



 Print Article  Close Window

From: www.cio.com

Why 'Agile Project Management Controls' Isn't an Oxymoron

– David Taber, CIO

April 11, 2013

Everybody knows I'm a shameless [Agile](#) zealot. Let's face it, though. Read the Agile manifesto with a cynical eye and the message seems an awful lot like, "We're artists, and if you want a great outcome, you can't manage our art projects the old fashioned way."

Cynicism aside, it's true that agile projects can't be managed with conventional [waterfall](#) techniques. But that doesn't mean they can't be managed. They need to be budgeted for and controlled, or else they'll never get approved. In other words, they need adult supervision.

Agile Isn't a Marathon, It's Many, Many Sprints

The tricky part is that the adult supervision for agile projects requires more continuous effort and, yes, more subtle attention than the classic waterfall techniques. In a typical waterfall project, the executives approve a budget and a calendar at the outset; after that, it's just the occasional black-and-white review meeting. The project managers can simply "check the boxes" as each requirement in the spec passes its acceptance criteria.

Analysis: [Who Says Agile Development Can't Be Faster?](#)

While creating that spec and the acceptance criteria may take a lot of up-front work, once you've kicked off the project waterfall monitoring and controls are pretty easy. Except they're not. All too often, the check box items aren't precise or all that relevant.

In the course of the project, discoveries can invalidate some of the details of the spec—and nobody has time to go back and bring the spec and test criteria up to date. So the twin enemies of every software project--scope creep and vague acceptance criteria--rot away the credibility of the budget, the schedule and the controls of waterfall projects.

In contrast, agile projects require much less up-front work; the specification and controls are developed and enforced every day of the project. Agile projects have no detailed specs—sometimes not even when they're completed.

Instead, the team must be entrusted with prioritizing the tasks by business value, and coming up with the "defining stories" for each release. The team runs its sprint within a defined schedule and budget. Since the sprint is typically short, there's little issue of overruns, and scope creep is counteracted by team decisions that "throw things overboard" to make the sprint deployable.

Tips: [How to Deal With Software Development Schedule Pressure](#)

The only thing you know about a sprint, then, is exactly when will it be over, not exactly what will be delivered. While there are tools to help with the mechanics of cards, stories and units of work, the

fundamental control mechanism for agile is the team and its interactions. This feels—and probably is—qualitative. It must drive the numbers guys crazy.

Making Agile Manageable: 18 Key Questions

I like the adage, "You can't fix what you can't measure." I also like the metaphor that agile sprints are like a train system; the credibility of the system comes from the frequency and reliability of train arrivals. Users won't be all that upset if a feature misses the original train as long as they can set their watch by the next train's arrival.

So, how do you make agile measurable in a consistent and defensible way? Actually, describing how to get hard measurements isn't that hard. Here are several metrics you'll want to consider and the questions you should ask in order to get your answers.

Sprint variances from stated schedule:

- Did feature development stop on time?
- Was integration done every night?
- Did the final tests start on time?
- Did the sprint complete on schedule?

Sprint compliance with coding standards:

- What percent of new variables has data definitions?
- What percent of the classes or modules have unit tests?
- What percent of lines of code were covered by tests?
- How many test records did their tests use?
- How many outcomes, or variations on expected results, were evaluated by the tests?

How-to: [6 Ways the Cloud Enhances Agile Software Development](#)

Sprint variances for stated budget (or level of effort):

- Did the sprint consume unbudgeted resources such as staff developers, contractors or test resources?
- Did it not consume budgeted resources that could or should have been allocated elsewhere?

Sprint compliance:

- Did the sprint break any deployment or security rules?
- Did the release go through the required audits and reviews?
- Did the release generate any items that need to be remediated in future sprints?

Sprint business value:

- How many "value points" were supposed to have been delivered by the sprint, and how many actually were? (Note that value points are purposely subjective. That way, the business unit can prioritize functionality by whatever has the most impact to the way they run their operation.)
- How much of the sprint effort was infrastructure and refactoring to keep the code flexible to accommodate business needs in an upcoming sprint?

Sprint repeatability:

- Can the team continue to do sprints in this way without team burnout or other side effects of "heroic effort"?
- Has the team created a *brief* postmortem document and lessons learned on personnel, organizational and technical issues?

Soft Measurement: Get Users (Including Executives) Involved

One key success factor in agile is user engagement throughout the sprint. The only way you get real quality, usability and user adoption is by having users involved.

Measuring the number of hours during the sprint that users were actively involved is a good start. Here's

the best part, though: It can't just be worker-bees from the business unit.

Case Studies: How [Hotels.com](#), [Lotus F1 Racing](#) and [Zappos](#) Use Agile

Why does an executive need to be actively engaged in agile? Of course, sponsorship and championing are important to motivating the organization, pulling them into the change management that is inevitable with significant IT projects.

But there's more we need from the execs, too. They need to articulate the priorities and rationales at a level of detail below platitudes and bullet points. The sprint teams need execs to help make the tough priority calls, providing the users with enough guidance and decision-making authority so that the worker bees can make decisions without being over-ridden.

When it comes to a measurement of executive participation, perhaps the best is "number of decisions that were overridden or had to be escalated to a VP to be resolved." Lower is better here.

Agile Makes Bean Counters Happy After All

The reality here is that agile doesn't give you beans to count the way waterfall projects do. Surprise! Here's an MBA who says, "Good." Beans aren't going to make your business any more profitable.

If you want more profitability, the key success factors are working on the things that matter, avoiding the stuff that's just make-work and ensuring users actually use what you produce. If you relentlessly focus on business value, you'll be able to edge the savvy CFO toward agile even though it looks "uncontrolled."

David Taber is the author of the new 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](#).

© 2012 CXO Media Inc.