to the project plan. And that’s not what happened. Instead, the *bow wave* pattern of failure stalked the project almost from the start. Fueled by the accumulated effects of an educated guess (SWAG) estimation process, the project picked up baggage, rather than momentum.

“[Requirements] were prioritized, and they got SWAGs, and they drew a line based on available resources,” lamented a team member. “They approved [requirements] before they were costed. Some things moved from re-

lease to release if they fell below the priority line.”

This practice of deferral sent ripples—bow waves—through the project. It wasn’t done maliciously or even consciously by the project teams. Quite the opposite. The effect was the end result of accumulated decisions that seemed right and expedient at the time.

The project managers didn’t recognize the problem, or understand that the bow wave is, unfortunately, a common pattern in software development programs. Deferred or dropped functionality and system requirements

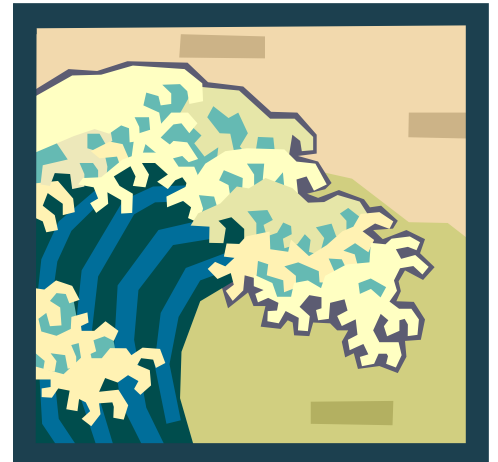
accumulate, piling up in front of the project in a wave that washes up over schedules and budgets, endangering delivery and project success.

Often, as in this project, the bow wave puts project teams in an impossible schedule squeeze.

“We don’t compromise on schedule delivery date,” noted a contractor manager, “and don’t compromise on quality, and can’t add staff, so the only variable is scope—we just kept dropping functionality. But eventually that meant we couldn’t handle all the records, and that meant we weren’t allowed to convert them [from the legacy system], and so the whole final delivery schedule got blown out of the water.” A growing mass of work had to be done at the very end—when risk was highest, and the deadline left no margin for further schedule slip.

It was “a three-year program in its 13th year.”

“We’re trying to put 8 pounds of slop into a 5 pound bag.”



Complexity Feeds The Wave

A number of errors fed the project’s bow wave. Perhaps the most damaging one was failing to account for effects of complexity.

The team used a sequential development process, paralleling the system they were assembling: they built the initial processing modules first

(handling the less complex records), and left the final modules (processing the most complex records) for last.

However, because no one really ana-

lyzed the complexity of the final modules during the planning phase, or the handling of the most complex records, the program didn’t accurately estimate the feasibility or the effort of completing these tasks.

As you might have guessed by now, the team didn’t meet its delivery deadline and at last report was still struggling with completing the final, most complex processing module.

The Bigger Picture

The bow wave effect is an example of a dynamic where a problem is solved with a “quick fix” that gives immediate results, but only temporarily solves the original problem (see diagram, right) [Kim 93]. The organization often knows that a more fundamental solution would be better in the long run, but feels unable to wait while such a solution is put in place. Over time the organization comes to rely on the quick fix, not realizing that it is undermining their ability to implement the fundamental long-term solution they need.

Illustration: Biting Off More Than You Can Chew

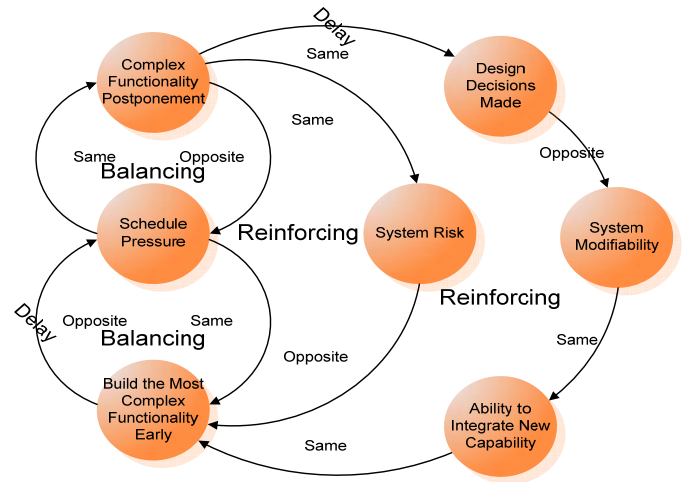
There is a pattern in evolutionary project development in which programs repeatedly fail to estimate correctly the amount of work that can be done with the resources and time available. Programs fail in this key step for many reasons, including a lack of estimation ability or historical data; the lack of a consistent development productivity rate; or deliberate underestimation of the effort needed so as to make the estimate meet the preconceived “correct” result.

The outcome is the bow wave effect—the result of deferring the development of functionality from one increment or spiral to the next, where there is now even more functionality to implement, so the problem recurs in the next increment.

Variation: Kicking the Can Down the Road

A complicating problem in spiral development is delay in tackling the hardest, riskiest tasks. Complex or risky tasks should be planned for an early iteration to reduce risk, but are often postponed until a later spiral. This makes the program’s cost and schedule performance look better in the near-term, but increases the risk in later spirals by delaying complex work for which there is now less available schedule, and less modifiability in other parts of the system to accommodate changes (see diagram, right).

A Causal Loop Diagram of the bow wave effect.



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.

Real-life Bow Waves

Here’s an example of bow wave behavior that many of us can identify with: a person who has trouble managing their finances, and so continually uses their credit card to maintain their standard of living instead of either reducing their spending or increasing their income. Over time they become dependent on the credit cards, but the increasing interest payments on their growing debt begin to undermine their ability to implement the appropriate long-term solution—balancing their budget.

Breaking The Pattern

How can a program recognize its own bow wave? By looking at how functionality was originally allocated to releases, seeing what has been deferred, and comparing that to the anticipated complexity, maturity, or risk of that functionality. Looking at work completed versus work remaining (and checking for consistency with schedule) also can highlight a bow wave.

To break out of the bow wave dynamic, a program must first understand the cause of the original problem (schedule pressure, in our example) that leads to the expedient solution, and re-examine the other possible solutions—especially those that are fundamental. However, making the distinction between “expedient” and “fundamental” solutions isn’t always clear-cut.

Once an appropriate fundamental solution is identified, the organization must then assess its current ability to implement that fundamental solution (Is there enough time left? Do we still have the right skill set?), understanding that their ability may have eroded due to their use of the expedient solution.

[Kim 93] Kim, Daniel H. *System Archetypes: Diagnosing Systemic Issues and Designing High-Leverage Interventions*, Vols. I, II, and III. Pegasus Communications, 1993.