Successfully implementing radical management at Salesforce.com

Stephen Denning

Stephen Denning's latest book, *The Leader's Guide to Radical Management* (Jossey-Bass, 2010), describes management principles and practices required to reinvent management to promote innovation and adaptation (steve@stevedenning.com). He is also the author of *The Leader's Guide to Storytelling* (2nd edition, 2011) and *The Secret Language of Leadership* (2007).

Salesforce.com is one of the most successful examples of a company transitioning from traditional management to radical management. This case explores the factors that made the implementation take hold throughout the company and identifies the pitfalls that were avoided. Salesforce.com provides on-demand services for customer-relationship management. As of mid-2011, the firm has more than 5,500 employees, revenues of around $1.7 billion and a market capitalization of around $17 billion. It routinely processes over 100 million transactions a day and has over 3 million subscribers. The services technology group, the focus of this case, is responsible for all product development inside the firm and has grown 50 percent per year since its formation eight years ago.

In April 2011, *Forbes* magazine examined how well chief executives delivered value to shareholders relative to their total compensation. By that measure, the most effective CEO on the planet was Marc Benioff of Salesforce.com.[1] Over the past five years Salesforce.com has delivered a 41 percent annual return to shareholders. This extraordinary performance – in fact, an example of an innovation turnaround – occurred after the firm instituted a set of radical practices that make it a model for business seeking to grow though continuous innovation.

The pervasive disease of slowing growth

Five years ago, Salesforce.com growth was in decline. In its early years, the group was delivering an average of four major software releases each year. By 2006, the pace had slowed to one major release a year. Observers attributed the innovation decline to the inevitable ''maturity'' of the company. Like most growing companies, over time Salesforce.com introduced systems intended to promote disciplined execution but which also put a brake on innovation. As in most large businesses, sooner or later bureaucracy sets in, collaboration across silos becomes difficult and innovation falters. The symptom is the creeping dysfunctionality of hierarchical bureaucracy.

What made Salesforce.com different is that when it innovativeness was threatened it adopted a radically different way of managing work. It implemented a set of agile, customer-driven, outcome-oriented, iterative management practices known in software development as Scrum.

None of these practices is by itself new. A few pioneering firms have learned to succeed by doing all the practices together in a disciplined way of getting all work done.

When the practices are generalized in this way, and applied beyond software development, they can be collectively described as ''dynamic linking,'' to distinguish them from the traditional management practices of hierarchical bureaucracy, where individuals reports to bosses to produce outputs.

 VOL. 39 NO. 6 2011, pp. 4-10, © Emerald Group Publishing Limited, ISSN 1087-8572 DOI 10.1108/10878571111176574

What are the practices of Scrum?

The practices of Scrum use a vocabulary that can sound esoteric outside the world of software development – terms such ''sprints,'' ''burndown charts,'' ''product owner,'' ''scrum-master.'' The core practices of Scrum are:

1. The work is organized in short cycles.
2. The management doesn't interrupt the team during a work cycle.
3. The team effectively reports to the client, not the manager.
4. The team estimates how much work is involved.
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. Work goals are defined before each cycle starts.
9. Work goals are defined through user stories.
10. Impediments are systematically removed.

Often when such a radically different approach to management is introduced in one part of the organization, there is tension at the interface between the traditional hierarchical bureaucracy and the new way of managing. Conflict arises because the two parts are operating in different modes and at different speeds. A common outcome is that the hierarchy effectively kills the new way of managing.

One way of solving the problem is to go all-out with change across the whole organization from the start. This is what Salesforce.com did in 2006. The results – both in terms of rapid innovation that is well received by customers and financial success – have been extraordinary.

Going all out with Scrum succeeded

By 2007, Salesforce.com had astonished the world of software development by successfully completing a transformation from traditional management to the radical management practices of Scrum in just three months.[2] The conventional wisdom is that implementation of such a radical change requires staff buy-in, and so a gradualist approach is usually adopted. Salesforce.com did the opposite. Developer Mike Cohn reports in his book, *Succeeding with Agile: Software Development Using Scrum* (Addison Wesley, 2009):

> During the first year of making the switch, Salesforce.com released 94 percent more features, delivered 38 percent more features per developer, and delivered over 500 percent more value to their customers compared to the previous year . . . Fifteen months after adopting Scrum, Salesforce.com surveyed its employees and found that 86 percent were having a ''good time'' or the ''best time'' working at the company. Prior to adopting Scrum, only 40 percent said the same thing. Further, 92 percent of employees said they would recommend an agile approach to others.

Why did Salesforce.com succeed?

The problems of Salesforce.com in 2006 were the familiar ones of hierarchical bureaucracy. The work was being done in a conventional sequential fashion. Software was late. Schedules were not met. Bugs were accumulating. Management couldn't figure out exactly why. The features that the teams most needed to be working on weren't identified as crucial until late in the production cycle. As the teams grew bigger, productivity declined.

The firm's leadership saw the need for change. It created a cross-functional team to address the problems of slowing velocity, decreased predictability, and product stability. This team rebuilt the software development process from the ground up using key values from the company's founding: keeping things simple, iterating quickly, and listening to customers. These values were a natural match for radical management.

The leadership saw the transformation not so much as a wholly new approach, but rather a return to the firm's core values. Three other elements helped the transition. First, the firm's on-demand software model was a good fit for iterative methods. Second, an extensive automated test system was already in place to provide the backbone for the new methodology. And third, a majority of the R&D organization was working at the same location.

A fast, open comprehensive approach

A document was drafted and shared that described the new process and its benefits, and explained why the firm was moving away from the old process. The team held forty-five one-hour meetings with key people at all levels in the organization. Feedback from these meetings was incorporated into the document after each meeting, molding the design of the new process and creating broad organizational support for change. This open communication feedback loop allowed a large number of people to participate in the design of the new process and engage actively in the solution.

One team in the organization had already successfully run a high-visibility project using iterative methods. This experience served as an example for other teams. At this point, the conventional wisdom would have been to pursue an incremental approach using pilot projects and a slow rollout. The management instead opted for a ''big-bang'' implementation, moving all teams to the new process at the same time. It was a difficult decision. The key factor driving it was a wish to avoid organizational dissonance and a desire for decisive action. Everyone would be doing the same thing at the same time.

The process started by sending a large group of people, initially program and functional managers, to training and buying training books for all staff. Three key members from the cross-functional team developed a consolidated presentation and training deck that included concepts from the current methodology. Two-hour training sessions were held for every team. In addition, training was given to the proxy client representatives who would be setting priorities as ''product owners.'' They also created an internal, wiki-based Web site as a reference for team members as they made the transition to the new methodology and for information about the change process.

The cross-functional team did its work in an iterative fashion and focused daily on whatever was needed to make the implementation successful. It created an overall roadmap for the entire process, provided coaching and guidance, identified and removed systemic impediments to change, monitored success, and evangelized the new way of working throughout the organization.

Key features of the change included a focus on team output rather than individual productivity and cross-functional teams that met in daily ''standups'' where team members shared what they had done yesterday, what they were going to do that day, and what impediments were being encountered.

> **"Over the past five years Salesforce.com has delivered a 41 percent annual return to shareholders. This extraordinary performance – in fact, an example of an innovation turnaround – occurred after the firm instituted a set of radical practices that make it a model for business seeking to grow though continuous innovation."**

> **“What made Salesforce.com different is that when it innovativeness was threatened it adopted a radically different way of managing work. It implemented a set of agile, customer-driven, outcome-oriented, iterative management practices known in software development as Scrum.”**

All teams used a simple iterative process with a common vocabulary, with prioritized work programs for each stage. They planned the work with user stories, estimated tasks with planning poker,[3] and defined organizational roles using the common Scrum terminology for all teams. The result was a new release of software every thirty days.

Key learnings from the Salesforce.com experience

In their own review of the lessons learned from the experience[4], managers Chris Fry and Steve Greene included several factors:

- Strong executive support. At several points in the transition, boundaries were tested. Without strong executive support, the transition might have failed. For example, a key executive decision was to stick to the release date regardless of the content of the release. Although many teams argued for more time to add more features, the executive management team stuck to the release date. Their ability to hold firm reinforced the principles of delivering early and often, reducing waste, and sticking to the deadline no matter what.
- A strong nucleus to lead the charge: Having a dedicated, fully empowered leadership team built from a cross-section of the organization was key to the program's success. This team was authorized to make decisions. It used the new methodology for its own work, and held meetings in a public space where everyone could see what was going on. This enhanced accessibility, transparency, and shared ownership of the transition. They also brought in industry experts and other companies that had adopted similar techniques.
- Principles ahead of mechanics. Focusing on the principles rather than the mechanics also helped people understand why the firm was moving to a new way of working. When teams ran into a problem, they could refer back to the principles and adjust anything they thought did not correlate with the principles.
- Total openness. During the introduction, radical transparency was a key to success. All of the daily meetings were held in a public place so that everyone could see how things were progressing. A task board was displayed on the public lunchroom wall so that everyone could track what was going on. The willingness to share information with everyone enabled people to adapt on a daily basis to what was happening.

Six common mistakes that Salesforce.com didn't make

Salesforce.com had more spectacular results than most other companies that have adopted the radical management principles of Scrum. As a rule of thumb, it is estimated that some 70 percent or more of Scrum implementations fall short of their projected benefits. What were the implementation mistakes observed in companies that Salesforce.com didn't make?

Mistake #1: Introduce the change as just another business process

The management at Salesforce.com saw that Scrum involved not just the adoption of a new business process, but constituted as a fundamental transformation of the way work was managed in the company. They were introducing a new way of thinking, speaking and acting in the workplace.

This meant major changes for both managers and workers:

- Those managing the work had to shift from being controllers of individuals to enablers of self-organizing teams, with responsibility for establishing priorities in a timely fashion and removing impediments as they are identified, rather than controlling each worker.
- Those doing the work had to shift from being individuals reporting to bosses in accordance with a big plan to having the responsibility and accountability of delivering work as part of a self-organizing team and doing the work in short cycles without interruption from any manager.
- Unlike traditional management, where defensive routines operate up and down the hierarchy, concealing both problems and responsibility for them, the new way of working makes both accomplishments and problems instantly visible. Accountability for both managers and workers is radically enhanced.
- These shifts constitute a deep change for many individuals. Unlike what typically happens in a bureaucracy, everyone is visibly accountable for what they have – or haven't – done.
- Sometimes the organization blames the new management approach for causing the problems that are now painfully visible, when in reality, the new management approach has simply revealed problems that have been festering for some time.
- The better organizations summon up the courage to begin the addressing the problems, while other organizations find it easier to push the problems back under the rug.

Thus it is one thing to add a new business processes. It is quite another for individuals to realign their whole way of thinking, speaking and acting in the workplace.

When firms adopt Agile or Scrum as merely a new business process, too often they fail to fully embrace the new values. When participants come back from a two-day training course, totally pumped up, they are usually faced with the same silos of hierarchical bureaucracy that won't work iteratively in accordance with the radical management philosophy. They are still facing big plans, complicated processes and top-down command-and-control leadership. In effect, the organization remains at war with itself.

As software developer, Mike Cottmeyer puts it: ''Agile adoption and agile transformation is not the same thing. Agile adoption is about introducing practices, the doing side of the equation. Agile transformation has to do with change, the being side of the equation.''[5]

The leadership at Salesforce.com grasped this difference and saw that if a radically different approach to management were to be introduced in one part of the organization, there would be a tension at the interface between the part of the company still doing traditional management and the part managing work in the new way. The two parts would be operating in different modes and at different speeds. So they opted to go all out with change right across the whole organization.

Mistake #2: Top management hedges its bets

Because the iterative client-driven self-organizing teams of Scrum and Agile represent a radical departure from the top-down command-and-control hierarchical bureaucracy, traditional management often adopts a wait-and-see approach. If the change succeeds,

> **"By 2007, Salesforce.com had astonished the world of software development by successfully completing a transformation from traditional management to the radical management practices of Scrum in just three months."**

management will embrace it and celebrate it as its own. If it fails, they can say that it was not their idea: just another management fad that didn't work.

By contrast, the leadership at Salesforce.com made it clear from the outset that they were committed to making the change work. They embraced it from the start. They supported it as implementation proceeded. They were there at critical points in the transition, when boundaries were tested. The leadership persisted until the new approach became part of the culture.

Mistake #3: Rigidly apply a methodology conceived elsewhere

Some firms try to implement Scrum as a rigid methodology with no allowance made for the different requirements of different contexts. It's implemented with the exact terminologies, job descriptions and procedures that have been worked out in other organizations. This can lead to considerable friction as the externally grown ideas don't fit the new environment. By contrast, Salesforce.com built on what had been learned in other organizations, but also adapted it to their firm's own context.

Mistake #4: Micromanage the change

The Scrum/Agile philosophy implies a radical shift in the traditional role of managers from controllers of individuals to enablers of self-organizing teams working in short iterations. It involves creating the space where those doing the work have the autonomy to apply their full talents and creativity to producing something will delight the customer, without interruption from managers. In effect, the customer becomes the boss. For many managers and workers, these are deep changes. If the shift itself is imposed from above in some peremptory eight-step program, there is a considerable risk that the new approach will be misinterpreted as a continuation of top-down command-and-control management. In contrast, the leadership at Salesforce.com modeled just the new management philosophy of direction-setting and enablement.

Mistake #5: Keep key management decisions secret

Hierarchical bureaucracy is notoriously non-transparent. Each layer of the bureaucracy tends to tell the layer above and the layer below what it wants to hear, rather than everything it needs to know. Finding out what's really going depends on access to informal networks. When Scrum is introduced into such a context, the risk of miscommunication and misunderstanding is high, as long-standing problems quickly emerge.

Salesforce.com dealt with the challenge by embracing total openness. The willingness to share information with everyone enabled people to adapt on a daily basis.

Mistake #6: Skimp on training and coaching

A study at Yahoo! showed that the provision of external coaches had a high payoff in terms of team productivity.[6] Traditional management tends to ignore such studies, because the mental model is that the managers are responsible for productivity. When the very future of the organization depends on success, skimping on coaching can be highly counterproductive. By contrast, Salesforce.com put a huge emphasis on training and coaching.

The application of Scrum beyond software

The success of software development at Salesforce.com, along with similar customer-driven iterative methods in auto manufacture at firms like Toyota, has led to the spread of this different way of managing to other company functions. Once a firm sees the dramatic benefits of client-driven iterative work practices in one area, it becomes natural to ask: Why not do all work in this fashion?

What to call these management practices as they expand beyond their origins? In software development they are known as Scrum and Agile Software Development. In manufacturing, they are often called Lean Manufacturing. The terminology of Scum, Agile and Lean are not strictly applicable in these wider applications. For example, the Agile Manifesto dwells on the goal of work as ''working software,'' which is appropriate for software development, but not relevant for other sectors.

Scrum, Agile and Lean are in effect subsets of a family of management practices, which can be called ''radical management.''

Notes

1. ''The best and worst CEOs for the buck'', *Forbes*, http://blogs.forbes.com/frederickallen/2011/04/14/americas-best-and-worst-bosses-for-the-buck/
2. The early history of iterative approaches in software development is described in detail by Craig Larman and Victor Basili in ''Iterative and incremental development: a brief history,'' *Computer*, 2003, 36(6), 47-56. Iterative approaches to work build on the 1930s work of Walter Shewhart, a quality expert at Bell Labs who proposed a series of short plan-do-study-act (PDSA) cycles for quality improvement: Shewhart, W. *Statistical Method from the Viewpoint of Quality Control*, New York: Dover, 1986. (Originally published 1939.)
3. Planning Poker: www.planningpoker.com/
4. Steve Green and Chris Fry: ''Large scale agile transformation,'' Agile 2007 conference: www.slideshare.net/sgreene/salesforcecom-agile-transformation-agile-2007-conference
5. ''Two dimensions of agile transformation'', Mike Cottmeyer, 15 January, 2011. www.leadingagile.com/2011/01/two-dimensions-of-agile-transformation/
6. For the twelve principles behind the manifesto: http://agilemanifesto.org/principles.html

Corresponding author

Stephen Denning can be contacted at: steve@stevedenning.com