Adopting an Agile Method

Articles

Posted by:

Posted on : 2007/9/19 22:50:00

The argument has been made: "We should be using an Agile software development method." And the command has rung out: "Make it so!" Now what do you do? How do you take that one-line "requirement", and make it so?

Adopting an Agile method is no different from any other change we might make to the methods and tools we use. We must determine why we are embarking on this course, choose the method that will satisfy the need most closely, then map out the path from where we are today to where we need to be. Then we can "make it so".

Author: Alan S. Koch, http://www.ASKProcess.com

Why Adopt an Agile Method?

Before we start changing things, lets first step back and be sure we understand what it is that we are trying to fix. After all, if there were not a problem with the way we do things today, there would be no reason to change. So, our starting point is the original argument for adopting an Agile method.

If there was significant discussion before the decision to adopt was made, then the information we are looking for may already exist. If little was discussed (or if there is no record of what was discussed), then you will need to work with the person who suggested the change to understand why it was suggested. Are there problems with the methods we use today? What are they? How bad are they? Who is affected by them? Or is there the expectation that although no large problem exists, an Agile method would allow us to improve over our current methods? What sort of improvement is anticipated? How significant is it expected to be? And who might be the beneficiaries of the improvement?

We also need to discuss with the decision-maker why they decided to approve the idea. What did they envision? What benefits do they expect? How do they think things will change as a result of adopting this new method? And most importantly, what would be this person’s definition of "success" in this endeavor?

Of course, when we discuss these things with multiple people, we may find that there has not yet been a meeting of the minds on exactly why we should adopt an Agile method. In that case, we must bring the diverging minds together (preferably in the same room) to discuss their different perceptions and agree on exactly why we want to do this.

Regardless of what it takes, we must end up with a clear and consistent description of the reason why we are adopting a new way of working. Indeed, without this, we are virtually assured of failure, and some of the major constituencies are likely to be unhappy with the end result. Conversely, with a clear understanding of the "why" of the effort, we are equipped to move forward with confidence.

Adopt Which Agile Practices?

Now that we know what we want to achieve, we can examine the various practices that we could adopt and decide which of them will be appropriate in our organization. This is a piece of the process that is easy to miss, especially if the mandate is to adopt a specific Agile method (like eXtreme Programming or Scrum). But even if a specific method has been prescribed, it may not make sense to adopt all of the practices of that method. We need to carefully consider the uniqueness of our organization and the projects we undertake to decide which practices will fit in, and will help us in achieving the goals we have identified.

The major areas that we must consider are:
* Our organization’s culture
* Our customers and how they prefer to interact with us
* The types of projects we do
* The tools and processes that we currently use
* The strengths and weaknesses of our software-related staff
* Organizational Culture

All of the Agile methods are built around the concept of a self-directed team. The development team is not told what it will do and when it will be done by. Rather, they are given broad direction about goals, and then they work with the customer to determine how to reach those goals.

If your organizational culture is build on a "Command and Control" style of management, then many of the practices of the Agile methods will represent a serious challenge to the organization. The managers will need to alter their management methods and adopt a more collaborative relationship with the development teams. This is a very difficult transition for many old-school managers to make, and could be a significant stumbling block to the successful adoption of any Agile practices.

Another organizational issue revolves around planning and commitment making. If your organization expects plans and requirements to be finalized up front, and then adhered to throughout the project, then the Agile methods’ incremental planning and requirements definition practices will be problematic. The Agile methods assume that we cannot know all of the things that will come up during the project, so they plan and define requirements at only a high level in the beginning. Then they add detail and revise them incrementally throughout the project, always keeping the project’s broad goals in sight.

Examine the practices that you expect to adopt, and for each, determine the extent to which it can be implemented given the realities of your organization’s culture, and the benefits you anticipate can be reasonably expected to accrue.

Customers

All of the Agile methods expect significant participation of the ultimate customer of the project with the development team throughout the project. This is often accomplished by having a representative of the customer working closely with the team on a regular basis. (Different Agile methods carry this model to various extremes – with eXtreme Programming being the most extreme.)

How active are your customers in your projects now? And more importantly, how readily would they increase their participation if you asked them to? Most customers have only limited resources to devote to the project, and any significant increase in interaction could represent an obstacle to them. Would your customer be able to commit the time and effort that the Agile practices expect of them?

Another consideration is the details of your contract with your customer. If you are developing software under contract, then the rules for interaction may be carved in stone and essentially unchangeable. Even if changing the rules of engagement were possible, would your customer be willing to do so? Often, companies use contracts to protect themselves from the organization they are contracting with. If your customer takes this view, then they may not be interested in collaborating more closely with your development team.

Examine the practices that you expect to adopt, and for each, determine the extent to which it can be implemented given the realities of your customers, and the benefits you anticipate can be reasonably expected to accrue.

Projects

Every project is unique and represents unique challenges and opportunities. Nonetheless, most organizations tend to take on projects that have a relatively predictable set of attributes. Some companies do mainly financial and administrative systems. Others do real-time embedded systems. What kinds of projects to your development teams generally work on? The types of projects that your teams take on have a certain amount of uncertainty to them. They represent a certain risk profile. And they include a certain amount to technological innovation.

Although the Agile methods were mainly developed to meet the needs of small projects that expect significant turbulence and change, they can also be used in many other kinds of environments. Most of the Agile methods expect that the development team can work face-to-face on a daily basis. If this is not the case in you organization, then some of the Agile practices may need to be modified to work with a distributed team.

The promoters of the Agile methods tell us that their methods can and should be used on any project. Indeed, some of them are experimenting with Agile distributed teams. While it may be true that Agile methods can be used on any project, the questions you must ask is, should you use the practices these methods prescribe on your projects!

This means going back to the purpose you identified for adopting an Agile method, and determining if the practices will support achieving those goals. Examine the practices that you expect to adopt, and for each, determine the extent to which it can be implemented given the realities of your projects, and the benefits you anticipate can be reasonably expected to accrue.

Tools and Processes

The methods we employ do not exist in a vacuum. They are strongly influenced by the environment in which we use them; and an important part of that environment includes the supporting tools and processes that we depend upon.

As things stand today, your organization already has a set of tools and processes in place. To what extent will those tools and processes be compatible with and support the new Agile practices you expect to adopt? Will your processes and tools (e.g. for requirements management, or for Quality Assurance) stand in the way of adopting Agile practices? And if so, is there latitude to change them (or get rid of them) in order to make the environment more conducive to the Agile practices?

The flip side is also true. Many of the Agile practices require specific tool and process support in order to be effective. For example, eXtreme Programming expects that automated testing tools are available to the entire programming team, and that they use them heavily. Are such tools available in your organization? And if so, are there enough licenses to allow this widespread use?

Another example is code control tools. Most of the Agile methods assume strong code control systems and strict adherence to the disciplines involved in using them. Liberal use of things like refactoring and shared ownership of code can only work in an environment that provides the necessary code control tools and processes.

Examine the practices that you expect to adopt, and for each, determine the extent to which it can be implemented given the realities of your tools and processes, and the benefits you anticipate can be reasonably expected to accrue.

Staff

As we noted earlier, the Agile methods expect that development teams will be self-directing. This means that instead of being told what to do, the team understands the objectives of the project, and they collaborate with the customer and each other to determine the most appropriate steps to take at each juncture.

In order to act autonomously, the team members must be willing and able to take on a new level of responsibilities and accountability. Although this may seem to be an easy step (based on the complaining we often hear from programmers), it is actually quite a leap to go from complaining about the status quo to taking on the responsibility for creating a new one!

Many of our programmers are not as bold as we might expect. They may have good ideas about things to change, but be unwilling to take on the accountability that comes with being a decision maker. To many folks who thrive on technical challenges, the messiness of project and customer challenges can be quite intimidating.

And of course, our development teams do not represent uniformity in capabilities. Some of our folks are indeed super stars! But others’ abilities are quite lacking. If we are honest with ourselves, we must admit that the average programmer on our teams is only average! And around half of them are below average. The Agile methods empower teams to direct their own work (in collaboration with the customer). It is true that as they work in this environment, our people will learn and grow in their ability to deal well with making project decisions. But how successful can they be at the beginning? And at what point will they reach their maximum potential?

Examine the practices that you expect to adopt, and for each, determine the extent to which it can be implemented given the realities of your staff today, and the benefits you anticipate can be reasonably

expected to accrue.

Adopting an Agile Method

After we have considered all of the things we just discussed, we will be equipped to decide which Agile practices would be appropriate in our organization, given our culture, customers, projects, tools, processes and staff. Armed with this information, we can evaluate each Agile practice against our objectives, and determine which ones we should adopt, and if we need to customize them in any way.

In essence, we will be "rolling our own" Agile method. But isn’t that what we need to do? We don’t want to adopt something that might have worked in some other organization. We want to adopt what will work in our company. We want to do the right thing so we can achieve our goals in this adoption, and improve our ability to do a good job of building good software. Originally published in the Spring 2006 issue of Methods & Tools