What Are We Arguing About?
A Framework for Defining Agile in our Organization

Kelly Weyrauch
Sr. Principal Software Engineer, Medtronic, Inc.
kelly.weyrauch@medtronic.com

Abstract

How do we implement Agile? Is Agile a good thing for my organization? What barriers will I have to overcome? All of these are important questions when deciding whether and how to apply Agile Software Development to an organization that is used to working in a different way. To enable a good discussion, the language of that discussion must be established.

As Medtronic adopted Agile, we realized how important it was to put effort into the language. We established a framework for that language, organizing our thoughts into three areas: Principles, Practices, and Benefits. When talking with visionaries or evangelists, we speak in terms of principles to understand why Agile is the way it is. With practitioners or pragmatists, we speak in terms of Practices to understand how to do Agile. With skeptics, we speak in terms of benefits to determine whether Agile is a good thing for us. In most cases, an effective conversation needs to address all three perspectives. The framework and language help us determine whether issues are related simply to communication barriers, or are related to the decisions of whether and how to apply Agile to our world.

1. Introduction

Medtronic, Inc’s Cardiac Rhythm Disease Management organization is a manufacturer of safety-critical medical devices, such as pacemakers and defibrillators. The organization is large and very successful, with a well-established culture and a robust development process.

A pacemaker or defibrillator system consists of several major subsystems, including:

- An implantable device that consists of mechanical, electrical hardware, and embedded firmware subsystems
- An external programmer that consists of a computer-like hardware platform and supporting operating system software
- Application Software that allows a clinician to configure the operation of the implanted device and read diagnostic data from it
- A lead that connects the implantable device to the patient’s heart,
- Other supporting subsystems.

Both the development and manufacturing of these medical device systems are regulated by the U.S. Food and Drug Administration (FDA) and other regulatory bodies. This requires us to have a robust development process that both ensures and demonstrates the quality and safety of the products we create.

In such a world, any significant change to culture or process can be difficult. Sometimes it is difficult because of the inertia inherent in a large organization. Sometimes it is difficult because we believe we have a good process, making it harder to convince ourselves that a change is good and worth the risk of breaking something that works well. And finally, sometimes it is difficult because it is hard to articulate a vision of where a change will bring us, and what the path will be like.

When Agile Software Development was brought to the software group, we quickly realized that not only were we learning something new, we would also have to learn how to talk about it, how to organize it, and how to present it to others.

A Brief History

Our Agile journey started in early 2000 when a few people read books and experimented with some of the concepts. We invited many speakers from outside Medtronic to talk about this emerging topic, and the idea got more visibility, though not much immediate implementation. An advanced-concepts project outside
the “typical” development projects applied some practices of Extreme Programming.

At the XP/Agile Universe Conference in 2002, the four Medtronic attendees got an “epiphany”, realizing Agile concepts made so much sense that we had to apply them to a greater degree, and on a more typical development project. The four attendees included:

- A software project manager for a mainline development project who could envision the benefits to her project, and would be able to address issues related to project management.
- A software architect who could envision the benefits to the quality of the software product, and would be able to address issues related to the technical aspects of software development.
- A process architect (the author of this paper) who could envision the benefits to the quality of the development process, and would be able to address issues related to applying Agile practices in a regulated environment.
- A senior developer who was already using XP practices, and would be able to help the team get over the learning curve.

Along with others at Medtronic who were already liking and practicing Agile methods, this gave us a group of people who could influence the changes that would be needed.

Immediately after the conference, we had many discussions and a few simple experiments with some of the Agile practices like daily meetings, pairing, backlogs, stories, and sprint planning.

By the end of 2002, the software project manager had decided to use Agile concepts on her project. Other key project leaders got engaged, educated, enthused – whatever and whenever possible.

Most of the initial practices focused on project management (backlogs, burndowns, stories, sprints, etc.), partly because we thought those practices would demonstrate the most immediate benefit, and partly because we were at the early stages of the project where we were doing more planning and defining, and less designing and coding.

In 2003, we had a full set of practices in place, expanding the project-management practices and adding more of the engineering practices. Agile concepts also began spreading to other projects.

By 2004, most of the main-line software projects were using some variant of Agile. We were coming to a consistent understanding of the principles behind what we were doing, but still had a fair amount of variation in the specific implementations. Partly this was because we were still experimenting and improving, and partly because each project brought its own unique perspective on the practices.

Today, we continue to adapt our application of Agile concepts. At one level, this means we take on large questions such as “How can we better integrate the Development and Verification Test disciplines?” At another level, this means we regularly use such things as team reflections to adjust our use of Agile practices.

Growing well beyond the small set of zealots who drove the initial investigations of Agile, we now have many people in many parts of the software organization engaged in many ways.

The adoption of Agile has been mostly a bottom-up and middle-out effort. The software engineering teams are getting information, providing motivation, and adapting the implementation of Agile through a combination of normal development activities and targeted process-improvement activities.

Higher-level organizational management, though supportive, is not driving the application of Agile. This seemed difficult at first, especially since many of our guest consultants emphasized that management involvement was vital to the success of Agile. We concluded that the development teams and project manager had enough influence and authority to implement Agile, getting help from senior management when necessary. The need to have a “push” from senior management was not necessary in our environment.

Removing Barriers

Is this brief history sounding too simple, or too rosy? Well, a 4-year journey has to be simplified to fit into a few paragraphs, and the journey was not always easy. Early on, there were a few significant barriers we had to overcome.

First impressions carry a lot of weight, and in an organization that recognizes the value of a robust process in delivering high-quality products, any first impression that is seen as a threat to quality can result in the quick dismissal of a new idea. Some first impressions of Agile provoked an allergic reaction at Medtronic, with a hasty conclusion that Agile can’t work in our environment. To compensate, some of the obvious barriers to the successful adoption of Agile had to be addressed immediately. Here are a few barriers we needed to overcome.

Barrier: Agile says you don’t need a plan

The emphasis placed on adaptability and responding to change led some people to believe that an Agile project’s schedule was so flexible that you could never
or should never pin it down. Furthermore, the emphasis placed on the ability to adjust a project’s scope led some people to believe that the primary lever of Agile Project Management was to just lop off features when the deadline approached.

To get over his barrier, we needed to educate ourselves on the key elements of Agile planning. When we discussed the principles behind Agile planning practices, it was easy to see that they were compatible with sound principles of planning we already had.

We avoided controversy by accepting that we had to speak the planning language of the large project, even if that meant we had to put a translation layer in between what the large project wanted and what the software teams were doing for more-detailed planning.

We were able to demonstrate that the principles and practices we’ve chosen have put even more emphasis on planning. We frequently used a quote we heard in many Agile forums: “If you can’t plan well, then plan often.” We have emphasized the model of planning at different levels of abstraction, tailored to the specific needs of the individual. This has given us tools to plan better and have more control. We plan and re-plan constantly – exactly opposite of the original worry.

**Barrier: Agile says you don’t need requirements**

This barrier started with a simple, basic misunderstanding. The Agile world certainly has requirements; they are just called something else with a unique spin. Stories, customer-collaboration, and acceptance tests are all forms of requirements.

Once we got over that, we then had to deal with the requirements documentation we are accustomed to. We decided early on that we weren’t ready to simply abandon that type of documentation; instead we took Ken Schwaber’s advice to deliver business value. Since the business derived reasonable business value from requirements documentation, we needed to produce it. We focused our energy on creating that documentation in a more iterative, customer-centered way.

We still have room for improving the way we create and manage requirements, but we no longer have the anxiety caused by a mistaken notion that we should simply throw them out.

**Barrier: Agile is undisciplined**

Some of the early “selling” of Agile methodologies seemed to over-emphasize radical, break-all-the-rules, revolution-based concepts that didn’t play well. In our safety-critical world, we strongly believe that a robust process is an important element to ensuring high quality. The side effect of that strong belief can be an over-reliance on prescriptive, mandated process rules that take the approach of imposing discipline upon a team.

We got over this barrier by emphasizing the different kind of discipline that Agile fosters – the discipline of individuals and teams versus the discipline of a process document. With a team of people that thoroughly embrace the values of high quality, we realized there was little reason to worry that discipline would disappear. In fact, when we demonstrated that we weren’t changing the fundamental elements of our process as documented in our Quality System, this barrier disappeared.

### 2. Establishing a Framework

While there are many reasons our path to adopting Agile methods has worked so well, one important reason is that we were able to establish a framework for communication – something we could use when deciding what Agile was going to look like, when deciding implementation details for Agile practices, when training the team, and when communicating to others who work with us.

This framework consists of:

- Recognition of three perspectives to communication: the benefits Agile brings, the principles that drive why Agile works, and the practices that define how it works.
- Identification of the benefits we were hoping to achieve and relating those benefits to things the organization already wanted from us.
- Description of the principles that would drive our specific implementation of Agile practices, aligning those principles with guiding principles of our organization.
- Selection of the practices that define our implementation of Agile techniques, and guidelines for how those practices would evolve.

Each of these elements of the framework is explained below.
3. Perspectives

When someone asks, "How does Pairing work?" how do you answer? If they want to know how to implement it, you would explain how the desks are set up, how you decide who to pair with and when to trade off, and so on. If, however, they want to know what it is that makes pairing work, you would explain the principles of collaboration and learning that pairing supports. And finally, if they want to know how much pairing costs compared to how much value there is, you would demonstrate the benefits of reducing review-time or refer to the literature that discusses the value of pairing. We needed something to help us focus our discussions.

The ‘Why’ question is just as confusing as a ‘How’ question, and is often more loaded with controversy. When someone asks, “Why is Pairing a good idea?” they may simply be seeking out information about the principles behind it, or they may be challenging you to prove its benefit. We needed something to help us determine what kind of ‘Why’ question was being asked and answered.

Early in our adoption of Agile methods, our communication, training, and decision-making focused on the practices of Agile – how it works, what we do. We put together a set of slides using a metaphor of a play, with actors (customers, project managers), props (backlogs, burndown charts), and acts (planning, sprint execution). When we first presented The Play, a couple of invited reviewers said, “That wasn’t what I wanted to see.” After much discussion, the presenter and the reviewers determined the problem was that while The Play was an action-adventure style that described how Agile practices work, the reviewers were looking for more of a drama that described why Agile works the way it does – the principles behind it. We needed something to satisfy different audiences.

What we learned was that there were different perspectives of describing Agile, and if a conversation came from a presenter with one perspective to a listener with a different perspective, confusion, misunderstanding, and frustration would result.

Lesson Learned: Fit the message to the audience.

We defined three perspectives to be addressed in most of our presentations and discussions about Agile. Figure 1 shows the three perspectives: Principles, Practices, and Benefits.

In the upper right corner of Figure 1 are the principles – the things that describe WHY Agile is the way it is. This is the perspective that early adopters, evangelists, and strategists use. They want to understand the principles behind Agile so they can adapt it to their world. They don’t want to be constrained by detailed rules or specific implementations, and often don’t need proof of benefit.

At the bottom of Figure 1 are the practices – the things that describe HOW TO DO Agile. This is the perspective that practitioners use. They want to know how to make it work, but may not want to know why.

In the upper left corner of Figure 1 are the benefits – the things that describe why Agile is a good thing. This is the perspective that skeptics use. They want to know why Agile brings value to them or the organization, so they can decide whether it is worth the time to talk about principles or practices.

In most cases, any one person has some mixture of all three perspectives, and may switch from one to another depending on the topic of conversation. We found this model to be very helpful in focusing a discussion and identifying issues to be addressed.

Stages of Learning

When an organization is learning something new, it must recognize that the people in the organization have different learning styles, abilities, and motivations. The education of the organization must take these differences into account.

The Dreyfus Model of Skills Acquisition [1][2] gave us a good model for organizing our education activities. The model describes five stages of learning, giving us insight into the best way to communicate the benefits, principles, and practices of Agile.
• **Stage 1 – Beginner.** Operating at this stage, people want to know just enough to accomplish a task. They work well with specific rules, and are vulnerable to confusion. They do not want or are not ready to handle discussions of principles. Though they may have an element of skepticism, since they are missing important context they are not ready to accept information on benefits.

• **Stage 2 – Advanced Beginner.** Operating at this stage, people are still focused on the practices to accomplish tasks, but are beginning to understand some of the principles behind the rules. They need information quickly to help them expand their understanding. They want to understand the benefits to help them learn and apply the principles and practices.

• **Stage 3 – Competent.** Operating at this stage, people are using the principles to shape their use of the practices. They are able to see the long term, and recognize when things are working well and when they aren’t. They are conscious of the benefits perspective, wanting to connect principles and practices to the benefits.

• **Stage 4 – Proficient.** Operating at this stage, people are looking at the entire framework, looking for conceptual models to organize their thinking. They understand the inter-relationship of principles, practices, and benefits so well that they may not be able to separate them.

• **Stage 5 – Expert.** Operating at this stage, people work primarily from intuition, and having internalized principles so thoroughly, they may have trouble explaining their reasoning to others. They are continually looking for better practices, and are frustrated by an over-reliance on specific rules. They may see benefits as so obvious that they forget to address them in conversations with others.

**Lesson Learned: Understand How Teams Learn**

The most useful thing from this model is the recognition of how it affects team dynamics as a team is learning Agile methods. We observed:

• If everyone is at Stage 1 or 2, it is very difficult to learn quickly, because most of the learning is trial and error. While this learning may be effective, it is slow and can be frustrating. A group needs someone at Stage 3 or 4 to help them avoid some time-consuming or frustrating missteps to move up the learning curve more quickly.

• The corollary to the first observation is that it takes time to move through the stages. People starting at Stage 1 and 2 do not move up to Stage 3 simply because someone presents them with a set of principles. The lesson here is to allow time to learn practices and grow into an understanding of principles.

• Stage 1 learners are uncomfortable with Stage 5. Partly this is because a Stage 1 learner does not understand the things that a Stage 5 learner takes for granted, so communication is difficult. Partly this is because a Stage 1 learner may see the Stage 5 learner as being too theoretical, too far away from the practical realities of the world. The lesson here is for everyone to recognize what level they are in, and in conversations with people at different levels, establish a common ground for communication and carefully manage discussions that cause tension.

• When people disagree about benefits, there is a tendency to assume the other is at a lower stage of learning. Each side of a disagreement assumes the other doesn’t understand the issue enough, thinking: “If you understood this better, you would agree with me.” The lesson here is to step back from a disagreement to establish understanding, and then move back to the resolution of the conflict.

**Lesson Learned: We learn by doing**

There are many ways to move through the stages of learning, and there are many unique learning styles within a large organization. Some people learn well from books, some learn from classroom-style discussions, and some learn from watching others, but the best learning comes from doing. By applying Agile practices, by doing the software development using Agile techniques, we learn what works and what doesn’t, we identify and eventually internalize principles, and we see for ourselves the benefits of what we are doing.
3. Defining Principles

The Agile Manifesto [3] is an excellent summary of the high-level values that explain why Agile is the way it is.

It also contains some allergens that can easily cause a sensitive organization to have a bad reaction. If the last sentence is not applied carefully, it is easy to overreact to what the manifesto is saying about the “things on the left”. If the organization values the “things on the right”, regardless of the degree, the manifesto can be quickly dismissed as something incompatible with the organization’s values.

Even though the values in the manifesto rang true for our organization, they were still a little too abstract. To guide the evolution of the practices, we needed to adopt principles that were more specific and more aligned with the values of our organization.

In books and papers on Agile [4,5,6,7] there are many principles that describe why Agile works the way it does. The fact that there are so many demonstrates the point that the principles must be tailored to the organization using Agile. We chose a set of principles that aligned with other principles already well established in our organization. Some examples are:

- **Mission.** Our entire corporation is strongly connected to its mission statement and the principles that derive from it, so we emphasized Agile principles that supported that mission. For example, one element of the mission is, “To strive without reserve for the greatest possible reliability and quality in our products…” We emphasized the principle of Inspect and Adapt, and implemented the practices of Test-Driven Design, the Customer Role, and others to improve the quality of both our product and our process. Another element of the mission is, “To recognize the personal worth of employees…” We emphasized the principles of Maximize Transparency and Cultivate Collaboration, and implemented practices like the Planning Game, Team Reflections, and Co-location to empower and engage everyone on the team.

- **Lean.** Our product development organization has a Lean initiative, so we emphasized the principles of Eliminating Waste and Inspect and Adapt, and implemented practices like the Customer Role, Stories, Sprint Planning and Release Planning to put focus on value-added activities.

- **Commitment and Empowerment.** Our organization values individual and team empowerment, so we emphasized Openness, Visibility, and Empowerment as driving principles.

**Manifesto for Agile Software Development**

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and Interactions** over Processes and Tools
- **Working Software** over Comprehensive Documentation
- **Customer Collaboration** over Contract Negotiation
- **Responding to Change** over Following a Plan

That is, while there is value in the items on the right, we value the items on the left more.

4. Establishing Practices

Agile is filled with a rich set of practices that make it work. From the big-picture practices of planning to the detailed conventions for the content of a 3x5 story card, the practices need to be established and evolved.

There are many dialects of Agile, often with variations within them. For an organization that is new to Agile, it can be hard to know where to start, and harder still to know when you are doing it right.

**Lesson Learned: There Is No Recipe**

After struggling at “Stage 1 and 2” where we wanted a recipe for “How to Apply Agile Practices”, we accepted that there was no single, all-encompassing recipe. Instead, we established expectations that the team owns the practices and has the obligation to adjust them as we learn more. Rather than teach ourselves a bunch of recipes, we taught ourselves how to cook.

**Lesson Learned: Establish a Meal Plan**

Continuing the cooking analogy, we recognized the need to have a meal plan that connected the recipes and cooking skills needed to make Agile work. The power of Agile is the synergy of the practices that support the principles, so it isn’t easy to simply pick out a single
Practice and say, “Now we are Agile.” On the other hand, it is also not easy to become proficient at all of the practices at once.

We built our Agile model from a combination of XP and Scrum. We created a description of this model (“The Play”, mentioned above) that described all of the practices we wanted. We accepted that some of them were easier to implement than others, and that some of them would need a lot of work before we would see value. This helped focus our process-improvement efforts, and helped avoid frustration of not instantly being as good as we wanted to be.

Lesson Learned: Keep it Fresh

Stretching the cooking analogy a bit further, we recognized the need to ensure our practices stayed fresh and useful, and not become stale and distasteful. We emphasized the principle of Inspect and Adapt and practices like Team Reflections to constantly improve practices.

5. Demonstrating Benefits

The benefits of Agile can be very subjective. Conferences and articles are filled with claims of obvious benefits, but despite their vigor, the claims are often subjective, and therefore easily argued. Even claims backed up by objective data can be argued as being “not relevant to MY world.” For an organization to change, a benefit must be demonstrated.

We established benefits in three ways: connecting to what we already value, asking people, and gathering objective data.

Lesson Learned: Connect Benefit to What You Already Value

We connected the principles and practices of Agile to things the organization already determined to be beneficial. Our organization has various initiatives for improving the way we work together such as customer-focused quality initiatives aimed at improving quality, project management initiatives aimed at improving predictability, and process improvement initiatives aimed at improving productivity. Simply showing how Agile supports those initiatives is a way to demonstrate the benefits of Agile.

Lesson Learned: Ask People

We sought out subjective benefit measures primarily by asking the development teams. We created a survey to gather opinions about the impact of Agile on development work. The survey asks about such things as quality, speed, visibility of progress, empowerment, enjoyment of work, etc., with two questions: how good we are at that thing, and to what degree does Agile help us get better at that thing. We have taken the survey many times with many teams to get insight into how the benefits change over time as our use of Agile evolves.

Several good things have come from these surveys:

- We were able to take pride in the benefits we were gaining from the effort we were putting into our evolution of Agile.
- We were able to identify benefits that were “believed” to be true, and from there identify additional activities to objectively quantify our belief.
- We were able to identify disconnects between the benefits we wanted and the practices that delivered them. For example, when the survey showed the “visibility” benefit was not as strong as we liked, we stepped up our efforts to make the backlog and burndown mechanisms more visible and understandable.

Lesson Learned: Objective Data Supports Subjective Data

We investigated ways to get objective data to measure benefit. This is particularly difficult as measures of productivity, predictability, and quality are hotly debated, but we are seeing trends in our data that support the subjective opinions we gathered from the surveys.

We continue to look at data coming from systems we already have in place. We see changes in behavior demonstrated by the data that come from our change-management system, using data we have collected for years. It didn’t take much effort to describe the change as something valuable, because we know what the data means to us.

6. Words

Specific words can either add clarity or cause confusion. Depending on the degree of change from the old way to a new way, there may be a need for significant effort to establish new terminology.

Lesson Learned: Choose Meaningful Terms and Use Them Consistently

Even simple word choices can be important. For example, the terms “Sprint”, “Increment”, “Iteration”,
and “Release” were often used in overlapping, competing, inconsistent ways. We were very conscious of the words used in presentation and conversations, and eventually, through constant practice, reduced the confusion.

7. Communicate

All of this so far has been a description of the way we defined and organized the message – but how is that message delivered? The simple answer is whenever, however, and with whomever you can.

We have company-wide presentations where we presented various elements of the framework, often using the perspectives picture shown in Figure 1, above, to set expectations for what we would cover.

We have large project team discussions, some aimed at training and some aimed at problem solving or decision making, to help everyone learn Agile principles and practices as we go.

Small project teams hold regular reflections aimed at refining the practices, usually using the principles to guide the adjustments. Other teams work on the benefits perspective, creating the surveys analyzing the results to make recommendations for improvements in the practices and application of the principles.

All of this helps us to reinforce and recalibrate the message. Especially in the early stages of adopting Agile concepts, when there are disagreements and arguments about specific issues, this communication framework helped us isolate the relevant parts of the argument, and therefore guide us to a solution.

8. Conclusion

As with any change to culture and process, the investigation and adoption of Agile Software Development counts on effective communication. In Medtronic’s experience with Agile, we found value in:

- Identifying the barriers to understanding, as well as the barriers to implementation, and address them directly.
- Establishing a Framework for discussion and decision-making. That framework addresses:
  - Principles – Why is Agile the way it is, what will guide us.
  - Practices – How does Agile work, how will we implement it.
  - Benefits – Why is Agile good for us, how will we know we are doing better.
- Using specific terms and concepts that increase understanding and avoid confusion.
- Communicating constantly.

As our use of Agile continues to evolve, we use this communication framework to focus on the things that will provide us the most benefit to improve. We ask ourselves:

- Are we adequately addressing each perspective?
- Are we seeing tangible benefits? Are we able to demonstrate them, even to people who may be skeptical?
- Are the benefits aligned with things our larger organization needs from us?
- Are the practices working? Are we adapting them to work better?
- Are the choices we make in our practices supporting the principles we value?

With Agile, like any process improvement activity, there is a lot of work in defining new processes and implementing them properly. There is often just as much effort, and benefit, in establishing the means to communicate about that work.

9. Acknowledgments

Many thanks go to my fellow Agilistas for contributing both to this paper and to the journey it represents: Jon Spence, Pam Dagoberg, Marilyn Lamoreux, Susan Johnson, Pete TerMaat, Gary Teskey, Dave Holmstrom, Matt Plante, and Darren McDonald. Special thanks go to Jon Spence and Rebecca Wirfs-Brock for their tremendous help in getting this story written into words.

10. References