

# PICTURE THIS: FIRST EXPERIENCE WITH AGILE? SIDESTEP THE PITFALLS.

BY KELLY JONES

**ICTURE THIS:** You work for a company that has always specialized in the waterfall methodology, wholeheartedly believing in its benefits – comprehensive analysis, heavy focus on documentation, high predictability of schedule, etc. Meanwhile, you hear rumors of a new software delivery process that trades comprehensive analysis for piecemeal analysis and replaces heavy documentation with function-specific cards. These rumors turn real when your colleague becomes sold on the idea of Agile methodology and its benefits – smaller, quicker releases of functionality, more nimble response to changing needs, and a process they believe produces a higher quality result. And now your colleague wants you to transition the waterfall development process of your company's CRM application to the Agile methodology.

This transition can go one of two ways. As a member of the application team, you may go through this process hitting every possible obstacle along the way and end up wondering why Agile is such a good idea. Or you can educate yourself on the potential pitfalls, do your best to avoid them, and greatly improve your chances of success.

The transition process may seem daunting, but Agile – done right – can indeed be the key to developing software that meets the customer's needs, performs just as expected, and delivers return on investment. As you plan your transition to Agile, consider these 10 most common pitfalls and their proactive solutions.



#### **PITFALL #1**

The team spends more time sitting in meetings than getting work done.

#### **PITFALL #2**

Environments never have the right new functionality in them.

#### **PITFALL #3**

There seem to be more defects per story card than words in the story.

#### **PROACTIVE SOLUTION #1**

True – the Agile methodology suggests a certain schedule of various types of meetings. But who’s to say your team needs to have all these meetings for the sake of meetings? Take a look at your team and project’s needs, and customize the purpose and frequency of your meetings. For example: Agile methodology advocates tasking meetings and estimation meetings. Tasking meetings are for the developers to discuss a card for the upcoming iteration and agree upon the step-by-step tasks that need to be accomplished in order to complete the work on the card. Estimation meetings are for the developers to assign an agreed-upon level of effort to the card. If you have a small team, you might combine these gatherings and have the developers estimate the level of effort immediately after discussing the tasks. By combining the purposes of two meetings into one, you’re freeing up your people’s time to focus on the work at hand.

#### **PROACTIVE SOLUTION #2**

This is a code branching issue, not an Agile-specific issue. However, because code is stored in smaller packages in Agile, there are a lot more branches in this world. Multiple developers touch different pieces of code in various environments, and without oversight to ensure that the code in the environments changes as the iterations change, functionality may or may not make it to the environment where it belongs, even in production. The key to environment consistency is to assign a developer to own each environment and ensure the right branches (based on iteration or release) are in that environment, regardless of which developer did the work on the branch.

#### **PROACTIVE SOLUTION #3**

Agile 101 dictates that a developer knows when to stop coding once all acceptance criteria are developed. Hold developers to this principle! Implementing unit testing accountability on your development team also makes a big difference. Often, this situation arises simply because story cards become too complex. If this is the case, consider using one large “epic” card to detail a large effort,



and link it to story cards for specifics on smaller pieces of the epic. Development of smaller, more manageable cards reduces the chance of a misunderstanding between the story and the developer, and makes it easier for the business analyst to clearly describe the business need. Ultimately, this means fewer defects.

#### **PITFALL #4**

Downstream changes are repeatedly filed as defects.

#### **PROACTIVE SOLUTION #4**

In the Agile world, developers are empowered to create the best solution for each card they work on. This leads to changes that occur downstream in the software development lifecycle. When a piece of functionality works differently than it has in the past, it can be interpreted as a defect if all stakeholders aren't aware of the change. Meanwhile, the developer will insist that it works just as designed. Here's another Agile 101 principle: Communication must go in all directions. When a design change happens in an application, communication must extend to all stakeholders, upstream and downstream. You'll find that many perceived "defects" aren't defects at all. With Agile, there's no such thing as over-communication!

#### **PITFALL #5**

Applications that interact with your software complain of broken processes.

#### **PROACTIVE SOLUTION #5**

See solution #4 – everything that's true about communicating within a team goes for communicating across applications. Processes like web services suddenly seem to be "broken," because they're not behaving as expected? Chances are, a change took place that didn't get communicated to all affected parties. Upstream and downstream cross-team communication must include all people impacted by your team's software.

#### **PITFALL #6**

Trying to reference original requirements on story cards is a hassle.

#### **PROACTIVE SOLUTION #6**

When functionality or process behavior comes into question, it can be difficult to check back on the original requirements if you can't find the right story card. From the beginning, make sure to manage your project using software with effective search capabilities. A really effective card management application would also include standardization and customization of the card template. Work with

# Here's another Agile 101 principle: Communication must go in all directions.

an application missing these features, and you might find yourself dreaming of the bygone days of master requirement documents.

## **PITFALL #7**

Developers describe story cards as “too technical.”

## **PROACTIVE SOLUTION #7**

Business analysts are often worried that if they don't write enough detail, the functionality won't precisely meet the business needs. Developers, on the other hand, are often concerned that all the detail on the story card doesn't actually present the optimal solution. Ideally, a designated solution architect would take an all-encompassing view of the project from both the business and technical perspectives; but if that's not a possibility, create working sessions where business analysts and developers can discuss possible solutions for each user story. This way, the story truly serves as a conversation starter. The business analyst can present the story and business specifications, and the developers can, in turn, offer potential technical solutions. Then the business analyst can feel confident that the chosen solution meets the business needs, and the developers will have ownership of the solution and feel comfortable with its implementation. Again, this gets at communication – one of the cornerstones of successful Agile methodology and a key to helping IT teams deliver on business goals.

## **PITFALL #8**

Current cards are hung up in delays while business analysts and developers wait.

## **PROACTIVE SOLUTION #8**

When business analysts get held up waiting for replies from the business and the iteration gets slowed down, it's likely because more analysis time was needed before the card was selected for the current iteration. If this happens on a regular basis, it becomes a problem. One tactic to ensure a business analyst has ample time is to enforce a hard cut-off date for the submission of new cards prior to the start of an iteration. Depending on the size of the overall organization, a few days to a week should be sufficient. If neither business representatives nor business analysts are allowed to miss this deadline, it prevents the flood gates from opening for last-minute changes. It may also be helpful to set the guideline that once a card is in the iteration, any changes that impact requirements will result in the card being pulled from the current iteration. Getting these habits established will set

# Rest assured, change always feels difficult at first, even if you're doing everything right.

the standard for high quality work from the beginning by preventing scope creep, unforeseen delays, and hurried efforts.

## **PITFALL #9**

Every iteration ends with commitments left incomplete.

## **PROACTIVE SOLUTION #9**

Three possible causes lead to this: 1) Acceptance criteria are being added or changed, resulting in scope creep that developers cannot accommodate; 2) Cards are not being estimated accurately; or 3) Business analysts do not have enough time to sign off on each card before the last day of the iteration. Solution #8 addresses the first possible cause. Avoiding the second cause may involve a little trial and error. It should help to take the highest time estimate thrown for each card as the final estimate (versus allowing the developers to meet in the middle). This makes it more likely that no matter which developer picks up this card to work on, it will be completed in the estimated time. If this doesn't work, track the reasons for each piece of work that causes an overage; then enlist the developers in addressing each of these causes every time a new piece of work is estimated. To avoid the third possible cause, make a practice of having any cards requiring business analyst sign-off (basically, any features that are not IT husbandry) be the first cards picked up for development in the iteration. This prevents business analysts from receiving a stack of cards ready for sign-off on the last day of the iteration.

## **PITFALL #10**

The whole transition feels like pushing a square peg into a round hole.

## **PROACTIVE SOLUTION #10**

Rest assured, change always feels difficult at first, even if you're doing everything right. Some of the basic change management principles work well in Agile transitions: secure buy-in from leaders and managers; find a few members of your team to act as a coalition driving the change; and celebrate small wins when things go well. And before you jump in and involve your team, start by learning about Agile's many flavors. A combination of slightly different methodology variations may be the ideal option for your team.

Bear in mind that Agile development is easier to implement for smaller pieces of software and that it takes a good bit of customization to meet the needs of a complex application that



interacts with multiple types of software (especially if it becomes necessary to blend it with waterfall development). These proactive solutions may head off all trouble for small to medium Agile engagements; but if your organization requires a larger, blended-methodology, a more complicated transition is to be expected. Having a dedicated solution architect on hand will help you account for the solution as a whole. And it bears repeating once more: communicate, communicate, communicate.

#### ONE FINAL NOTE: IT'S ALL ABOUT PEOPLE

When a team is made up of individuals who are proficient in their roles, change is tough enough. When dealing with team members who aren't well suited to the task, it's monumentally more difficult. To make sure your team is ready to roll with the changes, manage any team member issues up front. If there's a knowledge gap, address it with training. If there's a willingness or capability issue, repurpose the team member to a different role or team. To take on a transition like this effectively, everyone needs to be on board. The change may not be simple; but in the end, it can be very effective at creating top-quality solutions that suit your customer's needs perfectly.

---

KELLY JONES ([kelly.jones@jabian.com](mailto:kelly.jones@jabian.com)) is a Senior Manager at Jabian, where she has gained experience in implementing IT transformation strategies and IT enabled solutions that create value for clients. Her passions lie in strategy, product management, and helping organizations realize the value of change. Kelly is also passionate about continuing education; she earned her MBA from Emory University and BBA from The University of Georgia.

---

© 2013 Jabian, LLC. All rights reserved. We'd love to hear your feedback on this article. Please send your comments to [journal@jabian.com](mailto:journal@jabian.com).