



## Value Creation in the Age of Collaboration (Part 3) — David vs. Goliath

by Nilofer Merchant

Back in the day, value creation was very different from what it is today. Large organizations (think IBM, P&G, General Motors) were needed to build something of value. Lots of engineers, lots of quality-assurance (QA) people were involved. That weighty size and scale formed the traditional corporate hierarchy.

Do you remember when large-scale companies were considered mini-governments because of their global influence? So do I.

Unlike that hierarchy and power model, the Web enables some thing entirely different. The Web creates a model of talent and ideas and self-organization. People seek out others with similar interests. Collaboration on a scale previously impossible now occurs on an ongoing basis. Mass sharing ideas on a global basis is easy using BaseCamp, wikis, and other software that makes your physical location in the world immaterial.

There are two fundamental "value chain" shifts in the last few years. And perhaps it's so early that few have caught on to the full implications of it. But let me try to name it and identify what it means to companies today.

### 1. Manufacturing costs reduced

Open source and a worldwide talent pool have brought in efficiencies in software creation. You no longer needed a team of software developers and QA team to produce. Monolithic applications required a certain amount of engineering resources to code, QA resources to test, etc.

Now, with lots of little developers each tackling a piece of an overall solution, development—and especially innovation—can occur much more rapidly. No one has to live near you, work near you—heck, even know you—to join together to create something.

And, in the hardware domain, even injection molding is now readily available to the "common man"; concepts can be built quickly and contract manufacturers can deploy things with Chinese-based production. Being able to lower the cost of production and the access to tools lower the cost of entry into any market and create many alternatives.

### 2. Investment costs reduced

Production costs have changed. With the AJAX architecture and then the ability to "mash-up" technologies online through sites like programmable Web.com, the Web has changed the fundamental production model from "make it all" to "make some piece" of the highest value. This has meant that very small teams of 5-6 people can enter a new market and make money from their specific vertical application or knowledge.

This allows not only more creativity to enter the production system but also a different way to create a "product."

Layer on top of that the software as a service (SaaS) model, and any company can enter a market without needing to invest upfront dollars to buy applications, storage, etc. This has lowered the cost of entry to any market and created many alternatives or substitutes.

But what do those two disruptions mean?

It means that there is a new type of David vs. Goliath model in play.

### **Small beats large at innovation**

Small companies with greater flexibility and imagination are taking advantage of this change in economics far better than large ones. Big companies like Dell or Adobe are trying to figure out how to rationalize it with their existing systems and investment models, while smaller companies are simply creating.

This ability to work, communicate, and play with others is the underlying aspect of Internet community. Whether related to playing games (e.g., Second Life, World of Warcraft), work (e.g., Zimbra's ZCS Web Client, Jive Integrated Server), social networking and search (e.g., Six Apart's Vox, Google), connection to others is a must. 37signals is an example, and its Backpack is a digital to-do list focused on productivity and organization across a team.

Comparing small companies with large ones, it's interesting to see how smaller companies have the creativity to develop market spaces, while larger ones lurk until the market is well established, then enter and try to swallow up everything.

As you might expect, Microsoft has picked up on the trend, offering Office Live Collaboration Beta. The features page is corporate and lifeless, missing the flexibility and creativity required for collaboration. In fact, only one of six paragraphs of copy dedicated to the product is directly related to sharing, interactivity, and interplay—the other five, in typical Microsoft corporate/tech speak, tout rather generic items such as online business apps, data access, storage space, data protection, and customer support.

Zimbra, a David to Microsoft's Goliath, provides extreme collaboration capabilities for many operating systems, including Ubuntu, VMware, rPath and Mandriva. Its test approach was different, too, involving users worldwide. Vendors like Zimbra know customers prefer

personalization and customization—and the benefits they provide empower users instead of usurping their power.

### **But sometimes big works too**

Google, with unparalleled financial strength, continues to astound with a myriad audacious offerings. It has the luxury of entering sectors with no clear monetization strategy—cause being a disproportionate market share followed by the desired effect: money. The company's ability to couple search with the Web, images, video, maps, and the ubiquitous Google "more" is amazing. Gmail's integration with Skype provides a prompt when an email is opened with a phone number in it, asking whether you want to make a call. And you receive added ability to IM if your friend is online. Google is desirable for the functionality and intelligent integration of offerings.

The David in this pairing is Six Apart, offering a portfolio of blog products pointed at specific markets. Since its founding in 2001, community and sharing have been at the heart of the company. Mena Trott speaks about how intense the Six Apart community is—even to the point of providing her with unsolicited counsel about her private life when they didn't understand her tongue-in-cheek blog entries. With 7 million users, two rounds of funding, and several acquisitions, Six Apart continues to embrace individual expression as it revs up its technical offerings.

Oracle has been involved in some ugly battles in Silicon Valley, and you have to wonder whether its victories have often been Pyrrhic. It offers the 10g Collaboration Suite, but, like Microsoft, Oracle seems not to get the essence. Collaboration is about creativity and innovation—Oracle's description and stiff site ("enables organizations to increase user productivity and service levels while reducing...") imply the exact opposite of sharing. Oracle's dedication to its proprietary system costs users flexibility and innovation.

### **Will nimble win the day?**

Where Goliath Oracle stumbles, David Jive Software is nimble, open, and flexible. The opposite of "big iron," its Integrated Server product is elegant, intuitive—all the things you'd want to use to encourage customers and partners to interact with you and enjoy the process. There is a fundamental difference in approach when software isn't overdeveloped. Not having a large legacy architecture benefits Jive tremendously.

With Ruby, AJAX, J2EE, and other lightweight technology, this is a difficult time for Goliaths that must drag huge legacy architectures into each upgrade, each new product line. At a certain point, the amount of energy required will lead to diminishing returns.

I don't see David replacing Goliath tomorrow, but I do see the potential for thousands of Davids to create new markets, innovative services, and products that respond to the needs of customers. If you're a Goliath that's forcing customers to mold themselves to your company—consider yourself warned. If you want to figure out what to do, creating a parallel team or effort to understand this David model might be a good place to start.

## Value chains always change

The thing with value chains is that they always change. What creates power in one market over time becomes an Achilles heel, and some new societal, technological, or leadership change causes a shift. What is hard to know is how the value chain is changing right now, when you're in the midst of the change—rather than later, in hindsight.

I rarely, if ever, play the role of predictor or prognosticator. But I'm going to suggest that we are fundamentally in the middle of a big value chain shift. Value is no longer being created by those that play in individual sandboxes.

Rather, those creating value are the companies and leaders who share ideas and focus on building on each others' ideas so that creation is not about building the coolest widget but building different pieces together to solve a particular need.

## Are you ready?

How well you enable integration with other technology solutions as well as how open you can be to build collaboratively is the question. Measures would be the level and support around developer engagement. Do you have open APIs, do you enable platforms such as Adobe's Apollo, eBay's e-commerce solution... and do you release content into open source or provide component enablement so that others can co-create with you?

Implied in this model is that you can develop faster and innovate faster. But it would also mean that you are creating something based on a particular user set's needs. And you could layer other work on top of (or sit on top of) the platforms to make the solution complete.

I am sure we need to be prepared to move to this model. Rather than thinking about a particular end point, think about this as what you can do to grow in value by adopting the latest business models and directions the industry is offering.

So for all the Goliaths with successful businesses today, here are some ideas of what to consider—a checklist of next steps:

- **What standards can you set?** Success in the Web apps world isn't just about setting an application standard that end-users adopt, it's about setting an API standard that other Web applications embrace. For example, Google Maps is a nice service but is also being widely adopted by other Web applications that need to display maps. By sharing this way, Google cuts off competitors and gets free advertising.

What core functions should you try to own on every other Web site? Build those functions, make sure your brand is visible within them, and expose APIs so other companies can use them.

- **How can you take advantage of modularization?** More and more Web apps are being built as small code modules that link to others over the Internet. Plan for how you'll

disaggregate and rebuild your applications' core functionality in this world. Which modules do you need to own, and which ones can you license, or just link to, from others? Can you create a community of developers who extend your application by offering online modules?

The goal is to leverage as much as possible from others. Ironically, that means it's unwise to put too many engineers on your Web app development—they'll just revert to doing everything themselves. Charter a small team with replicating your core product by working with allies online. You may be surprised by how much they can accomplish.

- **Who are your new competitors?** Because Web apps break the barriers between companies, you have a new set of natural competitors you haven't thought of before. Anyone who's trying to build a core community for your users is a potential competitor and needs to be watched.

For example, if you make image editing software, you're now in competition with the photofinishing companies. Like you, they want to solve the problems of creative professionals, and unlike you they have a photo printing business that can subsidize the rest of their online activity. They would be happy to give away a competitor to your software in order to sell more prints.

What do you do? Buy one of them? Create a Web app module that they can give away (enabling you to build brand presence and sell people up to your paid services)? Ally with one of them to beat the others? You need to answer these questions now, before there's a competitive crisis (by which time it may be too late to act).

- **What revenue experiments can you run?** Established software companies often balk at creating Web applications because they're not sure how they'll make money in the Web apps world. Real Web apps companies don't let that stop them. They know that the industry is still in its infancy and you can't figure out the revenue model until you dig in and try things.

So get started, and run a series of experiments on where you can make money. You may find that it'll be a mix of advertising, professional services, support fees, charging for premium applications and services, and commissions for people who sell products and services through you. As long as you keep your expenses low (like the Web app companies do), you'll be able to afford these experiments. Then pour on the resources when you find the right mix.

- **Run the businesses side by side.** That one isn't actually a question, but it's our strong advice. Now that you're actively developing the Web 2.0 version of your company, defend your existing applications franchise for as long as you can. That means focusing on the functionality that's hard to implement using Web app tools, and keeping your core customers buying upgrades for as long as possible. Your goal is to make sure that as your users get cannibalized by the Web apps world, they get cannibalized by you rather than someone else.

But you're not trying to dramatically increase the rate of cannibalization. In practical terms, this means you shouldn't artificially constrain either your Web apps team or your traditional apps team. But you should assign the traditional apps with the goal of generating profit, while the Web apps folks pursue the goals of user growth, API adoption, and alliance building.

July, 2007