Mindtree
Welcome to possible

Agile – Principles over practice

Blog series

Dwight Kingdon
Consultant and Agile Coach

Dwight Kingdon is an Agile Coach at Mindtree. He is a thought leader in Agile methods and best practices, leveraging many years of software development, analysis, project management and leadership experience. Dwight has 25+ years of project management experience leading complex information technology projects, and over nine years of Agile/Scrum experience coaching and leading high profile, mission critical projects.

Part 1

I gave a short presentation in August at the Agile 2014 conference in Orlando titled "Agile – Principles over Practice". I believe that prioritizing practices over principles is one of the key reasons for Agile failures. In this, and future blogs, I'll dig a little further into the root cause of Agile failures and touch on avoiding some common pitfalls.

I've been in the IT industry for 25 years, and was introduced to the Agile framework nine years ago. For me, Agile was love at first sight. I think the reason I took to Agile so readily was that many of the principles were already part of my Standard Operating Procedure. I valued simplicity, I developed software in iterations so I could get feedback and make course corrections, I preferred communicating face-to-face, and I put my customers and developers in the same room often so they could collaborate and ensure we were on the right path.

What I was doing in my early years wasn't truly Agile, but the principles I followed – many of which are in line with the Agile principles – allowed me to consistently deliver projects on time, on budget, and with the quality that my customers loved.

Some time back, I consulted with a company that brought me in to work with them on their Agile journey. A couple of their IT leaders told me that they were already Agile. They had all the right roles in place, their ceremonies seemed to be well run, and the artifacts looked clean and up to date – they looked like they had it all together.

But when I asked tough questions and began digging into how they interact with each other, I saw a different picture. There was distrust between IT and the business units because of which there was no real collaboration, almost all their communication was by email for "CYA" purposes, and there was a heavy status reporting burden due to lack of transparency.

The practices all seemed to be there – on the surface it looked like things were running smoothly, but underneath the calm was a broken machine. They were doing all the easy parts of Agile – the practices, but weren't paying attention to the harder parts of Agile – the principles. Real Agile transformation is a total culture change. Successful Agile implementation is about establishing a culture that embraces open communication and collaboration between business and technical people across the enterprise. It's about continuous improvement through inspection and adaption, and a culture of transparency and accountability.

I've learned that you can do all the practices perfectly and still fail at Agile. Not paying attention to the principles behind Agile is one of the leading causes of failure. You may see some improvement in your ability to respond to change, and you might even provide working software faster – for a while. But it won't be too long, though, until the promises of Agile become less evident. Teams struggle to keep up the pace, software doesn't match user expectations as often, and then Agile is deemed a failure. Some will go back to their old ways – ways that are more comfortable.

Incorporating Agile principles is a harder part of Agile – things like working through communication issues, distrust and lack of accountability. Working with the principles often gets left out because that takes more time and effort than we may want to invest. But it's almost always the harder things that give us the most benefit. Culture change – the principles... those are the harder parts of Agile.

Part 2

Incorporating Agile principles is the harder part of Agile – things like working through communication issues, distrust and lack of accountability. We often might skip working with these principles because it involves more time and effort than we may want to invest. But it's almost always the harder things that give us the most benefit. Culture changes and principles... these are the harder parts of Agile.

So what are some of these principles or culture changes that are the harder parts of Agile?

One of the first difficulties is communication. I often see people using the Scrum Master to deliver their messages to others in new Scrum teams. They are used to having a project manager, and a Scrum Master is a whole new ball game. I was once a Scrum Master working with a brand new Agile team, when three different team members came to me, trying to communicate "through" me during our first sprint. One of them said- "Dwight, I need this information from Debbie before I can start coding." Debbie was only 20 feet away from us, but they were accustomed to having a project manager handling all their communications. A key Agile principle is communicating face-to-face whenever possible. But for many, face-to-face communication can be scary, especially when they're used to living in their cubicle without having to talk to people very often. So each time this happens, I immediately walk the team member over to the other person's desk and have them communicate the message directly. Help your teams get into the habit of communicating face-to-face, using that as their first option instead of as a last resort. Wasted time is not Agile.

Collaboration is another one of the hard parts. Agile is all about people working together, talking together and coming up with great ideas together. We are fundamentally at risk whenever someone works on something by themselves. At our Mindtree Gainesville Delivery Center, our Agile pods/teams have video walls with large screen live video feeds all day long for teams that can't be physically in the same location. This creates a virtual collocated workspace. Team members at any location step up to the video wall and collaborate together, face to face, as though they were in the same room.
A collaboration issue I sometimes see is when a waterfall project manager becomes a Scrum Master. Project managers can make great Scrum Masters, but watch out for those who bring a command and control style along with them. Agile is not about command and control, but some new Scrum Masters tend to manage and direct instead of collaborating with stakeholders and the team. Great Agile teams are not managed- they're encouraged to collaborate and figure things out themselves. Often, the lesson is learned better by experience (good or bad) than by being told what to do.

I also see a similar pattern with new Product Owners, except that here, the behavior is the opposite. Many new Product Owners are used to creating requirements, handing them off to the development team and then sitting back and waiting for the finished product, with little or no further involvement. Continuous collaboration between the Product Owner and the team, along with timely course correction, helps ensure that what's delivered at the end of the sprint is just what stakeholders intended. But this can be time consuming for the Product Owner, and many new to the role are either not ready for the commitment, or just don't know that they need to be so involved. I encourage Product Owners to get involved with the project team throughout the sprint. This way, the Sprint Review becomes purely a formality, as the Product Owner has already seen several iterations of the software during the sprint and guided the team to build exactly what the stakeholders want. This is what an ideal work situation should be like.

In conclusion, collaboration and communication are two of the main hurdles dealing with corporate culture and principles that we need to get ourselves and our companies to do well. They can make a big difference! The principles can sometimes just seem like common sense, but common sense is in reality, not very common! We have to make a conscious effort to execute the harder parts of Agile, and not just the easier parts, i.e. the practices.

Part 3

As stated in my earlier blogs, incorporating principles is the harder part of Agile – things like working through communication issues, distrust and lack of accountability. We often might skip working with the principles because it involves more time and effort than we may want to invest. But it's almost always the harder things that give us the most benefit. Culture changes and principles – these are the harder parts of Agile.

This week, we'll look at three more of the more challenging parts of Agile – Transparency, Retrospection and Simplicity.

Transparency is being honest – telling it like it is; it's sometimes being vulnerable. I once had a leader at a client's place who asked me to spin the reporting of a project's status so it would look better to senior leadership. I explained the importance of transparency, which he humbly accepted, and the resulting frankness set a precedent of honesty and openness that continued, no matter what the results were. We can't fix things if problems are swept under the rug. In Agile, transparency means bringing issues out in the open where they can be dealt with.

Agile lifecycle tools like Rally and Version One are great, but for teams and stakeholders to see what's really going on, they have to take time to go into the tool and find the information in order to see it. In my experience, very few people outside the team take the time to do that regularly. I encourage teams to make projects more visible by putting large Scrum boards and Burn-down charts on the walls and showcasing this information to the world. I learned that leadership really appreciates the visibility this brings to projects. This can also help reduce the need for status reporting since anyone can look at the team wall and see exactly what is going on.

Another transparency tip I learned stems from the fact that many new Agile team members are reluctant to raise obstacles during stand-ups. One of the primary functions of the Scrum Master is to remove obstacles so the team can focus on delivering software. But if obstacles are not raised, the Scrum Master can't help remove them. I remind new teams at the beginning of every stand-up meeting to bring up even potential obstacles, if there's any chance something might delay their work or cause them to not live up to their sprint commitment. The quieter team members especially need encouragement to speak up. I like to reward those who speak up and raise obstacles in the first few sprints, especially the quieter people, by recognizing them.

Retrospection (inspection and adaption) is one of the harder parts of Agile. I like to compare retrospection to software documentation. Software documentation is often not done because it's considered low priority compared to moving on to the next project. In the same way, some Agile teams skip Sprint Retrospectives because they think they don't have time, or they don't see the value of doing them. You can only have continual improvement when you pause to reflect on what's working or not working, and making a conscious decision to adjust things. Little tweaks here and there can make the difference between success and failure of a project. It is hence vital to conduct Retrospectives after every sprint and actually implement the valuable changes that come from the retrospective. I heard someone say once that "action without reflection leads to burnout, and reflection without action leads to cynicism".

Finally, Simplicity is one of the harder parts. Agile is very simple – the four values in the Manifesto and 12 Principles. Scrum has three Roles, three Artifacts and four Ceremonies. Everything else out there is someone's attempt to improve on the Agile philosophy. Some of these ideas are great additions and some just complicate the simple concept of Agile. Software documentation is a great example of how Agile can simplify our projects. There are two keys to effective but simplified Agile documentation. The first is to minimize artifacts to only what's really needed to get the job done. Documentation still has value, don't get me wrong – corporate governance still exists and usually requires specific documents, and new software has to be maintained and typically needs support documentation. But, there's sometimes a tendency to create documents just because "we've always done them before", whether they have value or not, whether they're ever read or not. Creating those documents uses up valuable sprint time. We need to understand and weigh the true cost of creating each document against the anticipated benefit. The second key to simplify Agile documentation is to minimize the effort to create artifacts. You can make it look fancy later if that's really important. If a hand-drawn and scanned diagram is sufficient for the team to move on, that's a great time saver.

These are a few of the corporate culture and principle-related harder things to get ourselves and our companies to do. And they can make a big difference.

The principles can sometimes just seem like common sense, but common sense is in reality, not very common. We have to make a conscious effort to do the harder parts of Agile, and not just the easier parts – the practices.

So, while Agile and Scrum practices are important, practices alone often lead to Agile failure. Agile principles and changing of software development culture are generally the harder parts, but they are what make Agile sustainable in the long run and maximize the great benefits it has to offer.

So what's the next step? Look at your team, pick one of the principles or areas of culture change needed – one of the hard parts – and help your team do what they need to do. Teams that keep working on the principles and culture-change hard enough and long enough will become the Agile elite, the few that truly understand Agile, and will unleash the true power that Agile has to offer.