

Perspectives on Scrum

By Allan Kelly, allan@allankelly.net, <http://www.allankelly.net>

Scrum, Scrum and Scrum... One of the great advantages of Scrum is that it is a lot better defined than Agile. Several texts carry the definitive version of what Scrum is (e.g. Deemer et al. 2008; Schwaber and Beedle 2002; Schwaber 2004; Sutherland and Schwaber 2011) However over the years this advantage has become diluted. I have come to believe there are three different meanings of the term “Scrum”. I find it useful to differentiate between these three terms: Hard Core Scrum, Common Scrum and Scrum as a synonym.

Scrum as a synonym of Agile

“Scrum” means “Agile” like “Hoover” means “Vaccum Cleaner” in English English, or like “QTips” means “Cotton Bugs” in American English, or increasingly “Google” means “Internet Search” whether you are using Google, Bing, Yahoo or some other search engine.

This is the way I believe a lot of software people use the term Scrum. Although Agile is more than Scrum alone often (perhaps even mostly) when people say Scrum they mean some a general form of Agile, something with iterations, stand-up meetings, perhaps User Stories, perhaps some technical practices, etc. etc.

Hard Core Scrum

This is Scrum without Project Manager or Architects, maybe even without Testers. This is Scrum of *Commitment* and *Abnormal Termination of Sprints*.

The *The Scrum Primer* (Deemer et al. 2008) says: “The Scrum Master is not the manager of the team or a project manager... there is no role of project manager in Scrum.” Some Scrum advocates go as far as to say: “If you have a project manager on a Scrum team you aren’t doing Scrum.”

This view sees Scrum as a kind of Communist Manifesto but with Managers as the Bourgeoisie. It would be funny if this weren’t so serious. I believe that if you take this interpretation you will soon run into opposition from a layer of managers. The paradox is that Scrum is as popular as it is because those manager often don’t like Extreme Programming and saw Scrum as the management friendly alternative.

(I sometimes jokingly call this meaning Scrum™ although in reality nobody owes the trademark “Scrum”.)

Common-Scrum (Scrum-Lite)

This is Scrum but without the anti-manager zeal. Of course for many people this strips Scrum of its primary asset, and they might be right. It might be that if you neuter Scrum you get a less effective result.

In this form of Scrum Project Managers still exist - assuming you had them to start with, a lot of places don't. However as Ken Schwaber says "The team leader, project leader or project manager often assume the Scrum Master role." (Schwaber and Beedle 2002)

In Common-Scrum the word commitment really means "agreement" because the team are probably using velocity to judge how much work they can do. Architects and other ranks, sorry, roles, still exists. And companies still kill teams - corporate psychiatry is rampant.

In other words: Scrum-Lite is full of compromises. It works for some companies, it doesn't work for others and Scrum™ purists will hate it.

References

Deemer, P., G. Benefield, C. Larman, and B. Vodde. 2008. "Scrum Primer." <http://www.scrumalliance.org/resources/339>.

Schwaber, K. 2004. *Agile Project Management with Scrum*. Microsoft Press.

Schwaber, K., and M. Beedle. 2002. *Agile Software Development with Scrum*. Addison-Wesley.

Sutherland, J., and K. Schwaber. 2011. "The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game." <http://www.scrum.org/Scrum-Guides>.

(c) Allan Kelly 2013 allan@allankelly.net

This essay is a work in progress. The author welcomes comments and feedback at the address above. April 2013

About the author

Allan Kelly has held just about every job in the software world, from system admin to development manager. Today he works as consultant, trainer and writer helping teams adopt and deepen Agile practices, and helping companies benefit from developing software. He specialises in working with software product companies and aligning products and processes with company strategy.

He is the author of two books "Business Patterns for Software Developers" and "Changing Software Development: Learning to be Agile", the originator of

Retrospective Dialogue Sheets (<http://www.dialoguesheets.com>), a regular conference speakers and frequent contributor to journals.

Allan lives in London and holds BSc and MBA degrees. More about Allan at <http://www.allankelly.net> and on Twitter as @allankellynet (<http://twitter.com/allankellynet>).