



## 11 Ways Agile Adoptions Fail

By Jean Tabaka



Hear more about this topic in the [StickyMinds SoundByte podcast interview with Jean Tabaka](#).

Agile methodologies have been taking some heat for when they appear to have failed to deliver expected benefits to an organization. In my travels as an agile coach, what I have found to be the case is that agile practices don't fail, rather the variations on agile adoption fail. Here are my top eleven failure modes. See which ones look painfully familiar to you:

### *1. Ineffective use of the retrospective*

Agile is all about inspecting and adapting. When I walk into an organization to help it evaluate its adoption of agile, my first question is, "When was the last time you ran a retrospective, and what did you do with the recommendations from that meeting?" Agile adoption hungers for the nutrients that real retrospection brings.

### *2. Inability to get everyone in the planning meetings*

I've worked with several organizations who insist they are "agile" and yet still have a subset of team leads completing all the iteration plans: what will be delivered, what the estimates are for delivery, and who is assigned to meet these delivery commitments. Agile requires full-team collaboration on any commitment; without this level of commitment, a team stays stuck in the wear and tear of command-and-control decision making.

### *3. Failure to pay attention to the infrastructure required*

A good "first order" practice of agile teams is to adopt the regular flow of functionality through immutable time boxes. But time boxes alone do not make an agile team. Real agile adopters constantly inspect and adapt their environment to support the elimination of waste and increase the continuous flow of value.

### *4. Bad ScrumMasters*

I work primarily with Scrum teams, and the teams that struggle the most are those in which the ScrumMaster has a history of command-and-control project management, or a decision-oriented technical lead. Unless these ScrumMasters move to a facilitative, servant-leader mode of team guidance, agile adoption will be only a thin veneer over non-empowered, demoralized teams.

### *5. Product Owner is Consistently Unavailable or There are Too Many Owners Who Disagree*

Product owners not passionately engaged in the agile adoption hold the software development team hostage until the expected benefits have been delivered. An unavailable product owner perpetuates the wait time and creates waste, which is a problem that more traditional approaches are known for. Multiple product owners who don't agree are the same as a non-existent product owner: They create waste as the team waits for them to come to consensus about their priorities.

### *6. Reverting to Form*

Agile is simple but hard. Giving up when it feels hard undermines the ability of the adoption to succeed. Teams need time and support to adopt the disciplines necessary for agile success.

### 7. *Obtaining Only "Checkbook Commitments" from Executive Management*

I call this "phoning in their role." An executive who announces "We are going agile," but does not provide rich resources, rich commitment, and readiness to change measures of success, simply demoralizes teams. Successful adoptions of agile can be associated consistently with a passionately engaged executive sponsor.

### 8. *Teams Lacking Authority and Decision-Making Ability*

The Agile Manifesto stresses individuals and interaction. It also stresses collaboration and self-organization. Like No. 2 and No. 4 above, an agile adoption will fail if a team truly does not own its decisions, assignments, estimates, agreements, and its ability to alter these when the team learns something new.

### 9. *Not Having an Onsite Evangelist for Remote Locations*

Agile adoption for distributed teams and offshore teams is difficult in the best of circumstances. These remote satellite locations not only need rich support for increased communication, they also need their own onsite evangelist who can keep the team engaged and in tune with how team members either are benefiting from agile or are misinterpreting it.

### 10. *A Culture that Does Not Support Learning*

Agile practices weave nicely into lean thinking. One of the seven tenets of lean thinking is "amplify learning." Like No. 1 above, an agile adoption effort that neither supports learning at the team level nor elevates it to the organizational level will result in frustration and rejection of the approach.

### 11. *Denial is Embraced Instead of the Brutal Truth*

The final truth about agile adoption is that it requires brutal honesty. Organizations must be willing to accept the truth of the high visibility that agile practices bring with them. Daily meetings, task boards, time-boxed inspection of priorities, estimates, assumptions, and issues all take a toll on organizations that are used to hiding their progress. A team that hides its broken builds or an organization that ignores its roadblocks ultimately will blame agile for its missed commitments and revert to older, established form.

So, as you consider a move to agile software development, or if you are in the midst of an agile adoption that is causing you pain with no gain, consider this list of eleven signs that your adoption really may not be about agile. Rather, you may be sitting in a Potemkin village of agile facades that needs razing before your adoption truly can be considered agile.

**Want more articles on agile and other best practices? Subscribe to *Better Software* magazine for FREE today at [www.BetterSoftware.com/agarts](http://www.BetterSoftware.com/agarts)**

### **About the Author**

**Jean Tabaka** is an agile coach with Rally Software who specializes in creating and mentoring agile software teams. Jean brings more than twenty-five years of experience in software development to the agile plate in a variety of organizational contexts including internal IT departments, ISVs, government agencies, and consulting organizations. A Certified Scrum Master, Certified Scrum Trainer, and Certified Professional Facilitator, Jean holds a master's in computer science from Johns Hopkins University and is the author of [Collaboration Explained: Facilitation Skills for Software Project Leaders](#). She can be reached at [jean.tabaka@gmail.com](mailto:jean.tabaka@gmail.com)

StickyMinds.com Weekly Column From 6/4/2007

© 2012 StickyMinds.com. All rights reserved.

StickyMinds.com is a division of [Software Quality Engineering](#).

[Privacy Policy](#) [Terms & Conditions](#) [Link to StickyMinds.com](#) [Feedback](#)

Reprinted with permission from the author.