



Intellyx White Paper

Avoiding the Sunk Cost Fallacy in the Mainframe Shop

Jason Bloomberg

April 4, 2016

In today's fast-paced digital world, the enterprises who still depend upon mainframes to run their business increasingly find themselves in a battle between the 'old way' and the 'new way' of doing things.

Existing mainframe teams, of course, tend to fall onto the 'old way' side of this equation. In many cases the people responsible for these systems are late in their careers, and there's a broad perception that such graybeards are set in their ways and resistant to change – a reputation that may or may not be deserved.

However, this common belief that all mainframe people are ready to retire is becoming less true by the day, as a newer generation of professional gradually takes their place. A far more common situation is for a mainframe organization to contain a mix of people of different ages, skill levels, and backgrounds.

As a result, the 'old way' thinking – often expressed as 'that's the way we've always done things' – is increasingly under pressure, as new business priorities as well as fresh blood bring new thinking to the mainframe organization.

And yet, the old ways are stubborn and tenacious. IT decision makers need better ways to make appropriate decisions when 'old way' thinking is getting in the way of progress – without falling into the trap of disregarding the valuable input of the most senior members of their technical staff.

An important starting point: recognizing when the 'sunk cost' fallacy is in play. This fallacy – also known as the 'good money after bad' fallacy – crops up in enterprise decision making with appalling regularity. Recognizing it for the fallacy it is can be an important first step to resolving many 'old way' thinking roadblocks.

The Sunk Cost Fallacy in Action

Here's how the fallacy works. Let's say you have a difficult challenge with two possible solutions, A and B. Either one would solve the problem equally well.

You've already sunk ten million dollars into A, and if you spend another million on A you'll solve the problem.

Option B, however, you haven't tried yet. However, you learn that you can solve the same problem by spending a hundred thousand dollars on option B.

Which is the better option? The correct answer, of course, is B, because it will only cost 10% as much as A to solve the problem.

In real world scenarios, however, people tend to select A – because so much money has already been spent on that option. You wouldn't want to change horses in midstream, the argument goes. You also wouldn't want to feel like you had wasted so much money on A, right?

Such thinking, however, is fallacious – and making the mistake of basing a decision on a sunk cost can be very expensive for your organization.

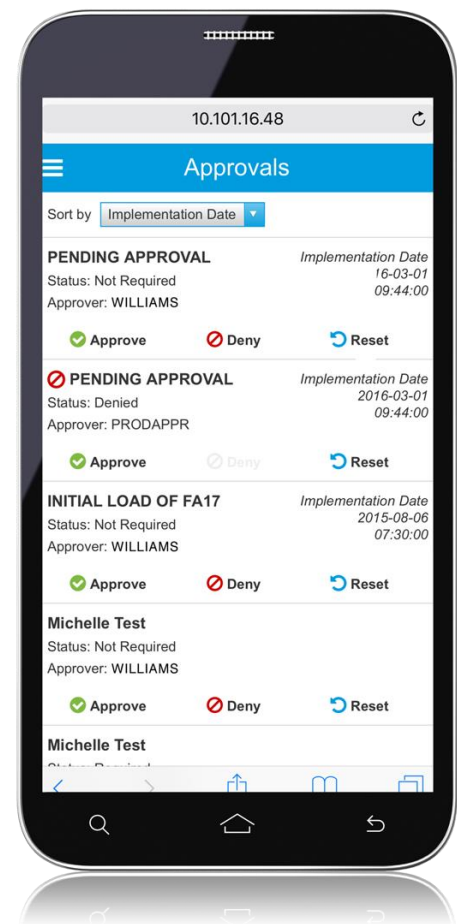
Targeting the Sunk Cost Fallacy in the Mainframe World

Many of the tools that mainframe developers and operators use daily they've been working with for years. Then a new, better tool comes along that will cost less (in both money and time) starting today.

Let's use software change management/source code management (SCM) tools as an example. One of the standard tools in this space is [Endevor](#) from CA Technologies. While it's firmly established in the shops that use it, there's no question that Endevor's technology is dated, and furthermore, tied to now-obsolete waterfall methodologies.

In fact, Endevor was never meant to work out of the box the way modern software is expected to. Instead, organizations had to spend many long person-hours crafting custom processors – essentially, custom programs – without which Endevor would never work.

Today, Endevor shops have already put in all the effort needed to get Endevor up and running. But then along comes a better



ISPW's Approvals Window (Source: Compuware)

tool like [Compuware's ISPW](#). While in Endeavor, even basic commands like Move and Delete require processors, whereas in ISPW, developers simply enter their configurations into a panel.

If an Endeavor shop could rely upon the status quo staying the same, that would be one thing – but today's reality is very different. New business pressures are driving entire IT organizations to move more quickly, and even the mainframe shop must now participate in an increasingly Agile-savvy, digital world.

Endeavor, unfortunately, comes from a different age – and Endeavor shops have built several custom processors over the years to get things to work. Every change, therefore, requires laborious changes to existing processors or the complex creation of new ones.

Furthermore, even managing processors is a time-consuming task, further putting up roadblocks to change. Endeavor shops end up coding entire solutions by hand in order to work around Endeavor's limitations.

With ISPW, in contrast, changes are a simple matter of updating the panel, and anyone on the team can quickly understand and make a change. As a result, ISPW dramatically reduces the time and effort necessary for administration and management of the source code environment.

It might seem, therefore, that switching from Endeavor to ISPW is an obvious decision – but in many cases, the decision is not nearly as clear-cut as it should be. The reason? *We've put so much effort and money into Endeavor up to this point that we are loathe to make a change.*

In other words, the argument for staying with Endeavor falls into the sunk cost fallacy trap. Such reasoning isn't simply a matter of opinion – it is an *irrational* conclusion based upon flawed logic.

Instead, decision makers should base their decisions on the costs in terms of both money and time *starting today*. Given Endeavor's massive inflexibility and time commitment, ISPW is a more rational choice moving forward.

**GIVEN ENDEAVOR'S MASSIVE
INFLEXIBILITY AND TIME
COMMITMENT, ISPW IS A
MORE RATIONAL CHOICE
MOVING FORWARD.**

The Intellyx Take

In the 1980's, Victor Kiam, the owner of the Remington Products shaver manufacturer, became famous for his catchphrase in TV commercials: "I liked the shaver so much, I bought the company."

CEO Chris O'Malley and his team at Compuware could say the same thing about ISPW.

As a mainframe tools vendor, Compuware develops code on the mainframe daily – perhaps far more than the typical enterprise mainframe shop ever would. Furthermore, for years Compuware was an Endeavor customer.

As Compuware transitioned from a waterfall to Agile development model, however, Endeavor couldn't keep up. The tool simply didn't fit into any kind of scrum or Kanban model for Agile development



Compuware might have attempted. They had to make a change – so they resisted the lure of the sunk cost fallacy and moved to ISPW for their SCM needs.

In a matter of months they were able to move to a fully Agile model – and now they are able to release updates to their products on a quarterly basis, a frequency previously unheard of in the mainframe world.

And yes, Compuware liked ISPW so much it bought the company.

Compuware is an [Intellyx](#) client. At the time of writing, CA Technologies is also an Intellyx client. No other organizations mentioned in this article are Intellyx clients. Intellyx retains full editorial control over the content of this article.