



 Print Article  Close Window

From: [www.cio.com](http://www.cio.com)

## Time and Distance Enemies of Agile Project Management

– David Taber, CIO

July 11, 2013

Although I'm a card-carrying agile bigot, even I have to admit that agility has its frailties. I even wrote a snarky checklist of the [best ways to make an agile project blow up](#). But that checklist was a bunch of symptoms to avoid, rather than something you can specifically measure and set project thresholds for.

Let's examine the two constitutional enemies of agile project management, then, and see how they *can* be measured.

### First Agile Project Management Enemy: Distance

The closeness of collaboration between developers and users makes Agile projects so cost-efficient and responsive. Ideally, there's continuity of relationships, with daily updates for all team members. After all, daily "stand up" meetings aren't for motivational or other pointy-headed-boss purposes.

LEFT ALIGN PIC



As a consequence, physical proximity helps. We recommend that the core team be within shouting distance of each other. If users discover a subtle new requirement, every hour the developers work before they understand the ramifications may be an hour wasted. If the developers identify an alternative strategy for satisfying a user requirement, it really matters what day the users validate (or reject) that alternative.

With physical distance comes increased opportunity for misunderstanding or delayed communications. Even if team members are just on a different floor of the building, you need more checkpoints and redundant communications to keep everyone in sync.

**Analysis: [Yahoo's Telecommuting Problem Is Management, Not Collaboration](#)  
More: [Citrix CEO Questions Efficacy of Yahoo's Remote Working Ban](#)**

Once the team's no longer in one building, the problem doesn't get much worse until some team members are in different time zones. The staggered work schedule makes tight communication that much harder.

There's another axis of distance, though: Distance on the org chart. An Agile team thrives by bridging the gaps and making direct connections between developers and users. Part of the Agile magic—minimizing the bureaucratic buffers that insulate effective collaboration—becomes more vulnerable as team members are farther and farther flung.

Things take an exponential jump when the team must span national borders. It's not just a matter of language. It's management culture and the ability to communicate subtle, perhaps painful points without miscommunication. This is particularly troublesome when one culture is very direct but another is highly nuanced and embarrassed to communicate any negative information.

Of course, technical and management tricks can mitigate each of these issues. It's amazing how effective even free services such as Google Drive and Google Hangouts can improve remote collaboration. But every one of those techniques requires some discipline and involves some cost to the project. When you have to take all those costs into account, then the flexibility and speed of Agile can be called into question.

**Case Study: [Library Competes Thanks to Agile Development Outsourcing](#)**

Our consultancy has found that the expense of actually being on site turns out to be a better deal for the client while, at the same time, less painful than trying to [manage offsite resources](#).

Other major agile shops take a different approach: Having on site all "soft touch" personnel, such as business analysts, architects and executive support, with all the worker bees overseas. Each deliverable is specified on site at the client's side, and all work products come together in overnight iterations.

This extension of the agile model works well for larger projects, as the required redundancy and overhead become negligible when 80 percent of the hours are in low-cost areas. However, I've yet to see a case where this can be reliably scaled down to a small firm, let alone a small project.

The best metric, then, is percent of team members that are not in the same building. Anything below 20 percent is negligible, anything between 20 and 49 percent requires special attention, and anything over 50 percent requires specific processes, tools and redundancies to keep the projects working.

## **Second Agile Project Management Enemy: Time**

It's ironic that agile's speed and responsiveness is at the root of a vulnerability. The whole point of agile is having quick iterations that respond to the current priority list. However, if you delay an agile project just a few weeks, you might need to redefine it.

Why? In any important business process, things evolve—sometimes quite quickly. If an agile project was conceived, then put on ice while the business process and the organizational chart evolved, then the stories (requirements) and defining requirements (priorities) for a sprint may be incorrectly defined or even irrelevant.

In contrast to waterfall projects, agile teams don't work with all-encompassing requirements and architecture tomes. By avoiding all that overhead, agile requirements are sparse and budgets are smaller. But they have an expiration date. To use a bad analogy, agile is fresh vegetables that are better

for you—but they don't have the shelf-life of the canned stuff.

This effect is particularly severe when budgets remain immovable. What you thought it would cost before all the interruptions, team member changes and modifications to thresholds, criteria and picklist values is not what it's going to cost at the end—even if none of the requirements changed.

### **Commentary: [Why Agile Isn't Working: Bringing Common Sense to Agile Principles](#)**

Delays and long interruptions of an agile project signal deeper issues: The project isn't that important, doesn't have the right team members, serves as a [political football](#), has goals that are a moving target or has simply been ill-conceived. Look for those things as a root cause of an agile project that can't get going in earnest.

There's a nasty consequence of delays and interruptions. They eat away at trust and credibility—for everyone. Companies have a pretty short memory, even if individuals have lasting impressions. Since trust is the critical ingredient of collaboration, and tight collaboration is the foundation of agile, you can see how delays are destructive.

Of course, delay causes completion date slip, but it also causes scope creep. Even if nothing else changes, delay multiplies learning curve and configuration management costs. Start and stop an agile project often enough, and overruns will be inevitable.

The best metric is expressing cumulative delay interruption as a percentage of the nominal agile sprint cycle. Anything below 50 percent is negligible, particularly if you have two-week cycles, anything between 50 and 150 percent requires special attention, and anything over 150 percent indicates that the project is already in trouble.

### **Optimal vs. Suboptimal Agile: The Proof's in the Pudding**

Our firm has a clear example of an agile project that was done on site with no interruptions, versus one that's offsite with typical big-company interruptions. The team was the same, the project scope was directly comparable—and the results are clear.

The first sprint involved a trip to Europe for an on-site intensive. The second offsite sprint, which actually was a smaller deliverable, cost at least 50 percent more (even including the travel savings) and took more than a month longer on the calendar. User satisfaction was equivalent.

Optimizing an agile project means minimizing time and space. If distance and delay are realities for your team, make sure to apply every countermeasure you can think of and lower expectations of the users and budget holders.

*David Taber is the author of the Prentice Hall book, "[Salesforce.com Secrets of Success](#)" and is the CEO of [SalesLogistix](#), a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel and India. Taber has more than 25 years of experience in high tech, including 10 years at the VP level or above.*

*Follow everything from CIO.com on Twitter [@CIOonline](#), [Facebook](#), [Google +](#) and [LinkedIn](#).*

© 2013 CXO Media Inc.