

By Daniel W. Rasmus



## The Tens—Collaboration Architecture Essentials

**1 Subordinate to Enterprise Architecture (EA)** The Collaboration Architecture is not a standalone architecture. Which EA approach you use will inform your choices about collaboration descriptions and syntax, but all models should include collaboration elements in its vision, documentation of collaborative interactions between individuals, applications and functions, collaboration standards and the technology architecture.

**2 Define Your Models** It isn't enough to draw a block diagram and call it an architecture. An architecture represents a response to the business, a blueprint for how work gets done. If you don't create models for how work gets done, then the technical architecture may have very little to do with the needs of the organization. Take the time to do it right and model work, then test to determine if your architecture can perform to your requirements.

**3 Understand the Standards** Just saying you are *standards-compliant* isn't enough. It is necessary that the organization understand the standards and their implications. Standards may limit you to certain features. E-mail has several standards associated with it, that create very different mail experiences (POP vs. IMAP for example). Standards are also the starting point for some vendor negotiations when discussing feature duplications and costs.

**4 Limit Your Tools** There are too many collaboration tools on the market. The explosion of social media fast on the heel of real-time communication through instant messaging and web conferencing creates a lot of great competitive products, and reduces costs — but those reduced costs can be expensive for an organization that over invests and under integrates their collaboration technology. The role of the architecture should be to rationalize and reduce the number of tools. If you do an architecture early, you can get ahead of the curve—do it late and you will spend time negotiating with functional departments over sacred cows rather than helping enable innovation.

**5 Choose the Deployment Model Wisely** Do you run your collaboration in-house, using an on-premises model, or do you outsource to a hosting company, or do you just let the cloud provide a service? Or do you create a hybrid? How you answer these questions will determine how much flexibility and access you have to underlying code, how (and through whom) you meet service level agreements and how much it costs to migrate when you change your mind.

**6 Focus on Services** If vendors offer what they promise, the architecture should focus on services and where those services are called. Regardless of what you actually implement, the architecture should focus on services as the set of capabilities that can be called by applications to provide a critical feature or function.

**7 Define What's In** Outsourcing contracts and other partnerships inject non-employee workers into projects and other work experiences. If those people choose to use their own tools to get work done, then they will fragment the collaboration culture. Part of any partnership agreement should be a clause that spells out work practices designed to keep employees and partners on the same page, and in the same workspace.

**8 Use it to Plan** The architecture is a blueprint. The blueprint should be updated to reflect the current state, but another version should exist that maps to organizational vision. By using architecture to plan, you can anticipate gaps in capabilities that lead to new partners evaluations, software design or active participation in a user group well ahead of when you might actually need the features or capabilities.

**9 Negotiate and Demand** As the owner (or leaser) of software, you have the right to negotiate with vendors. Collaboration is so popular that every major vendor offers some form of it (sometimes as part of the license, sometimes as an added-cost option). You don't need more than one of anything (unless you can't call the service you need.) The architecture helps define which services you deploy and where they could be called. Ask your application vendors to call the services you specify, not the ones they offer. It may take time, but if vendors want to define value through services-oriented architectures and standards in their marketing rhetoric, they need to be held to their marketing promises in reality.

**10 Challenge the Organization** IT needs to challenge the organization to use the architecture as a lens for rationalization. IT will eventually look like a hero for keeping costs down and customers in business, even if they do miss a few bells or whistles they personally wished for. A collaboration architecture should reflect the co-created vision of how the organization intends to work together, even if that vision is a work in progress.