The Software Revolution

During the past decade, a revolution has occurred in the way large private-sector organizations acquire the software that underpins their daily operations and generates the data they need to make better long-term decisions. Rather than building such software themselves, or paying someone else to build it according to their specifications, many—and in some sectors, most—organizations have elected to purchase large Enterprise Resource Planning (ERP) systems. Well-known ERP vendors, including SAP, Baan, PeopleSoft, and Oracle, have grown rapidly as a result. Although their growth has slowed temporarily as prospective customers batten down the technological hatches for the transition to Y2K, these ERP companies are riding the crest of a wave driven by economic and technical factors that make their continued success likely.

Michael R. Vitale is Professor in the Centre for Management of Information Technology, Melbourne Business School, University of Melbourne. Earlier in his career he was the Director of Administrative Computing at Dartmouth College and later Associate Professor at Harvard Business School.
Some academic institutions, including MIT, have also purchased ERP systems, but other institutions continue to struggle with homegrown administrative software that is often more expensive, riskier, and less functional that what is available on the market. The usual excuse for undertaking work not in an effort involved is that the institution’s processes and procedures are so special that they must be supported by custom-built systems. The premise of this article is that most institutions would be much better off buying software and then adapting the built-in best-practice processes that come with the packages. There is simply no easy way for institutions to dilute their resources, and distract their managers’ attention, by focusing on software development rather than their primary educational mission.

Software Development as an Inherited Disease

More than forty years after mankind began writing software, the difficulties on software projects remain dismal. The typical software development project will take on average twice as long as budgeted, and without all of the promised functionality. And that’s the good news—many projects deliver nothing at all. The failed management of software development, including the failure of CASMAC, noted by the Australian auditor-general’s comments, quoted above, regarding the management processes of the UniPower universities, is probably indicative of the overall level of process quality in many colleges and universities. Yet these same processes are sometimes so well established that it will cost tens of millions of dollars to be even able to modify the requirements of a given software system to do anything more than run it. The software market is richly populated with customizable packages, vendors are willing to agree to pay costs estimated at another spent almost A$5 million. By mid-1999, at least two of the UniPower universities had purchased ERP systems to carry out the tasks they had hoped to perform via CASMAC. The story of CASMAC is unusual only because it is better documented than most other software failures. The overall lesson is clear: software development remains a difficult, expensive, and risky undertaking, even when a professional software development organization is involved. Most universities have no business doing it.

The Objections

The primary objections to purchasing software are, first, that it is expensive and people cannot make any modifications to the source code. This is a valid concern, at significant cost, by software vendors. The Objections, which in any case would simply refuse to issue an elaborative or detailed request for proposal (RFP). The software developed by vendors is likely to be small, and if the university follows the lessons learned, at significant cost, by private institutions, it will not attempt to make any modifications to the source code of the package. This approach starts in the software market, rather than the university. It begins with the observation that the commercial software market has begun to develop an industry of detailed solutions that is readily available. The economics of commercial software developers are well understood and heavily favored vendors who can sell multiple copies of a given package, rather than developing one-off solutions. The primary objections to purchased software are, first, that it is expensive and people cannot make any modifications to the source code. This is a valid concern, at significant cost, by software vendors. The software market is richly populated with customizable packages, vendors are willing to agree to pay costs estimated at another spent almost A$5 million. By mid-1999, at least two of the UniPower universities had purchased ERP systems to carry out the tasks they had hoped to perform via CASMAC. The story of CASMAC is unusual only because it is better documented than most other software failures. The overall lesson is clear: software development remains a difficult, expensive, and risky undertaking, even when a professional software development organization is involved. Most universities have no business doing it.

The Alternatives

The CAUSE side of EDUCAUSE began as the College and University Systems Exchange, founded in the United States in 1971 to facilitate the sharing of administrative software among institutions of higher education. By 1972 CAUSE had created a software library on magnetic tape and purchased cards and had begun to move into more current software. The CAUSE side of EDUCAUSE began as the College and University Systems Exchange, founded in the United States in 1971 to facilitate the sharing of administrative software among institutions of higher education. By 1972 CAUSE had created a software library on magnetic tape and purchased cards and had begun to move into more current software. The CAUSE side of EDUCAUSE began as the College and University Systems Exchange, founded in the United States in 1971 to facilitate the sharing of administrative software among institutions of higher education. By 1972 CAUSE had created a software library on magnetic tape and purchased cards and had begun to move into more current software. The CAUSE side of EDUCAUSE began as the College and University Systems Exchange, founded in the United States in 1971 to facilitate the sharing of administrative software among institutions of higher education. By 1972 CAUSE had created a software library on magnetic tape and purchased cards and had begun to move into more current software.

The benefit for most colleges and universities is to find the prewritten software package that best fits their information technology architecture, their strategy, and their way of doing business.

The better option for most colleges and universities is to find the prewritten software package that best fits their information technology architecture, their strategy, and their way of doing business.