



By [Michi Broman](#)

We are often asked by potential clients if we “do Agile.” Being part of an outside firm, fitting into a client’s agile process can be a curious and interesting challenge given the variety of ways we see agile methodologies applied.

We sometimes feel that clients want “Agile” because it’s new and modern. While many larger companies follow a Waterfall process, it seems like no one wants to admit “We do Waterfall,” as if those who do may be old and in the way. For those in each of the camps, it seems that ne’er the two should meet.

While a recent New York Times article, [Trouble in Start-Up Land](#), did not mention Agile Development directly, it did talk about cultural differences between the new and old guard. The new guard sees a “...platonic vision of how work should be: a tight-knit group of friends, pushing themselves to greatness.” This falls solidly in the Agile camp, with small teams that self-regulate, with no hierarchy.

For me, the greatest thing about the Agile process is that the sprints require real team collaboration. Collaboration is part of the definition of scrum, and success depends on it. It’s a fun and exciting way of building something that quickly yields amazing results: very New School.

The downside is that without a good framework, agreed-upon principles, and rigor, the Agile process can break down. When requirements are not clear and the technology is complicated, work slows down, deliveries slip, and oftentimes, more people are added to try to stay on schedule. This can make things worse. I have been in situations where a flimsy architecture meant that it took longer to build a product than if we had designed the product fully before handing it off to development. (Writing this makes me feel like an Old School Lady dropping a wet blanket on the awesome New Way of work.)

An Agile-method project that is currently working well for Blink is where we have embedded a designer into our client's team. He is a little ahead of the developers, completing the interaction design within each sprint, and attending daily scrums that include design reviews. We have another project where we have embedded a researcher into a team, and while this is not Agile development, the principles of collaboration and a tight-knit team are providing an environment where great work is flourishing.

Embedded doesn't always work though, especially if we are not in your company. Because of this (and because most companies have not adopted a true Agile method), more often than not, we hand off very detailed annotations and specifications to a client's development team. The advantage to this is that we've thought through the entire product and how some features may affect others from a user experience perspective. This can provide great designs for very complex products, and the infrastructure to support complex design can be built in a solid and methodical way.

One place where we've managed a kind of hybrid approach is with one of our health-care industry clients. The client has a very established back-end system and is a leader in the industry. However, in the past year the client has lost business to companies with slick front-end apps. If you look closely at the competition though, the [user experiences](#) are very shallow and break down pretty quickly. However, in a sales call, that wasn't apparent, and our client must modernize to stay relevant.

Designing their new front-end applications on multiple devices has been a joy. We're prioritizing all our epics and stories now and will soon start designing. And we're collaborating closely with the client's dev team before they begin building. Their architecture is really solid (Old School solid), and I'm confident that we can rapidly roll out the product (New School rapid), with the entire system being completed this year. This has been one of the most interesting and challenging projects I've worked on, and I'm confident that our client will blow its competition out of the water.

I'm hoping that a "right-approach-for-the-project" can become the norm, and that the acrimony between Old School/New School, Agile/Waterfall can stop. Solid architecture is not a bad thing. Rapid iteration is not a bad thing. Collaborating on bringing great products to market is always the best way to work.