

Post-Agilism – Beyond the Shock of the New

Jason Gorman

Starting in the late 19th century, the world was shaken to its core. In art, literature, music, science and philosophy, ideas we had come to accept as gospel truth (quite literally) were suddenly turned on their head by radical new thinking.

Modernism was a tidal wave of uncertainty and unpredictability that swept away hundreds of years of steady, linear progress after Newton had shown us that the world was essentially clockwork and predictable.

In the twilight of the 19th century, Monet, Renoir, Sisley, Pissarro, Cézanne and Bazille joined together in Paris to form the loose alliance we now call *impressionism*. And, beyond them, composers like Ravel and Debussy, and writers such as Baudelaire and Verlaine, took up the impressionist cause and incorporated *impressionist values* into their work. More and more people jumped on to the impressionist bandwagon, of course. And not all of them were, strictly speaking, impressionists. In its origins, impressionism was the culmination of a long process of evolution in styles and techniques, born of very talented, and – more importantly – highly trained and disciplined artists. As the popularity of impressionism increased, the movement became a gravy train. And those that jumped on to the gravy train much further down the line often lacked the training and the discipline that had brought impressionism into being in the first place. By 1910, the term “impressionist” had almost completely lost its meaning.

Software development has had its own little modernist revolution: in the form of the **Agile** movement. At the dawn of the 21st century, Kent Beck, Martin Fowler, Ron Jeffries, Scott Ambler, Robert Martin, Ward Cunningham and others joined together at Snowbird ski resort in the Wasatch mountains of Utah to form what we now call the *Agile Alliance*. And to the millions of software professionals who had spent decades working in an industrial, Newtonian paradigm for *software engineering*, their ideas were every bit as shocking.

According to the Agilists, you didn't need a detailed long-term project plan. You didn't need a detailed up-front design. You didn't need a complete set of well-defined requirements to begin coding. Hell, you didn't even need a sophisticated UML modeling tool or a high-priced requirements management solution – you could just do doodles on scraps of paper and write your requirements down on bits of card! It flew in the face of everything many of us had been taught about *software engineering*. Surely it was something we should do wearing white coats using very expensive equipment, like proper engineers?

Agile Software Development was truly shocking, and the fur flew in IT establishments across the globe whenever the ‘A’ word was dared to be uttered by the trailblazers who sought to adopt these radical new ideas on their own projects.

5 years later, and the dust is just starting to settle. Like impressionism, Agile Software Development was born out of talent, training and discipline. And just like impressionism, after a while it has become a gravy train. The term “Agile” is becoming more mainstream. More and more job advertisements ask for “Agile experience”, and even consumer computing magazines have started to refer to “Agile Software Development” as if your average laptop user should know what that means. We see the term transplanted in front of a wide variety of products and services – and it appears in millions upon millions of mission statements and glossy sales brochures. And there is no doubting that those who are jumping on the Agile bandwagon today often lack the training and discipline of the Agile founders. **Agile is losing its meaning.**

So what happens next? When we’re eating Agile cereal and wearing Agile shoes and listening to Agile synth pop on our Agile radios, what will it mean to be an “Agile developer”?

First we need to separate the word from the idea. Think of music. One of the earliest of the modern composers, Eric Satie, experimented with rich harmonies – 7th chords, 9th chords, 13th chords and so on. These rich harmonies were heavily influential on later composers like Ravel and Debussy, who – in turn – influenced even more modern and shocking composers like Stravinsky and Schoenberg. We can trace these rich harmonies down through more than 100 years of music – they are the musical equivalent of a barium meal, highlighting the path of these radical ideas as they circulated through the veins and arteries of the musical world. They run like seams of gold through a million hours of original compositions in dozens of different musical styles. These big harmonies are like the DNA passed down from the earliest micro-organisms, which we still carry today.

Impressionism didn’t end, it just *evolved*: passed down through generations of new intellectual and artistic styles, it is alive and well today. We just don’t call ourselves “impressionists” any more.

And we can already see this process going on in Agile Software Development. Successful “Agile DNA” is being incorporated into new generations of development methodology, along with DNA from nearly 6 decades of software development.

Those of us who consider ourselves **Post-Agilists** have taken what worked and cross-bred it with the best bits of dozens of other approaches and disciplines, creating new variants that have the potential to be even more exciting, daring and shocking.

The Agile movement has successfully challenged the existing order and shaken the software industry out of a potential rut, bogged down by outmoded 19th century industrial thinking and “big process” dogma. It has opened the door to a very wide range of possibilities, and is now the catalyst for a Cambrian explosion of new ideas on how to deliver software and systems with bizarre, exotic-sounding names like *Pliant Programming* and *Nonlinear Management*.

The Agile gravy train hasn’t run its course yet, and will no doubt be a money-spinner for a few years to come. But for those of us looking for new adventures beyond broader horizons, these are very exciting times indeed. Anything is possible!

© Jason Gorman, 2006.

Jason Gorman is an independent consultant, author, trainer and coach based in London, England. He has 14 years' experience working across a broad range of industries, applications and technologies, and specializes in business and systems architecture, agile methods, and the unification of corporate and IT strategy through simple, lightweight and rigorous techniques. His web site, parlezuml.com, is currently the most popular UML training resource on the Internet, according to alexa.com, and his tutorials have been used by over 180,000 people since 2003.