Why Agile May Be Fragile

By Phanish Puranam , Roland Berger Chaired Professor of Strategy and Organisation Design at INSEAD, and Julien Clément , Assistant Professor of Organisational Behaviour, Stanford Graduate School of Business

The hype around Agile organisations needs some debunking, so that Agile's actual value can be salvaged.

Every few years or so, the management consulting community latches onto a new cure for everything that ails large (i.e. potential customer) organisations. The hype cycle unfolds in predictable phases: the new practice seems to work in a number of organisations and gets highly publicised, companies of all sizes and kinds are encouraged to adopt it, a few prominent failures occur, and infatuation gives way to disillusionment as the magic medicine doesn't seem so magic anymore. The cycle unfolds again (and again) as the fallen wonder gets replaced by the "next big thing". In the process, a perfectly sensible practice that could work well for some organisations to solve some problems may become unfashionable enough to be rejected by most organisations. The baby is gone with the bathwater.

We fear that a similar fate awaits the latest panacea – the set of organisational practices that have come to be known as "Agile". At this point, every top management consulting company in the world has some version of a "see how Agile you are" diagnostic survey, as well as a "here is

how we can help you become Agile" offering advertised on its website. The first premise, often stated explicitly, is that every company, whether an FMCG giant or a national bank, needs to become Agile. Fair enough – who wants to be "clumsy"?

The second, not so explicit, is that the particular set of organisational practices being advertised, largely revolving around the use of cross-functional teams with self-organising elements, is *the* path to get there. The evidence for the second premise is simply non-existent at this point. In fact, everything we know from the study of organisation design suggests it is quite unlikely to be true.

To be clear, there is no disputing the appeal of organisations that are both efficient and innovative, which seems to be the promise of "Agile". This is a problem that organisational scientists have been studying for a very long time under a variety of names ("**ambidexterity**", "**exploration vs. exploitation**", "**organic vs. mechanistic structures**" to name but a few). We think we understand quite well why it is hard to accomplish both, as well as the nuances around the kinds of solutions that are possible under specific conditions.

We'll go further: We are fans of Agile practices as a solution to a distinct class of problems in certain contexts, such as those involving a search, be it for new designs, product features or even management policies. Agile practices use rapid, parallel and inexpensive iterations to solve search problems. If there are parts of your organisation that can benefit from this kind of problem solving, then Agile design may be for you. Even then, not all search problems may benefit from Agile practices. The relevance of Agile is murkier still if the context is one of execution, e.g. when the firm seeks to attain high levels of reliability with well understood processes.

What we want to caution against is the indiscriminate application of Agile beyond search situations, in large part because the inevitable disillusionment may then discourage even sensible adoption. Let's keep that poor baby from disappearing with the bathwater.

Where Agile comes from

Agile practices grew out of a product development framework widely adopted by the software industry. Also called "scrum", its central principle is to set up highly autonomous and independent teams, each working on a

different aspect of a larger, complex problem. Instead of marching in bureaucratic formation, they are given freedom to "sprint", i.e. rapidly advance in parallel towards a solution with minimal top-down oversight. Interdependencies between the various teams are managed through frequent meetings and any rework is accepted as part of the iterative process. Upon completion of the immediate task, teams either dissolve or take on a new project. When it works well, it yields a heady mix of autonomy, immediate and measurable accomplishment, as well as a sense of team cohesion.

The logic behind extending Agile to entire organisations is that they increasingly face conditions very similar to those of software organisations: the pressure to excel at rapid search and to be egalitarian, in a bid to keep knowledge workers motivated. Therefore, the argument goes, we should do what they (seem to) do successfully, which is Agile.

As we see it, there are three problems with this argument. First, the importance of good old-fashioned execution relative to search varies a lot across sectors. Not all are as search-intensive as software development. Second, we are not sure these practices work as effectively outside software development, even for search problems. Third, even in the software industry, the evidence that Agile practices can be scaled effectively is still thin.

Speeding through rapid parallel development is a chimera unless the pieces can truly be worked on in parallel. Production technologies vary enormously in this regard (a property known as "decomposability"). All that parallel sprinting can end in a crash at the finish line unless the pieces fit together. There is no obvious reason to trust that the degree of decomposability in, say, life sciences R&D or new product design in FMCG will be the same as in software production. Software architects carefully plan for modular work by specifying interfaces and design rules that allow for this parallelism. They also have the good fortune of dealing with a technology that allows this to be reasonably transparent. With code, what you see is what it does. The rest of us must tread carefully.

It is possible that a day will come when all execution-related activity can be automated (either with robotics or AI), leaving humans free to focus on pure search. This in turn may involve heavy reliance on computer simulations rather than expensive real-world prototyping. On that day, Agile practices might be the norm for every part of every organisation. We just don't think

that day is tomorrow.

Where Agile practices may work well

So, does this mean Agile won't work for you? No, only that you should be careful. We suggest a "concentric circles" approach to applying Agile, with software development activities at the core. Cautious extension to other search-intensive domains that have many software attributes (low cost of prototyping, parallelism) may come next. Product design, ideation, technical problem solving are good candidates.

But you can do more than just guess whether Agile may work for you. One of the exercises we assign to our **Org 2.0** students is to design a prototyping experiment – with properly randomised treatment and control groups – to generate evidence that Agile practices can work in a particular organisational context. One low-cost way to run such an experiment is gamification. Think of an activity that could be run over a few hours or days (and potentially shares critical similarities with your employees' usual work environment), randomly divide your employees into "hierarchical" and "Agile" teams, and analyse which group did better and why. Our colleague **Maria Guadalupe** is soon launching a course on how to get Agile right, which should be very valuable too.

The appeal of autonomy

We speculate that the promise of autonomy and hierarchy-free organising (and resulting employee engagement) may also be an important driver of widespread interest in Agile. We have **written before about this** and our basic viewpoint has not changed: As of today, we still don't know of any robust, broadly applicable mechanism besides hierarchy for scaling collaboration among a diverse set of interdependent individuals. Yes, there may be more delegation than before, **better tools than the simple org chart to design hierarchies**, and more attempts to mute the observable differences in status across layers, but the basic multi-layered hierarchical framework is still the one that seems to work in the broadest set of circumstances.

A close examination of prominent exceptions (and there aren't so many) reveals that they operate under some fairly exceptional circumstances. The idea that we can take an existing set of employees and production technology and seamlessly reorganise them into a non-hierarchical structure

has, at this point, almost no evidence to support it. We fervently wish it weren't so, being egalitarians ourselves. And we should certainly keep trying to find a robust and scalable alternative to hierarchy, because a preference for autonomy and a dislike of inequality are both important and widely held values. But Agile as it stands today is unlikely to be that solution.

[Phanish Puranam](#) *is an INSEAD Professor of Strategy and the Roland Berger Chaired Professor of Strategy and Organisation Design.*

[Julien Clément](#) *is an Assistant Professor of Organisational Behaviour at the Stanford Graduate School of Business.*

Follow INSEAD Knowledge on [Twitter](#) and [Facebook](#).

Find article at

https://knowledge.insead.edu/leadership-organisations/why-agile-may-be-fragile

---

About the author(s)

Phanish Puranam is the Roland Berger Chaired Professor of Strategy and Organisation Design at INSEAD.

Julien Clément Julien Clément is an Assistant Professor of Organisational Behaviour at the Stanford Graduate School of Business.