

# Fire the Architects!

By Marek Jakubik

Those who know me well may be quite surprised by this rather dramatic call for dismissal. Indeed, over the many years of my career as IT manager and executive, I have spent immeasurable amounts of energy supporting the principles of IT architecture and its projects. Then, inevitably, I must have become older and wiser.

It is a typical cycle of human evolution, isn't it? From idealistic, youthful exuberance, to a wise, balanced pragmatism of older years. And so, today I'll share with you what I learned about the otherwise elegant and sensible ideas promoted by IT architects. More important, I'll share my views on how to "rehire the architects," i.e., define an environment in which architects effectively and consistently can create value.

"Consistently" is an important word here. I have no doubt many people can point to examples of architecture-based approaches to IT that did create a positive value for their enterprises. In my experience, however, these examples must be viewed as the oases of success in an otherwise dry and barren landscape cultivated by IT architects.

## Why IT Architects Fail

A fundamental problem for many architecture programs is they are void of a basic definition—specifically, what is architecture, and why are we doing this? In many companies, architecture remains a mystery even to its practitioners (some, perhaps, prefer it that way) and certainly to its alleged beneficiaries. As a result, the value of architecture continually is questioned.

Very few architecture teams show the capacity and willingness to communicate clearly the value of their programs. The challenge is magnified by the high degree of complexity and abstraction inherent to this matter. After all, architecture deals with a complex web of interactions between large numbers of computers, networks, systems, data, and users. And, since most of these interactions are non-sensory, it is very difficult to represent and explain their nature to people.

Architects create drawings and scaled-down models to allow others to visualize their concepts. By comparison, data architects use something called entity-relationship diagrams, while process

architects' visualization tools include use-case diagrams and Universal Modeling Language scripts. Don't you want to say, "huh?" right now?

To be sure, the science and practice of IT architecture is making progress. Still, the nature of tools, methods, and the underlying skills is overly rooted in technical analysis rather than based on a deep understanding of business processes and business drivers.

Ergo, rule no. 1: One cannot be a good architect without a) an excellent understanding of the business and b) the ability and willingness to invest in communicating the nature and value of IT architecture to an enterprise.

Vendors do not help either. Following IBM's domination of the architectural domain that ended in the '90s, several large vendors—most notably Microsoft, IBM, Sun, Oracle, and SAP—are locked in an ongoing battle to occupy the land of architectural standards. And although it is exceedingly unlikely any one of them ever will dominate the others, the battle will continue. The reason is even a small piece of architectural terrain claimed by a particular vendor can be converted into a juicy increment in its profit margins.

Hence, rule no. 2 and a corollary: Beware of proprietary vendors' solutions. If you haven't yet, take a serious look at Open Standards, especially those related to infrastructure. One reason to do so is some of them have matured to be a very viable alternative (e.g., Linux) to proprietary solutions. The other is the "anti-proprietary" message such a strategy sends to the IT vendors.

The final and most difficult hurdle architecture units struggle with is lack of a proven economic model to justify broad investments in architectures. This point is likely the single most important cause of IT's inability to sell architectural programs.

To address this hurdle, I'm going to use an analogy from a discipline where architectures actually do deliver.

## City Planning

I ran into this metaphor years ago while working with Paul Winter—probably one of the best IT architects I ever have known. Paul told me he found a city allegory useful when explaining the

nature of the three-tier architecture. In his city model, those three tiers were:

- Grids: everything above, on, and below ground (roads, sidewalks, bridges, water lines, storm and sanitary sewers, telephones, gas, and cable networks)
- Services (ambulance, fire, police, public works, public health, parks, etc.)
- Conventions (such as by-laws, traffic rules, civil law, culture-based norms, etc.)

I have found a similar urban representation quite useful in discussions about IT architecture, especially on the enterprise level. In my mental model, managing a complex IT environment very closely resembles managing a city.

As a city grows, new objects and buildings (systems) are built and need to be fit into the existing utility and transportation grids (hardware, software, and network infrastructure). Public services (enterprise services) must be planned to deliver more or new types of support. Norms and regulations (data standards, interfaces) must be observed and sometimes modified to improve service delivery and traffic flows (transactions) and eliminate conflicts.

Let's see what happens when we apply this mental model to a situation in which a developer wants to put up a new 50-story office building in the city. What would the city planning department need to do?

Generally, and regardless of where the city is located, the planners will focus on assuring two key conditions are met: Will the new thing fit? Does it meet the city codes? The first of the questions is the big one. For if the answer is no, an expensive investment in the infrastructure must likely be made first.

The second question is less rigorous and depends on local conditions. Cultural, social, and business norms are quite different in Columbus from those in Boston, Paris, Toronto, or Warsaw. They also differ in respect to technical standards and degree of complexity—usually related to a city's (company) size and topography (industry). All these factors mean that, although they follow a very similar process, no two cities (enterprise systems) look the same.

In this model, the role of city planners (IT architects) can be summarized as one of enforcers of standards and codes at the city

(enterprise) level. Conversely, IT architects should interfere only minimally with the internal design of individual structures beyond aspects that affect their interaction (interfaces) with the surroundings and some fundamental features such as safety (systems security).

Such an approach leads logically to the two principles all architectures must observe and enforce:

- Do not experiment with infrastructure; it must be solid and expandable, but there is no room for mistakes, as when they occur, they are very, very costly.
- Reward using shared infrastructure services; note no sane real-estate developers would build their own power generators and sewage plants, as it plainly is too expensive. Similarly, companies' internal economy rules must make shared solutions economically rewarding.

Keeping the city model in mind, let's now discuss how to rehire the architects, i.e., define their role in a framework that can produce value in the enterprise.

### Two Strategies for Rehiring

First a "basic" strategy. Focus on infrastructure services. Catalog them, make them known, available, and as inexpensive as possible. Remember not to fall into the vendor- or self-induced temptation to experiment. If you must, and you have the R&D budget, run experiments in a tightly controlled, limited setting where failures can be afforded.

Your basic drivers are to minimize the costs, duplication, and risk of failures. Please note you still have a significant degree of flexibility in defining where the scope of the "infrastructure services" ends. My advice: Do not abuse it. Start small and expand the scope as your capabilities and the process become proven.

The key underlying concept here is a "domain"—a set of applications and databases that are managed as units for business reasons (for an expanded description and discussion of the domain-based approach, see the article "Designing IT for Business" in the McKinsey Quarterly 2003, number 3). Typical examples include customer, product, or channel domains. The first effect of such an approach is it modularizes IT systems along boundaries that are logical to the busi-

ness. Consequently, domains are a foundation for creating business ownership and accountability for IT while relying on a business-driven collaboration with IT.

In this model, standard interactions within and between the domains are defined as "services," which become the primary responsibility of the IT architecture team. Done properly, the "services" are implemented only once and are available to all domains, thus reducing overall cost and complexity.

Experience from early adopters of the domain-based architecture approach suggests significant benefits in reduced IT costs, faster time to market, and encouragement of a higher degree of business accountability for IT.

### Final Thoughts

There are other conditions that will play a role in your decision. For one, a need for comprehensive architectural solutions and their affordability are dependent on company size. The new approaches (for in-stance, domain based) have a very different cost-benefits equation in large, global enterprises vs. midsize, regional carriers.

Second, a lot depends on your company's business philosophy. Those driven largely by the next quarter's bottom line are unlikely to embrace a comprehensive IT architecture program—no matter how promising.

And, since I spent much of this article's space on the city metaphor, a final thought and a warning from my friend Paul: "In general, metaphors can be extremely useful when they help the understanding or stimulate creative and imaginative thinking—a new and even exciting way to look at an old picture. Once the imagination is fired up and a new idea is hatched, the metaphor should be discarded. Too often it seduces less imaginative apparatchiks to think dogmatically."

---

*Marek Jakubik, a former CIO of Zurich Financial Services and Pitney Bowes, is a co-founder and CEO of the Insurance Technology Group ([www.insurancetg.com](http://www.insurancetg.com)). He can be reached at [marek.jakubik@insurancetg.com](mailto:marek.jakubik@insurancetg.com) or by phone at 416-214-3445.*

The CIO Chronicles is a recurring column published every two months in Technology Decisions magazine, a publication of the National Underwriter Company.