

## **Fear of Agile: 7 Guiding Principles to Make Agile more Agile**

By Jerry Manas, PMP

Much has been made of the advantages of Agile Project Management, yet many organizations are slow to embrace it. Part of the reason is that they misinterpret “agile” as “winging it.” Another reason is that they feel that Agile PM is for software development houses not catering to specific customer requirements. A typical comment is, “Maybe a software shop can afford to let the scope evolve, but we need full specific requirements from our customers right up front in order to know how to estimate or bill properly. We can’t afford to guess at the outcome.”

Perhaps Agile PM is misunderstood. Or maybe it needs to be made more... shall we say, *agile*, so that organizations will accept it more broadly.

To start with, let’s recap the main characteristics of Agile PM:

- Piecemeal iterations with fixed time and cost, but evolving scope as learnings emerge
- Rapid and ongoing delivery of value (i.e. working software) throughout the project
- Close and frequent collaboration between developers and customers to minimize misunderstandings
- Developers are trusted to deliver to customer needs; they are told what is needed, not how to accomplish it

Collectively, these characteristics serve three purposes:

- To minimize the impacts of uncertainty
- To create a more energetic and engaged team
- To speed up the delivery of value

It is difficult to debate these benefits. However, there are still the fears of management regarding the need to think through the detailed requirements up front, so that everything works on paper before it needs to work electronically. It becomes a philosophical debate about how much is knowable up front, and about what’s more efficient—getting it right on paper first or trusting the developer to provide a fairly decent iteration that leaves room for the joint creativity of the developer and customer to finish the job through multiple iterations. Which results in less waste?

The former (getting it right on paper first) is more comforting to management. The latter (evolutionary co-creation of the result) is often more comforting to the customer and the developer---and can arguably result in faster and higher value, provided

the objectives are clear. Another big concern of management is, “How do we know what *done* looks like?” They worry that Agile PM is another word for sanctioned scope creep. Finally, managers worry about having developers and customers in the same conversation, let alone the same room, as the two parties often don’t see eye to eye. Let’s face it, in many cases, developers are from Mars and business customers are from Venus.

One way to address the fears of management and to possibly make Agile PM more palatable to the masses is to adopt and publicize the following 7 principles.

### **1. One Size Does Not Fit All**

Clearly there should be differences in approach depending on whether software is being developed for the general public or if it’s being developed to solve a specific customer need. With the former, it can be quite appropriate to take a “just do something fast and let it evolve” approach as long as it is in line with general objectives. When trying to meet a complex customer request, however, it could lead to a lot of wasted time and effort if details are not examined early. I’ve seen far too many projects having to undergo rewrites because something in scope for a future iteration wasn’t examined early enough.

Likewise, one would approach a project with great uncertainty different that they would where the specifics can be clearly defined up front. The key is to make the strategy fit the situation. One size does not fit all.

### **2. Symptoms before Solutions**

A wise friend once told me “empathy before strategy.” I always remember those words. So often, people (men, in particular) are so quick to jump into strategy before fully understanding and empathizing with someone’s plight. If a child falls, the father begins to admonish the child about being more careful before addressing the child’s immediate needs. He may not even understand how the child fell to begin with. Men often jump too quickly into solution mode. Physicians have it right though. They are trained to seek to understand the *symptoms* before they examine causes and cures. And the good ones will discuss the patient’s lifestyle as well so as to build rapport and understanding, and to apply a more holistic treatment.

It is the same with project management. We must first seek to understand the business problem or opportunity before we ask what is needed or desired. Often what is desired doesn’t address the business problem, and nobody realizes it because they never thought to ask what the business problem is to begin with. This speaks to the lean manufacturing principle: *Decide as late as possible*. We first need to probe into the symptoms and ask plenty of “why,” “who,” and “what-if” questions. We need to probe about desired outcomes. Then we can begin to think about causes and solutions. If we do this, then our software iterations will be that much more on the mark.

### **3. Seeing is Understanding**

The Japanese lean manufacturing philosophy is “go and see for yourself.” They know that so much waste is due to misinterpretations, miscommunication, and false assumptions. This can be avoided by simply seeing the problem in action, understanding the as-is situation, and getting a good feel for the desired to-be state. Certainly use case documents help with this. But there is nothing like doing a physical walk-through to see the issue in person. If that cannot be arranged, then at least talk to the people who are experiencing the problem and do a mental walk-through of the current state, noting where the issues are.

By broadening and deepening our understanding of the problem, we can be much better equipped to address the real issue and not waste time on extraneous functionality. This is time well spent, and can lead to asking better questions and providing more valuable solutions. It can also decrease the number of software iterations and scope changes, thus pacifying managers afraid of scope creep.

### **4. Consider the Whole System**

In our quest for the perfect solution, we often overlook related needs outside of the direct programs being developed. This includes impacts on business processes, historical data, training needs (for both the end user and support personnel), interfaces, upstream and downstream systems, potential reports, ongoing support, configuration/setup, and more. It’s rare to have a program that is an island unto itself, and often a new program brings unexpected collateral damage.

Management often equates Agile PM with “software geeks” who are solely focused on technical details. A concerted effort toward broadening the analysis to include “the whole system” can go a long way to remove that image---and the fear it conjures.

### **5. Watch Your Assumptions**

Henry Ford said, “If I had asked my customers what they wanted, they would have said a faster horse.” Often, a customer asks for something because they think or assume that it’s the only way to accomplish the task, based on their own sphere of knowledge. And we march along delivering what they asked but not effectively solving the problem. This is but one type of assumption that’s typically made in project requests, and can often be remedied by focusing on the symptoms and letting the software experts propose solutions. Other times, assumptions are made when a customer describes a symptom or tells us some fact about data. We assume what they’re telling us is true, but sometimes it is not.

The popular phrase, “Trust, but verify” comes to mind. Many software errors happen because erroneous assumptions were made and not followed up on. As a result, rework must be done. Even in iterative software development, details matter. This is where it pays to have a good business analyst (I’ve often thought of the word *analyst* to mean “one who is anal,” but that’s often what it takes when it comes to validating customer requirements). Management often thinks that Agile PM means not validating details up front, but that doesn’t have to be the case. That’s a faulty assumption on their part, so we must educate them.

## **6. Expect Change, but Manage It**

One of the biggest misconceptions about Agile PM is that change is not managed. It’s true that Agile PM expects and welcomes changes to requirements, even late in the game, as it can result in higher value to the customer. It’s also true that Agile PM assumes fixed time and cost iterations, so the customer must be willing to make tradeoffs in requirements to stay within the current iteration’s time and cost limit. If the customer must have the additional features in the current iteration (i.e. they are of extremely high value or mandatory), then a change request is needed to add the necessary time and cost. This is usually escalated to senior management. Otherwise, tradeoffs are made and some things are pushed to the next iteration (this can be managed by the project team). Minor changes are just handled and absorbed.

This is truly a flexible and efficient model that manages change quite well, especially combined with better up front analysis (as proposed in our earlier principles). In essence, the focus of Agile PM is always to concentrate on the highest value items in the current iteration, and stay within the time and cost limits. In this way, value is delivered in an ongoing manner, much like time-released medicine. This isn’t chaos; it’s good customer service and a more efficient process.

## **7. Bridge the Culture Gap**

Management is often taken aback when they hear that Agile PM involves heavy interaction between developers and customers. And they have a point. Some software developers (not all) are meant to be coding behind the scenes and have difficulty communicating clearly to a customer. In extreme cases, a developer can appear abrasive to the customer---this is especially damaging with external customers. In such cases, it is not unwise to have an intermediary, either a business analyst or a customer liaison to help bridge those gaps.

In some organizations, the project manager wears multiple hats and also serves as the business analyst (though I’d argue that different skill sets are involved and it can shortchange the project manager’s ability to facilitate – but I digress). Education can also help developers understand the nuances of client relations, though in extreme cases it will have minimal impact. Each case is different, so we need to adopt the appropriate strategy-

---education or an intermediary. In any case, this culture gap often exists and we need to be aware of it.

## Summary

The above 7 principles, if publicized and adopted, can go a long way toward appeasing the fears of management, and making Agile PM more accepted. It can also boost the natural benefits of Agile PM---greater risk reduction, higher engagement, and faster value delivery. And it's hard to argue with that.

Jerry Manas is President of the Marengo Group, author of *Managing the Gray Areas* (RMC Publications, January 2008) and *Napoleon on Project Management* (Nelson Business, April 2006), and co-author of *42 Rules for Creating WE* (Superstar Press, August 2009). Through the Marengo Group, Jerry helps project and virtual teams achieve high performance using techniques and practices that result in greater alignment, leaner processes, and more strategic use of technology. Jerry is a founding member of *The Creating We Institute* ([www.creatingweinstitute.com](http://www.creatingweinstitute.com)) and co-founder of *PMThink!* ([www.pmthink.com](http://www.pmthink.com)), a popular project management blog site. Visit his website at [www.marengogroup.com](http://www.marengogroup.com).