# Position Paper for OOPSLA 2003 Workshop #2: Are Agile Methodologies Really Different?

Steve Berczuk
Aliaswire, Inc – www.aliaswire.com
steve(at)berczuk.com/www.berczuk.com

My primary interest in this workshop is that I am moderating a panel *What's so eXtreme About Doing Things Right?* (http://oopsla.acm.org/oopsla2003/files/pan-9.html) on Thursday Oct 30 with a similar theme. My tendency is to wonder not what is different between the agile methods, but rather, what the difference is between using an agile method & just following "good" practice. Rather than asking "are agile methodologies really different (from each other)?" I wonder "are agile methodologies really different (from good practice)?"

On any project you are faced with the problem of how to deliver a certain amount of functionality to a customer with a certain level of quality given the constraints of a project (time, money, resources, skill levels of the implementation team, etc). Defining *quality* as "value to some person" (Gerry Weinberg's QSM Vol-1, p 7), your process should also help you and your customer to discover what the software needs to do to create value. To do this, the process should foster communication, and manage the customer's expectations. The best way to foster communication and manage expectations is refine the project based on frequent feedback. Using an agile process just makes explicit a management model that relies on these things. I suspect that most management models rely on these things as well; they are just not explicitly enforced.

All agile methods have techniques to assist with all of these aspects of good project management. If there are any differences, they are along the dimensions of *team and project size*, *time scale* (daily vs. project), and *organizational level* at which the practices are defined (developer vs. manager).

For example:

- XP practices are geared towards the level of the development team.
- Scrum practices are oriented more towards management. Scrum encourages you to organize and prioritize so the useful functionality gets delivered at a reliable rate. XP-like practices can help you to get there, but Scrum does not demand them.
- Lean Software Development works at the team and management level so that you can deliver useful software while delaying decisions as long as possible, thus allowing for feature evolution. Lean principles guide the developer activities and planning practices, but the details are up to the implementer.
- The Crystal methods are about communication, and prerequisite for any agile (or indeed, effective) development process.

These methods are all about feedback, prioritization, and value-delivery within a time constraint. The methods are complimentary; each method provides detailed guidance at various levels, but every project needs to work at *every* level. Al methods assume some basic level of "process" (for example, no agile method can work unless the team uses an appropriate SCM process). An agile method gives one a framework, but every framework has gaps that projects need to fill in using their skills and knowledge. The methods assume that the teams will follow appropriate good practices appropriate with the values of the method.

All methods are, in the end, about managing uncertainty. All projects are about delivering functionality with the provided constraints. Agile methods have much to teach all of us about how to deliver good software on an appropriate schedule. The fact that we have "methods" is an artifact of people taking comfort in a packaged solution.