

SOFTWARE QUALITY CONNECTION

What Do You Mean, “Agile”?

Agile software development practices are all the rage, but many teams are not as agile as they claim to be. Many companies adopt a “hybrid” approach. Here’s how those work.

by [Lisa Morgan](#) | April 25, 2011

Being “agile” is a modern badge of honor. If you’re agile, you’re cool: progressive, team-oriented, and customer focused. You produce better quality way faster than the turtles mired in 500-page requirements documents. You rule because you’re agile. But how agile are you, *really*?

In discussion forums, agile developers bash waterfall, saying it’s archaic, siloed, and slow at a time when modern, collaborative, and fast can make or break a company. Yet it turns out that not all companies want to be, should be, or can be agile.

Many organizations claim to be agile, but an overwhelming number of software teams are not purely agile; they’re some sort of hybrid. As of Q3 2009, Forrester Research estimated that 35% or more of development teams were “[in some stage of agile adoption](#).” [A more recent survey](#) (PDF) by requirements management software provider Jama Software indicated that of 808 respondents, 19% are purely agile, 40% are hybrid, and 27% are waterfall.

Kapil Nagpal, director of customer solutions at outsourcing firm [Nagarro](#) concurs, based on his firm’s mission-critical work for global customers. He estimates 30% – 35% are purely agile, 20% are waterfall, and the rest use some sort of hybrid approach.

Nagarro’s Nagpal says that large customers such as Luftansa and General Electric (GE) cling to waterfall practices because they prefer to define requirements up front and get buy-in. Companies that are highly regulated also adhere to many formalities associated with waterfall so their organizations can withstand audits.

“[Customers doing waterfall] want to know how much an initiative will cost, which you can’t do in agile,” says Nagpal. “In some cases, companies might do waterfall but during the implementation they may switch to some sort of agile. If you understand 80%-90% of the requirements it is possible to break large development projects into more iterative pieces.”

Requirements is the operative word. [Well-defined and meticulously memorialized requirements](#) are essential to waterfall practices. Requirements are also the main reason why hybrid organizations are not purely agile. They want something more than a use case or a user story and something less than requirements-by-the-pound.

“The difference between the [hybrids] and the Agile purists is that the majority of companies still need to maintain the structure and control inherent with [requirements management best practices](#),” says John Simpson, VP of Marketing at Jama Software.

According to Forrester Research Senior Analyst Tom Grant, [too much documentation can cripple a team’s velocity](#) but the lack of documentation can damage a team’s success.

Customers Notice a Difference

Last year, I interviewed 14 vendors and 27 of their customers for an industry report that focused on [Pay-Per-Click \(PPC\) advertising practices](#). Interestingly, although I didn’t ask them about it, several customers I interviewed made a distinction between PPC platform providers who cling to annual or semi-annual major release cycles and those who frequently update their offerings. These marketers have no idea what “agile development” means but they are capable of describing its effects.

The perceived difference played out in three ways: release velocity, vendor commitment to customers, and customer loyalty.

Release velocity can make or break a sale in some organizations because of the perception that software vendors with faster release cycles are “more in tune with” market dynamics. And the customers have a point, since PPC tactics and the search engine algorithms upon which they are based change so often.

Morphing customer suggestions into product features was also a big hit. In fact it was like listening to a string of Windows 7 commercials in which the actors crow, “I invented XYZ.” Customers were delighted that their ideas had been translated into product features – not three years down the road, mind you, but within a few months. The result was greater customer loyalty, to the point of evangelism.

(It is fair to point out that that the agile or hybrid PPC platform providers with comparatively fast release cycles also have a Software-as-a-Service offering which enables them to roll out new features quicker and easier than their counterparts who are still selling packaged software.)

The Dirty Little Secret

Companies known for their nimble execution may not actually be purely agile; they may be a hybrid. For example, PPC platform vendor [Acquisio](#) was once purely agile but it has become less agile over time, according to Richard Couture, its co-founder.

“If you’re purely agile you’re applying agile software development by the book,” he says. “We started out trying to be agile but it always [took longer], was more complicated, and more costly than we anticipated.”

During development, projects are frozen for a week for testing, which doesn’t mesh well with the two-week sprint cycles.

“When we were on a bi-monthly release cycle, QA was tough,” Couture says. “The longer release cycle is also easier for people developing sales, marketing, and support documentation.”

When Acquisio was doing two-week sprints, modules were being developed by different teams. That resulted in coding redundancies and an inconsistent UI, which is definitely not a good thing if your target market is ad agencies. By slowing things down, the company has been able to manage development more strategically.

[Marin Software](#), another PPC platform provider, had a similar experience. When the company was small, its iterations were shorter, meaning one- to four-week sprints were common. As the company grew, it became harder to manage features, so the company switched to a three-month release cycle. However, it has gone back to two to four week cycles, which VP of engineering Joe Chang does not call sprints.

“A lot of companies can’t survive without being agile because it’s a new form of delivering value,” Chang says. “Our product specs are always rather lengthy. We have annotated designs, features, and business requirements, knowing we’ll be iterating and fine-tuning.”

Instead of rushing to market with incomplete features, Chang prefers releasing well-conceived features. When several customers request a feature, it is validated internally against Marin’s roadmap.

“Quality improves as the team builds smaller chunks because they get rapid feedback,” he says. “With smaller chunks, you can address problems more quickly and gauge how well a feature is working.”

Can Agile Projects Be Outsourced?

If you believe that agile projects cannot be outsourced because the team needs to physically meet daily, you’re not alone. Marin has an offshoring team in China, but Chang carefully chooses what he delegates.

“Teams have to talk, which is one of the traditional barriers,” he says. “Technology allows us to outsource and have an aggressive development schedule. It used to be you’d define a spec, outsource it, and then get the product back; but now developers have to be effective managers and understand the business. If you can have a conversation, it truly benefits your offshore or onshore development team.”

Nagarro’s Nagpal admits that outsourcing agile projects can be challenging. That’s why his firm uses collaborative tools and videoconferencing to accommodate its clients, including its largest customer which has daily standup meetings.

“Most of our clients are looking for an iterative model [in which] we demonstrate progress and get feedback,” he says. “If a customer wants a fixed price, then we need to impose a hybrid model because we have to understand their requirements.”

Team size, previous exposure to agile methodologies, project scope, and a customer's working style all influence how Nagarro works with its customers. Some customers insist on using a particular methodology but in the absence of that Nagarro recommends a hybrid that combines well-defined and well-documented requirements as well as prototypes and iterations to ensure everything is on track and on schedule.

Two problems with waterfall practices are changing requirements and scope creep, which cause rework. Within a typical nine-month to one year development cycle, three or four months might pass before Nagarro receives any feedback.

“Agile methods allow us to present our work in short increments,” said Nagpal. “[We prefer] a hybrid approach because true agile only works for companies that know how to do it well through collaboration and meetings.”

So, how agile are *you*? Are you purely agile or some sort of hybrid? What determines your approach? We'd like to know what you're doing now, how that differs from before, and any insights you may have on the subject.