CIO

Print Article Close Window

From: www.cio.com

Is Your Cloud Support Agile? Take Our Quiz

– David Taber, CIO

January 16, 2012

As I've written endlessly, cloud projects are a nearly perfect fit for agile project development techniques. Even if your finance guys insist upon fixed price and fixed schedule, the tendency in the cloud is iterative development and scrum teams.

But agile really is focused on development —all the agile manifesto authors came from the development discipline — and not on the ongoing obligations of system support. Where agile optimizes speed and business value of innovation, it doesn't have much to say about MTTR or cost optimization. Do agile principles help in support organizations? You bet! Do they pan out in your support team? You be the judge:

Are Your Developers Doing Support?

Serious points off if they are. A key metric of the support function is the capability to use low-cost resources to solve all but the most intricate problems. Sure, your developers can probably do support faster and better than a level-one support tech, but that hardly means they should. (All too often, the developers won't even bother to close a case, let alone improve checklists and tests.) If your developers can't extricate themselves from support cases, it means they can't be doing higher value work. And it means your support organization can't go down the experience curve. It's also a pretty clear indication you've got a problem with&

Is There Sufficient Support Knowledge Base?

The support team needs FAQs, troubleshooting checklists, escalation rules, release notes, application notes, revised test vectors, normative log files, and training materials. All organized into a solid Knowledge Base. Unfortunately, many agile developers eschew the creation of these documents during the development cycle. It's viewed as overhead, bureaucracy, and —in some of the more extreme sects of agile —as a misleading abomination ("the code is the only documentation that matters!"). If you can't get the developers to do documentation, you'll need an explicit technology transfer function to transform the production artifacts into a Knowledge Base that the support organization and customers can digest. Make sure your deployment schedule and budget have placeholders for this transformation process (which will have to be continuous and repeated with each iterative release).

Is the production system continuing to use continuous integration principles, with scheduled periodic pushes?

In enterprise-scale cloud systems, there are enough bug fixes, data corrections, user-privilege modifications, and access control updates that configuration change needs to be managed on an ongoing basis. And the right way to do that is by leveraging sandboxes, snapshot/diff tools, continuous integration/test cycles, and automated pushes to production. All too often, support organizations apply their changes directly to the production cloud — because it's "too important to wait." This practice tends to spawn new bugs, and can lead to chaotic degradation of the support process. That isn't agile, that's just unruly.

Are Tickets Being Managed as Ranked Agile Cards?

One thing that cases and bug reports share with agile cards and stories is atomicity. So they can be filtered, sorted and ranked with similar techniques. But the standard severity and priority indicators of support cases don't provide enough business context for making good Agile priority decisions. While severity (take a look at VMware's definitions) gives a clear indication of the functional impact and Priority provides the SLA for response time, it really helps to understand Business Impact (how many users or dollars are affected) so the support team can make good tradeoffs. Particularly for Cloud applications, it's important to have support thinking along profit-maximizing (or at least user-delight) lines, as "cost center" thinking leads to poor resource allocations.

Are Feature Requests (RFEs) Being Budgeted in the Same Way as Escalated Tickets (Cases)?

A case really is different from an RFE. But an escalated case that needs to be handled by the same developers as a new feature cuts into their time and budget. So the resource allocation should be managed in a coherent way to prevent over-runs.

How Much Time Does Support Spend in Meetings?

This is a centerpiece of the agile movement: keeping developers away from bureaucracy, red-tape and unproductive meetings. But agile doesn't mean "fewer meetings" —it means shorter, better ones that directly involve the user. The agile "stand up" meeting is something you should be seeing on a daily basis in your support organization, and it should be short and punchy and&nobody should be sitting down. Points off for any meeting that last more than 15 minutes. Double that if anyone in the meeting yawns —the surest indication of unnecessary discussion.

What New Metrics Have You Put in Place for Agile Support?

All the standard support metrics are applicable to an agile support team. But it's worth monitoring a few new ones each month:

- Number of times an out-of-cycle push had to be made from test to production (this is an indicator of (1) process violations, or (2) bugs or problems caused by a "fix").
- Number of KB / FAQ contributions made (this is an indicator of the progress of post-hoc documentation).
- Number of changes/additions pushed into production without a corresponding KB contribution (this is an indicator of whether you're falling behind or not).

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, and David has over 25 years experience in high tech, including 10 years at the VP level or above.