
If Agile Isn't Working, Don't Blame Your Team (Right Away)



By [Luk Van Wassenhove](#) , INSEAD

Project managers need to keep their eyes wide open when it comes to Agile norms.

Agile's short iterations and steady deadlines can instill a sense of urgency, boosting team productivity. But while schedule pressure creates discipline, it is important to find the sweet spot between *laissez-faire* and burnout. Iterations that are shorter than necessary end up hurting project management performance. The best Agile software development projects weigh the trade-offs by gathering data instead of falling for illusions of control.

In our experience, firms tend to adopt Agile's normative recommendations wholesale and fail to consider the impact these may have on their teams. There is such a thing as excess agility: When initial review dates come too early, problems from the first iteration cascade into the next, quickly piling up. As the team tries to make up for delays, the constant overtime affects productivity by increasing exhaustion and turnover. Team leaders need to prevent a vicious cycle where excessive schedule pressure leads to higher error rates, backlogs, ill-advised shortcuts, poor quality assurance and lower

quality products.

Kim van Oorschot of BI Norwegian Business School, Kishore Sengupta of Cambridge Judge Business School and I have long collaborated on research regarding software project management. In our latest paper, “[Under Pressure: The Effects of Iteration Lengths on Agile Software Development Performance](#)”, we show that the widely advocated 20-working-day Agile iterative cycle might not optimise the quality, cost and duration of projects.

Balancing schedule pressure, team behaviour and performance

Building on an extensively validated model of software development, we combined the technical aspects of Agile with the human factors underlying it, such as learning, experience and turnover. We used real-life data, involving a team of five people scheduled to work 260 days on a project. We analysed the effects of various iteration lengths (i.e. schedule pressure) on team behaviour and project management performance. We found that a moderate iterative speed of 43 to 65 working days (two to three months) optimised project quality, cost and duration.

At this moderate speed, the number of undetected coding errors was on average 39 percent lower than at higher levels of agility (163 vs. 263 undetected errors). The project costs were 26 percent lower (1289 vs. 1737 person-days expended). A moderate iteration speed still led to some overtime, but not to the extent that it increased exhaustion and turnover. The team’s sustainable productivity and learning reduced the need for overwork, which itself curtailed any inclination to take shortcuts that feed the errors and rework loop.

Of course, different types of projects will call for different iteration lengths, if only because the tolerance for errors will vary from one industry to the next. Sometimes the most important thing is not to be the first to market, but to get it right – think of military software applications. On the other hand, environments with highly volatile specifications may call for very short intervals of one month or even less. In most others, they may be unwise. So, what is a manager to do?

The full picture

Many software firms capture data such as the number of bugs and lines of code, and whether projects were completed on time and within cost. While this is good, it does not provide the full picture. Managers also need to gather data on the human factors of projects, such as staff fatigue and turnover, and model their overall impact on project management. For instance, it takes time, money and effort to hire and train new people. When staff leave, the firm loses a treasure trove of knowledge.

Just like most people, organisations often look for simple solutions. They do not necessarily want to do lengthy assessments. When a project ends, they hurriedly move on to the next. However, firms would do well to experiment and try to learn more. We estimate that only 20 percent of companies bother to do so. Instead of relying on evidence, too many firms prefer to make decisions using a cookie-cutter approach that merely gives them an illusion of control.

Frankly speaking, if the widespread one-month Agile cycles do not work for your firm, do not assume your workers are idiots. By measuring all variables – including the oft-forgotten human factors – you can best figure out what makes sense in your environment and thus keep both your employees and your customers happy. Firms that do that can gain an unparalleled edge.

Find article at

<https://knowledge.insead.edu/operations/if-agile-isnt-working-dont-blame-your-team-right-away>

About the author(s)

Luk Van Wassenhove is an Emeritus Professor of Technology and Operations Management and the Henry Ford Chaired Professor of Manufacturing, Emeritus at INSEAD. He leads the **INSEAD Humanitarian Research Group** as the academic director.