

Organizational Memory Failure Brings Loss of Control over I.T.

Conrad Weisert, Information Disciplines, Inc., Chicago
September 1, 2002

Generation gaps

More and more I.T. organizations are discovering that problems they thought they had solved a decade or two ago are challenging their current staffs. Some organizations merely waste resources in rediscovering old solutions, but they're the fortunate ones. Others are paying huge ongoing penalties, as they never regain the benefits of the solutions they once had.

Some loss of organizational memory arises naturally through the turnover of people. Younger staff members were never aware of the issues, while their older colleagues have retired or moved on to other roles in the organization.

The obvious remedy is a permanent repository of the ongoing collected wisdom of the organization. Written policies, standards, guidelines, procedures, tools, and techniques, when kept up-to-date, serve as the core of organizational memory. A younger staff member consulting such material discovers how the organization has already dealt with some issue. If that solution seems obsolete or inappropriate in today's environment, then he or she is free to propose an update or replacement solution.

Throwing wisdom away

The trouble is that many permanent repositories of information turn out not to be permanent after all.

In some cases the material lacks the structure to support retrieval. A chronological series of technical bulletins, for example, has a short useful life.

In other cases an organization deliberately discards part or all of its written foundation. Such an action is often taken by a newly appointed manager who doesn't understand the value of the information and is eager to repudiate a symbol of "red-tape bureaucracy" from the discredited previous regime. It can also be the result of hasty, unplanned decentralization of I.T. responsibility.

A costly example from the past

The so-called "Y2K crisis" imposed huge costs around the world. Although many organizations who suffered those costs had been naive in ignoring date representation until it was too late, others had established by 1980 date-representation standards, only to see them thrown away after a reorganization or a change in management. They ultimately faced 6- or 7-figure projects to assure that their internally developed application software would work after December 1999.

A costly example from the present

Many large system development projects are going through a serious *users' requirements crisis*. Their organizations have forgotten how to gather, document, review, and approve specifications of what a proposed new application system is to do. Some of them, concluding that rigorous users' requirements are hopelessly elusive, are giving up, falling back on *incremental* strategies under which reliable estimates of time and cost are impossible.¹

Opportunistic fad methodology gurus and CASE² tool purveyors are exploiting their confusion, by disparaging what they characterize as "traditional methods". Following their lead, new generation systems analysts and managers are naively embracing requirements documentation techniques that didn't work in the 1960s, such as unstructured *want lists* and *sequential narratives*. The results are the same today as they were back then.

What can you do?

Whenever we propose to establish some new bit of support infrastructure, such as a standards manual, we're obliged to put forth a justification, usually in the form of return on investment (ROI) analysis. On the other hand when we propose to abolish or dismantle existing support infrastructure, no such justification is prepared. We expect upper management to view our action as a simple cost saving, an elimination of bureaucratic overhead.

That means that a manager is rarely held accountable for the negative consequences of destroying organizational memory, and may even get rewarded in the short term for doing so.

If you're in upper management you must start demanding a stronger burden of justification for eliminating infrastructure that was once justified, especially any body of written material. If a high-level manager had asked in 1986: "Why don't we need this date representation standard any more?" his or her organization would have avoided the agony of costly Y2K retrofitting. If anyone had asked in 1992: "Why do we no longer need guidance on structured requirements documentation?" a lot of large projects that are now in deep trouble might be making acceptable progress.

¹ Last year an award winning entrepreneur, explaining to a forum of fellow executives that a \$5 million original estimate had turned into a \$30 million project, advised his audience that "software development costs are unpredictable.

² Computer assisted software engineering