

Scrum

[Steve Denning](#), Contributor

RADICAL MANAGEMENT: Rethinking leadership and innovation

If there was a Nobel Prize for management, and if there was any justice in the world, I believe that the prize would be awarded, among others, to Jeff Sutherland, Ken Schwaber and Mike Cohn for their contributions to the invention of Scrum. Until recently, like most people, I had never heard of Scrum. This is not surprising, as it is rarely mentioned in general management textbooks or discussed in business schools. I came across Scrum several years ago, almost by accident.

My discovery of Scrum

At the time, I was working on a book focused on resolving an old management conundrum: how do you combine rapid innovation with disciplined execution? I was proceeding by asking people if they knew about any such workplaces where this was already happening.

I surprised to find that an unusually high proportion of the workplaces that I heard about were in software development. Initially I didn't pay it any attention. After all, these were geeks, and they talked in a strange, barely comprehensible vocabulary. What could I possibly learn about management from people who had, I imagined, gone into computing because they preferred machines to people?

A colleague, Hans Samios, a manager at Intergraph Corporation in Alabama, contacted me and suggested that I check out what was happening in software development firms, under the names of Agile and in particular, the practices known as Scrum. I had never heard of Scrum, but I decided to check it out.

The history of Scrum

I learned that the first full implementation of Scrum had occurred in 1993 when Jeff Sutherland along with John Scumniotales and Jeff McKenna implemented Scrum at the Easel Corporation.

They drew on the inspiration of the classic 1986 HBR article "The New New Product Development Game," where Takeuchi and Nonaka had compared a new holistic approach to innovation to the sport of rugby, where the whole team "tries to go to the distance as a unit, passing the ball back and forth". This paper in turn drew on a long experience of iterative methods.[\[1\]](#)

In 1995, Sutherland and Schwaber jointly presented a paper, "The SCRUM Development Process," at Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) Conference '95 in Austin, Texas, its first public appearance.

In 2001, Sutherland, Schwaber, and fifteen colleagues got together in Snowbird, Colorado, and drafted the [Agile Manifesto](#), which became a clarion call to software developers around the globe to pursue this radically different type of management.[\[2\]](#)

Since then, Sutherland, Schwaber, and their colleagues have gone on to generate thousands of high-performance teams in hundreds of companies all around the world under the labels of *Scrum* and *Agile*.

Other important contributions to the Scrum practices were made by Mike Cohn with the development of user stories as the principal tool for describing client-oriented goals of work, along with the development of story points as a way of measuring the quantity of work and the velocity of teams.

What are the practices of Scrum?

When I checked out what was going on in these companies, I could see that underneath the cover of an esoteric terminology, these software developers had discovered a solution to the problem of combining disciplined execution of high-level intellectual work with continuous innovation.

If you extract the practices of Scrum from the esoteric vocabulary in which it is expressed for software developers (“sprints”, “burndown charts”, “product owner”, “scrum-master”) it comprises in essence the following core practices:

1. Organize work in *short cycles*:
2. The management *doesn't interrupt* the team during a work cycle.
3. The team reports to *the client*, not the manager:
4. The team estimates *how much time* work will take:
5. The team decides *how much* work it can do in an iteration:
6. The team decides *how* to do the work in the iteration:
7. The team *measures its own performance*:
8. Define work goals *before* each cycle starts:
9. Define work goals through *user stories*:
10. Systematically *remove impediments*:

None of these practices is by itself new. What is new is doing all the practices together in a disciplined way of getting all work done.

When the practices are generalized in this way, beyond software development, they can be collectively described as [dynamic linking](#), to distinguish them from the traditional practices of hierarchical bureaucracy, where individuals reports to bosses to produce outputs.

Teams using the practices that Sutherland and his colleagues had pioneered have been unexpectedly productive. These were not just improvements where the teams were just *slightly* better than the norm. The best teams routinely obtain productivity increases of 200 to 400 percent, changes that are potentially *industry-disruptive* improvements.

Some mixed implementation results of Scrum

Nevertheless, despite the *enormous potential* that individual teams and departments have shown with Scrum, the overall picture of *implementation* has been quite mixed. More than 70% of Scrum implementations have failed to achieve their goals.

Most of these implementations with mixed results, which Sutherland derisively calls “Scrum-but”, are examples of a failure to implement the full array of Scrum practices. When only some of the practices are implemented, such as doing the work in short cycles but interrupting the team during the cycle, the potential gains in productivity don’t occur.

In part these problems of implementation have flowed from the way Scrum is sometimes viewed and introduced.

For instance, according to the Wikipedia, “Scrum is an iterative, incremental framework for project management often seen in agile software development, a type of software engineering.”

When you try to embed Scrum as a project management framework within a larger setting of a traditional management of hierarchical bureaucracy, there are inevitable tensions. Usually the prevailing culture of hierarchical bureaucracy is triumphant.

This experience sometimes leads people to believe that Scrum is just another management fad that didn’t work.

What the experience really shows is the opposite: that hierarchical bureaucracy is a work culture that no longer fits the marketplace of 2011. Scrum works. It’s the traditional management culture that doesn’t work.

The potential of Scrum as a different way of managing

The real question is: what can you accomplish if you execute Scrum well? The answer is now apparent with the results shown by Salesforce.com, which has been growing by 41% over a sustained period, and whose CEO, Marc Benioff, has been identified by Forbes as the most valuable CEO on the planet.

A turning point for Salesforce.com came in 2006, when the leadership realized that as the firm had grown, innovation in the firm had started to slow down. Instead of doing what most firms do, i.e. trying harder with more-of-the same kind of management, Salesforce.com adopted a *different* kind of management: Scrum.

That was the point of my article [Six Common Mistakes That Salesforce.com Didn’t Make](#).

Unlike many firms that have tried to implement Scrum, the leadership at Salesforce.com saw that Scrum involved not just the adoption of a new business process, or a framework for managing software development, but rather as a fundamental transformation of the way work was managed in the company. They realized that they were introducing a new way of thinking, speaking and acting in the workplace for both managers and workers. They committed to it boldly and the results have been extraordinary.

Application of Scrum beyond software

The success of software development at firms like Salesforce.com [CRM], along similar customer-driven iterative methods in auto manufacture at firms like Toyota, has led to the spread of this different way of managing to related fields.

- The Quality Software Engineering group at IBM [IBM] is responsible for software development processes and practices across the company. As part of the effort to promulgate Scrum in developing software, an iterative process of working was adopted for doing change management.
- At the Chicago software firm Total Attorneys, iterative work patterns were so successful that they spread to the staff of call centers: small cross-functional teams work in cycles of three weeks.
- At the Danish software firm, Systematic, iterative methods have been spreading from software development to other parts of the firm.
- At the Swedish software firm Trifork, iterative methods have spread from software development to conference management.
- And OpenView Venture Partners, a Boston-based venture capital firm, has expanded client-driven iterations into consulting and finance.

Once a firm sees the dramatic benefits of small client-driven iterations in one area, it becomes natural to ask: Why not do all work in this fashion?

What to call these radically different management practices? In manufacturing, they are known as Lean. In software development they are known as Scrum and Agile. When they are applied to management in general, none of those terms is really applicable, as they carry the baggage of their origins in software and manufacturing. The Agile Manifesto for instance dwells on the goal of work as “working software”, which is fine for software development, but not relevant to other sectors.

Scrum, Agile and Lean are in effect subsets of a radical shift in management more generally, or what I have called *radical management*.