Which of the 'Real World' does your Software Team live in ?

Agile Practices favor collocation of teams because it improves efficiency and effectiveness of communication, and team productivity. However, there is an increasing trend at the business level to distribute work to remote teams, taking advantage of operating in multiple time zones, talent availability, cost arbitrage and other business factors. Still, most Agilists would prefer collocated teams, even at the cost of ignoring economics of the big picture. I feel many of us are living in denial mode as far as remote work is concerned. Just because Agile Principles 'favor' collocation, this is no good reason to collocate the teams, come what may. One shouldn't sub-optimize the whole in order to super-optimize the parts.

If you are lucky to have an 'ideal real world': small, collocated teams, than by all means, go Agile in its full letter and spirit. However, if higher business imperatives require your teams to be in multiple locations, then you can't be pushing your tool just because that's your comfort zone! A far better approach would be to first understand organization's constraints and expectations in distributing the work to remote teams. What benefits are being sought, and are actually being accrued? Once done, that is a great starting point to make distributed teams more productive - so, it is no more a question of pure faith vs. pagan. It is a question of acknowledging the ground realities, and finding out the best way to help teams in the given context.

We all know how difficult multisite development is. Yes, that is the real world. Could we go back to our management or customer and tell them that our software development process doesn't handle multisite work very well, so how about putting us all in a single location because that's what we know best? Theoretically, one might do this, but I think the company has hired me not so that I could replay obvious excuses to them, but show my knowledge, skills and ability to manage complex and real-world problems like these. Precisely the reason I am in this job - if those real world problems were not there, would there be a justification for my role? So, I would perhaps be better off doing as best a job as I can without putting pre-conditions to performance. A good start would be to acknowledge the real world instead of rejecting it just because it doesn't match the book definition.

That brings me to a very fundamental thought - What *is* the real world? Is there a universal definition of real world, or is *your* real world different from *mine*? Is there a universal process that can heal all problems in the real world (whether there is only one real world or multiple real worlds)? Why should we always think that a process *must* be 'macho' enough to be able to solve all type of real world problems? How can a process dictate how the 'real world' should look like? I feel a dogmatic implementation of Agile could actually jeopardize projects. The pragmatic approach is to establish your 'real world' unapologetically and find what works best for you.

So, which of the 'real worlds' does your software team live in - an 'ideal real world' or a 'real real world' ?