Agile Methods, the Emersonian Worldview, and the Dance of Agency

Brian Marick
marick@testing.com

I was fortunate to be at the Snowbird workshop that led to the *Manifesto for Agile Software Development*. That manifesto has two parts: a statement of values and a statement of principles. The statement of values came easily. The statement of principles was a struggle. It was hard to reach consensus on some of them. One difficult principle was this one:

> The best architectures, requirements, and designs
> emerge from self-organizing teams.

The words "emerge" and "self-organizing" evoke the literature of Complex Adaptive Systems and Complexity Science. Indeed, the independent writings of many Manifesto authors are much more explicit. For example, Jim Highsmith's *Adaptive Software Development* has 31 index entries for "complex adaptive systems," and Schwaber and Beedle's *Agile Software Development with Scrum* has a section titled "The Complexity Science View of Scrum."

But some Manifesto authors were less comfortable with the implied metaphor AGILE PROJECTS ARE COMPLEX ADAPTIVE SYSTEMS, so the principle as published is something of a compromise. I've long had a feeling that there's some other principle struggling to get out from behind it – some other, more general metaphor that would better capture a consensus.

I want to be very clear here. I am in no way taking a position on the relevance of the complexity literature to agile methods. I'm not qualified to have an opinion, so I defer to those who are. My questions here are why metaphors of emergence should be popular among so many agilists, why they make others of them uncomfortable, and whether a different metaphor could better crystallize what agile methods have in common.

This summer, I may have found that metaphor. I hope this workshop will tell me if I did.

My idea is that people attracted to agile projects share a worldview quite like the one Cornel West attributes to Ralph Waldo Emerson in his (West's) *The American Evasion of Philosophy*.

1. Emerson held that "the basic nature of things, the fundamental way the world is, is itself incomplete and in flux" (p. 15). Moreover, the world and humans are bound up together: the world is the result of the work of people, and it actively solicits "the experimental makings, workings, and doings of human beings" (p. 15).
2. Emerson believed that this basic nature makes the world joyous. It gives people an opportunity to exercise their native powers with success, because the world is fundamentally supportive of human striving.

3. And finally, Emerson believed that human powers haven't yet been fully unleashed, but they could be through the "genius of individuals willing to rely on and trust themselves" (p. 16).

Let's look at a familiar agile claim in the light of this Emersonian worldview.

> **Claim:** Good global design can emerge from constant refactoring that is driven by micro-design rules such as "eliminate duplicate code" and "minimize the number of classes and methods."

That sounds like familiar accounts of emergence, the stories we hear about flocks of birds, termite nests, and ant trails. Once more, pleasing global structure emerges from simple rules with local scope.

But I think there's something more going on. It is not *just* that blind local rules produce surprising global effects – it's that the nature of software both solicits change and is predisposed to make that change successful. Software *can* be soft, given the right techniques – discovered through human striving – and proper trust.

In the lingo of philosophers, it's an account that gives both the working software and the people working on it *agency* (the ability to act, even the ability to intend). For an illustration of this attitude, see Kent Beck's *Smalltalk Best Practice Patterns*. On page 3, he says, in a recounted conversation, "Since the code seems to be telling us to do this, let's try it." Many programmers have the feeling that the code "wants" to be a certain way, or "resists" certain changes. I speculate that an agile programmer is one who takes that feeling seriously, who thinks of programming as what Andrew Pickering calls "the dance of agency" in his *The Mangle of Practice*. It's a dance in which the human and the working code are partners, responding to each other's lead.

This notion of a dance of agency is simultaneously soothing and weird. It's soothing because it gives humans a role beyond being a mechanism for applying local rules. The "best architectures … emerge" principle suggests a lack of human agency, and it was this that some of the Snowbird authors seemed to be objecting to. It's weird because it's not in keeping with the determinedly mechanistic rhetoric of our time. (The world soliciting humans? Come on!) And it's not in keeping with the rhetoric of software engineering, which makes flux regrettable, human limitations fundamental, and a search for stable truths more important than finding new things to do.

No, it's definitely too weird to say that the principle hidden behind the emergence principle is this one:

> Agile projects are Emersonian optimists, engaged in a dance of agency with their working code and the changing business world.

But you know… I think they really are.