IEEE Software
www.computer.org/software

Enterprise Architects Join the Team

Rebecca J. Parsons

Vol. 22, No. 5
September/October 2005

IEEE COMPUTER SOCIETY

Editor: Martin Fowler ■ ThoughtWorks ■ fowler@acm.org

Enterprise Architects Join the Team

Rebecca J. Parsons

IT organizations are starting to recognize that having some level of enterprise standards can remove needless complexity from application portfolios and help enable the responsiveness that businesses demand. Myriad organizational structures can implement such standards, but most result in some form of a centralized group of enterprise architects. (We can argue whether this is the best approach another time.) I've spent much of the last five years acting as the interface between such enterprise architects and large development teams working to deliver complex applications. Based on this experience, I propose a new form of interaction to improve team effectiveness: enterprise architects should join the development team.

Architects and developers: A balancing act

The needs that enterprise architecture initiatives attempt to address are real. An IT organization can better respond to a business's demands if the applications in its portfolio

- are easy to integrate,
- implement similar processes in the same way, and
- have compatible perspectives on the enterprise's shared data.

Additionally, the organization's ability to support its application portfolio depends on the mix of technologies deployed and on the applications' durability and performance characteristics, both individually and collectively. Enterprise architecture organizations help development organizations address these issues.

Development organizations also operate under pressure from business groups to deliver more functionality, more quickly, and with higher quality. Agile development methodologies offer one way to align an application development effort's results with business needs. However, these methodologies often raise concerns due to the application design's presumed ad hoc nature—rather than start with a consistent vision of the application, the solution evolves as developers address the requirements. Central IT groups often view this as a random process that produces unreliable applications with unknown characteristics. This is clearly one—but by no means the only—possible outcome of this process.

Because we must balance the enterprise architect's goals with the needs of the agile development organization and the users driving the application's development, it helps to have enterprise architects join the development team. This provides opportunities to address the needs of all stakeholders: architects, developers, and business users.

Understanding trade-offs

As part of the development team, architects primarily act as customers, providing requirements that relate to the enterprise architecture. As in any agile development effort, these requirements must include acceptance criteria that explicitly define what it means to "complete" a requirement. The architect resolves any confusion about the requirement in cooperation with the development team in the same way that business users clarify requirements with developers and testers.

An enterprise architecture group's standards (requirements) are often difficult to explain or define. Even worse, some objectives are incompatible with each other. For example, optimizing for resilience results in more complex code; simplifying code often means less resilience. Extensive diagnostics can significantly impact performance. So, development teams often must compromise one requirement in support of another, and if the architect is part of the development team, he or she is aware of the challenges and can help make decisions.

Also, because the requirements have concrete acceptance criteria, the architects, along with the broader IT organization, can verify that the development effort conforms to the requirements. It's easy to verify that an application can run on a particular operating system. However, it's harder to determine whether the application implements the supportability standard, so seeing the results from acceptance tests can help.

The architects also help prioritize the agile team's stories, so they see the tradeoffs developers must make to balance enterprise architecture needs with specific business stories or with other architecture requirements. The architects can advocate the enterprise architecture requirements while seeing first-hand which requirements cause the most difficulty for development efforts. Understanding the trade-offs made helps them better anticipate a particular trade-off's impact. Under deadline pressure, business functionality usually takes precedence, so an IT organization often must support an application that fails to comply with the standards in some (often unspecified) way. The architect's participation in the team provides a mechanism by which the IT organization can make a case for their requirements during development—while there's still time to do something about it.

Justifying costs

Another challenge that enterprise architects face is justifying the ongoing cost of their initiatives to the business. Often all the business sees is the cost of implementing the enterprise standards—personnel costs and delays (development teams often ascribe delays to the constraints the enterprise group places on them). Explaining an initiative's benefits without a business-relevant vocabulary is difficult, so joining the development team helps architects understand the business's specific drivers and vocabulary. It also raises their profile with the business. Establishing relationships between business users and architects provides a foundation for the architecture team and helps justify the need for an enterprise architecture.

Building relationships

Establishing relationships between the architects and developers is another important benefit of this approach. In the traditional setup, an "us versus them" attitude often exists between the architecture team and the rest of the development organization. The objectives of these two groups often conflict. The development teams are primarily concerned with delivering a specific system in the most efficient way, so their perspective is necessarily local. The architecture team must have a much broader perspective, because it must balance a particular application's needs against those of the entire enterprise portfolio. What's best in the local setting is often not what's best globally. By working together, the architects and developers can begin to appreciate their respective objectives and openly deal with conflicts when they arise.

This level of cooperation requires the same kind of resource commitment as that expected from business users. The same problems of organizational inertia, skill set mismatch, and role realignment apply to the involvement of architects. Addressing this approach's social and organizational issues is essential. The role and skill requirements for architecture teams often differ from those of development teams. While more senior people often take architecture roles, those people aren't always as proficient in developing in the current language or technology. This model will place architects in uncomfortable positions if they're not proficient in the current technologies. However, we can deal with these issues to make the teams function smoothly.

The architects can advocate the enterprise architecture requirements while seeing first-hand which requirements cause the most difficulty for development efforts.

The parallels between architecture and business requirements are compelling. Agile methods have proven successful in delivering applications with improved value to business customers, and successful business applications keep the business users closely involved. We can achieve the same success by involving enterprise architects as customers for their requirements.

Rebecca J. Parsons is a technology principal and senior architect at ThoughtWorks. Contact her at rjparson@thoughtworks.com.