

This article appeared in the  
December 2004/January 2005 issue of



Subscribe instantly at  
[www.bijonline.com](http://www.bijonline.com)

- Free in the U.S.
- \$48 per year in Canada and Mexico
- \$96 per year everywhere else

# ENTERPRISE INTEGRITY

## The Next SOA

marketing focuses more on creating buzz than on conveying business value. It's so bad, that this marketing behavior has become a joke: How many CMOs (Chief Marketing Officers) does it take to create a TLA (Three-Letter Acronym)? (Get it?) And, increasingly, even if an engineering innovation, a product's primary value is to solve a technical problem, achieving new business value only by reducing the IT costs such as TCO (Total Cost of Ownership).

In computing's early decades, IT was a scarce, extremely expensive resource. This dictated its restricted use to highly repeatable business activities triggered by predictable business events. Considerable care was taken to drive IT efforts from business requirements, encouraging cost-effective IT use. This was achieved with some efficacy by business analysts, a profession for which one could be trained and employed well into the '80s. Unfortunately, as software technology became increasingly obscure "under the covers" and simultaneously and superficially easy to use and understand, the likelihood of business requirements miscommunication from user to developer increased disproportionately.

The problem of obtaining proper requirements specifications resulted, in part, from the communication gap that developed between users and developers. That gap has had two components, only one of which was recognized (albeit inaccurately). The first was a "post office syndrome"—as the number of transitions involved in gathering and communicating requirements increases, the number of possible (and likely) mistranslations grows exponentially. The need to reduce communication barriers helped fuel the fervor for prototyping, object-oriented, and model-driven software development approaches. Unfortunately, it also eliminated the business analyst's role without conveying his or her knowledge of the business or interviewing subtleties, to the armies of software designers and developers now involved in requirements gathering. One result: Strategically important improvement opportunities often go unrecognized and unaddressed by IT during iterative prototyping.

The second gap component is more subtle, arising because IT too often gathers requirements directly from business users. Although senior business and IT management consider budgeting and authorization, they generally step aside for requirements gathering. In the early days of IT, this was reasonable. Software was used to automate highly repeatable operational activities and perceived as having little to offer strategic business activities. Indeed, the role of IT in

**IT** must reformulate its goals to survive, and software vendors had better pay attention. Too often, and for a variety of reasons, software

managerial decision support was minimal. All this has changed: Software is more sophisticated and can now not only support, but strongly affect, strategic business activities.

Without IT, businesses have little hope of maintaining the frantic pace demanded by drivers such as regulatory compliance, global competition, real-time enterprise, and so on, ad nauseam. Unfortunately, existing IT infrastructures and how businesses establish IT requirements are both mired in costs that result from a high inertia past badly evolved. The window of opportunity for businesses is now too short to reintroduce a long, costly, error-prone analysis and design phase. IT must somehow faithfully support strategic business decisions and requirements (without introducing technological buckshot), and in near real-time.

We've long promoted managing IT as a provider of business services to its business clients, and Service-Oriented Architecture (SOA) has the potential for being the conduit of those services. Unfortunately, first-generation SOA efforts have focused almost entirely on providing a conduit for IT services rather than business services. Treating Web Services as the *sine qua non* of SOA is hardly defensible as a business requirement. There are, in fact, good technical reasons not to implement every service as a Web Service—and why should business users have to care?

Few technical standards are motivated by business requirements (read Web Services standards if you doubt), being heavily driven by technical and vendor agendas. Nonetheless, advocates repeatedly—sometimes intentionally—confuse business terms with recently invented technical terms. Business event, transaction, activity, and process (and its variants such as orchestration, choreography, and coordination), as used in Web Services, carry tremendous, constraining technological baggage not assumed in the corresponding business terms.

We can't really analyze and design business requirements unless IT uses business language appropriately. But once it does, SOA can transition beyond IT services and model-driven architectures to a Business Services *Orchestration* Architecture, with IT as administrator. Business modeling tools should enable a model-driven business's decision-makers to convey real-time business requirements to IT while managing decisions. That goal is IT's only hope for survival in a world where *enterprise integrity* is affirmed or denied by every material business decision. **bij**

### About the Author

David McGoveran is president of Alternative Technologies. He has more than 25 years of experience with mission-critical applications and has authored numerous technical articles on application integration.

e-Mail: [mcgoveran@bijonline.com](mailto:mcgoveran@bijonline.com)

Website: [www.alternativetech.com](http://www.alternativetech.com)



BY DAVID MCGOVERAN