Flaccid Scrum

By Ken Schwaber, 5/1/09

Comments are arising about flaccid Scrum. Flaccidity is usually not attributed with the word “good” or “desirable” so this attracted my attention.

When we wrote the Agile Manifesto in 2001, we intended to provide an alternative to waterfall and the emergent new heavy processes such as RUP. An alternative certainly was needed, since the project failure rate was hovering near 50% and our customers were willing to hire people half-way around the world in hopes that they could do better than their local developers.

Scrum has been a very widely adopted Agile process, used for managing such complex work as systems development and development of product releases. When waterfall is no longer in place, however, a lot of long standing habits and dysfunctions have come to light. This is particularly true with Scrum, because transparency is emphasized in Scrum projects.

Some of the dysfunctions are those that Martin Fowler (see first framed box at end of this article) comments on, specifically poor quality product and completely inadequate development practices and infrastructure. These arose because the effects of them couldn’t be seen very clearly in a waterfall project. In a Scrum project, the impact of poor quality caused by inadequate practices and tooling are seen in every Sprint.

A ScrumMaster is responsible for teaching others how to use Scrum. If it were as simple as an explanation, the work would already be done. However, we come from over thirty years of waterfall development, and its practices are in our muscle memory. To change these practices will likely be as difficult as for a free spending corporate giant adopting lean thinking.

The primary habits that hinder us are flaccid developers and flaccid customers who believe in magic, as in:

- **Unskilled developers** - most developers working in a team are unable to build an increment of product within an iteration. They are unfamiliar with modern engineering and quality practices, and they don’t have an infrastructure supportive of these practices.

Ignorant customer - most customers are still used to throwing a book of requirements over the wall to development and wait for the slips to start occurring, all the time adding the inevitable and unavoidable changes. Belief in magic - most customers and managers still believe that if they want something badly enough and pressure developers enough to do it, that it will happen. They don't understand that the pressure valve is quality and long term product sustainability and viability.

Scrum makes these habits and the consequences of them transparent. My belief was that the transparency would be met with determined effort to change, to improve, and to make ours the great profession that it started to be and could become again. Imagine my surprise when, instead of improvement, many users of Scrum instead "customized" or "modified" Scrum so that these habits and their consequences would no longer be visible. This reinstates all the flaws of waterfall, except with far more damage since they are realized every iteration rather than just once per project.

As Martin points out, Scrum doesn't address product management, development practices, quality practices, or even requirements management. It simply points it out when they are inadequate.

We have started working with communities that do address these practices. We are working with them to train their professionals how to make their practices work within the Scrum framework. With software developers, this is pretty straightforward, because Extreme Programming is directly applicable. In project management, we are working with PMI and IPMA to incorporate Scrum practices. We are also working with product management professional groups for them to understand and incorporate Scrum practices.

Over the next year, you will start to see certification programs, such as a Certified Scrum Developer program and a Certified Scrum Product Owner program. These programs will teach people how to competently perform the work of their profession within the Scrum framework. For instance, the Certified Scrum Developer program will teach development teams how to select work and turn it into a potentially shippable increment of functionality within one iteration, at a quality fit for the purpose of the product. This training will be using various best practices and development technologies.

Now everyone is saying, "Oh, no, not another certification after all the problems with the Certified ScrumMaster program." They are only saying this because they believe that certification creates perfection. Given the muscle memory of bad habits in our profession, the best that certification will provide is acknowledgement that the

participants have been trained as well as possible in the Scrum approach and that they have intellectually apprehended these new techniques, thinking, and approaches. Only informed, coached, supported practice will make these changes stick, though, and no certification can create change.

I have been inordinately pleased with the Certified ScrumMaster program and certification. As a direct result of the program, waterfall is in retreat and Scrum and Agile are in ascendance. The problems in our profession have been exposed, can be quantified, and now can be remedied. The Certified Scrum Trainers (who provide the training in Scrum) aren't perfect, and many of their students only imperfectly understand and practice Scrum. Many are so appalled by what Scrum exposes in their organizations that they modify Scrum instead of fixing the problems.
Yet, they have a vision of improving our profession, or a better way of working, and they have persisted. Congratulations to them.

We have exposed a problem, and we have a tool - Scrum - that will aid us in understanding and tracking our improvement in our profession. It will be a long hard road. Bit by bit, though, we are starting to enjoy our profession again and our customers are beginning to trust us and the products they can make with us. Time is on our side.

Mark Paulk recently asked me what I thought of those who were changing Scrum. The second framed box has my response.

Best wishes to those who are joining us on our journey to improve our profession. I look forward to a long, hard, fruitful, fulfilling endeavor.

Martin Fowler's Comments:
There's a mess I've heard about with quite a few projects recently. It works out like this:

They want to use an agile process, and pick Scrum.

They adopt the Scrum practices, and maybe even the principles.

After a while progress is slow because the code base is a mess.

The scrum community needs to redouble its efforts to ensure that people understand the importance of strong technical practices.

January 29 of this year, Martin Fowler, posted a comment about flaccid Scrum (HYPERLINK "http://www.martinfowler.com/bliki/FlaccidScrum.html" http://www.martinfowler.com/bliki/FlaccidScrum.html)

My response to Mark Paulk's question:

Scrum is a tool, a framework, which can be used to build complex products. It does not prescribe any of the common engineering, people, risk management, or other practices. For instance, it doesn't say the team has to be collocated.

What Scrum does provide is feedback so that someone using Scrum can improve the results. For instance, if someone wants productivity and quality and can have a collocated team, Scrum will point this out. If the person starts with a dispersed team and compares its productivity to another collocated team, conclusions can be reached. An intelligent person would then change (continuous process improvement).

So using Scrum correctly means following all of its rules, which expose everything (transparently) for inspection and adaptation.
An intelligent person would then inspect what Scrum is making transparent and make changes to optimize the results. Presumably, the changes are cost justified.
Someone can use Scrum perfectly and ignore what is made transparent.
Someone can use Scrum imperfectly and act on some of the things that have been made transparent.
Someone who uses Scrum perfectly and acts more intelligently than anyone else on what has been made transparent will out compete anyone else.

flac·cid

Pronunciation:\'fla-səd also 'flak-səd\

Function: adjective

1 a: not firm or stiff; also: lacking normal or youthful firmness 〈flaccid muscles〉 bof a plant part: deficient in turgor

2: lacking vigor or force 〈flaccid leadership〉