

Agility is Not a Methodology.



Don MacVittie, 2013-23-04

Lori and I were discussing the entire topic of agility the other day, she looking at it from the SDN prospective, and I from the evolutionary perspective for development. While I know there has been a metric ton of writing about the topic, since I've used agile development, and find it to be useful, but didn't become a fanatic on my first agile project, I thought some perspective might be in order.

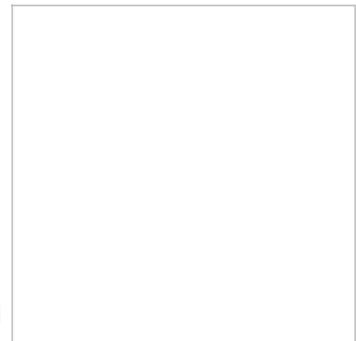
There is a reason why absolutists always fail in technology. Not just because high-tech is always changing – which makes absolutism fleeting at best - but because the part that determines whether computer scientists get to be absolutists on topics as varied as “object purism”, “No [fill in mobile device since first Palm] on my network”, or “We're an X shop” is not at all scientific. Nor is making absolute statements, unless you're discussing a proven fact.

Money and by extension people's productivity determine what kind of shop you are, what you use or don't use, what standards are set in stone and what are not. And sometimes the market. I used to work with a network manager that told me “We're a Nortel shop, we'll never use anything else.” Uhhmmm... Wonder how that's working out for him.

And over the last several years, agile/lean development has that air of “My OS is better than yours!” to it. In these back-and-forth discussions, I have tried to maintain perspective, no matter what topic is polarizing my fellow geeks this year. What you *like* is irrelevant. Use the right tool for the job. Sometimes, that's not your favorite, because frankly, there are no silver bullets.

Don't get me wrong, I firmly believe the evolution of agile is what finally broke the “Mythical Man Month” mentality that had ruled over and held back IT as badly as Hume has decimated philosophy with regards to actual science. I hate that they let Hume hold them up while science continues to forge new ground and face issues that philosophers could help with, and I hated that IT shops believed there was no faster way to get code out the door once the team was bigger than one person.

Agile improved developer productivity through a variety of methods, but I feel the biggest was focusing on one thing at a time and reducing non-coding activities. All in all it is a great improvement.



But our field of endeavor tends toward zealotry, and the agile pundits out there sound alarmingly like the cloud pundits or any of the other absolutists we've burned through in the past 20 years or so “all other things are obsolete, we have our flavor of agile now!”. I'll agree that agile methods are right for a huge percentage of IT projects, I just won't agree they are for all IT projects.

The thing is, there are cases – more than one might think – where agile isn't the best solution. The larger the code base, the larger the team, the more distributed the team, the less and less it a project is suited to agile methods. Oh, like so many methodologies, you can force-fit it, but I'm talking about *best* solution, not *possible* solution. And the more documentation required, well, agile teams don't tend to excel in that department, viewing a document as a waste when you could be writing code. And that doesn't even touch on the array of things like “don't generalize for re-use, you can't predict the future” slamming into “We are building a corporate repository...”

But the thing is, like all such methodologies, it does introduce some great tools to be put to use even on projects that don't lend themselves to its overall methodology. And some that hold it back even when the methodology is the right one. Sponsors vary in actual support from “assigned to me, whatever” to “we really need this, what can I do to help get it right?” And that is just the nature of any methodology that requires sponsors for each project. Just as drafted militaries tend to be lower quality than volunteer ones, the sponsor process includes both good and bad.

From the good bits, breaking the problem down as small as possible is a great way to achieve focus, just beware of blinders. The future *is* coming, and sometimes you *can* reasonably predict it. Others may need to re-use some of the source – assuming the team didn't write it so specific as to be useless, and where documentation is concerned, the team needs to focus on new developers a couple years from now, when the SMEs won't necessarily be available, not the minimum to communicate today.

But the most agile thing that comes out of agile? Business management is now aware that development can be faster than (maximum estimate X 2, measured in the next highest unit of measure). Agile projects are sometimes late, just as any other, but they are generally tracked better by the nature of labor division, so expectations are managed better. And upper management can now drive the mentality that will make IT more responsive.

And in the end, that's the point. If senior management says "agility is our goal", and is willing to put the money and effort behind it, the fact that some projects are better suited to other, more overarching methodologies doesn't matter, but saying "We need tool X and it will make us more agile, here's how" will get budget, improving overall IT responsiveness.

Most of us don't get to choose the methodology (or lack thereof) in use for our team/group/department, but we do get to point out when the One True Methodology mentality is a weight and not helping. Do so. Strive, as most of us do, to do what is right for the organization, not what is the tool of today.

And embrace the bits that help no matter what. Focus on small problems, don't over-document (but don't under-document either, there will be others behind you that need to get up to speed), and unit/functional test at that one-business-problem level.

In the end, I titled this blog "Agile is not a development methodology", simply because it is a mindset. A mindset that starts at the top, and prioritizes getting the business what they need in the fastest manner possible while still meeting requirements. You know, what we should have been (and many of us were) doing long before agile came along.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com