WHITEPAPER

salesforce.com Success. Not Software.®

Agile Development Meets Cloud Computing for Extraordinary Results at Salesforce.com

force.com™ Dream. Build. Succeed.™

Contents

Executive Summary

Since agile development was invented in the mid-1990s, it has revolutionized how software is created by emphasizing short development cycles based on fast customer feedback.

With agile development, the application is constantly subjected to the reality check of actual users putting it through its paces. As a result, developers are less likely to get ahead of themselves by guessing what people will want. Nor are programming teams forced to estimate what they can deliver months in advance—an unattainable feat. Instead, R&D is more likely to build features people actually want. And because developers are looking ahead for weeks, not months, major new releases are far more likely to be delivered on time. Developers using this methodology call the process "continuous improvement." For the organizations that have made this process work for them, there is no going back.

But for much of its history, agile development was missing a crucial component: a development platform that supports, rather than thwarts, the rapid development cycles that make the methodology work. In traditional software environments, new software distribution is an ordeal that requires patches, reinstallation, and help from the support team. In such an environment, months or even years are needed to get a new distribution into the hands of users. Incorporating their feedback into the next release then requires comparable time.

It is here that cloud computing makes a substantial difference. Cloud computing eliminates the cumbersome distribution requirements that can bring agile development to a crawl. There are no patches to distribute, and no reinstallations are needed. With cloud computing, new distributions are installed on hosted servers and made available to users immediately. As a result, it's possible that the application you run today was modified just the night before. Looking back with 20-20 hindsight, it's clear that cloud computing is the component agile development has been waiting for.

One of the best examples of bringing together agile development and cloud computing is the experience of salesforce.com where, in late 2006, the R&D team moved to agile development. Few enterprise companies have adopted agile development as successfully as salesforce.com. The reason now seems clear: agile development was made to order for the Force.com platform—and vice versa—by leveraging the advantages of cloud computing. After just 2 weeks of training, the development staff, consisting of 30 teams, made the transition to agile development.

As salesforce.com has shared its experience with agile development on the Force.com platform with its customers, many have reported comparable successes. This paper shows the dramatic results possible when agile development meets Force.com.

Agile Development Brings Dramatic Improvements

In the last 7 years, salesforce.com's R&D organization has grown from a small team of developers to a large, cross-functional organization. As the organization scaled up and the complexity of the service increased, release dates for major releases began to slip. By 2006, during the last of the company's "pre-agile" days, salesforce.com had dropped from four seasonal releases down to one—a development that goes against the very grain of cloud computing. That release, which took about 15 months to complete, was the wakeup call the company needed to make a radical change in its development process.

That change was agile development. Since moving to this model, R&D has deployed each of the four successive major releases on the day it was scheduled. Salesforce.com is back on a regular delivery schedule, and R&D is confident that new releases will continue to be delivered on time, every time. Figure 1 compares pre- and post-agile development timelines and releases at salesforce.com.

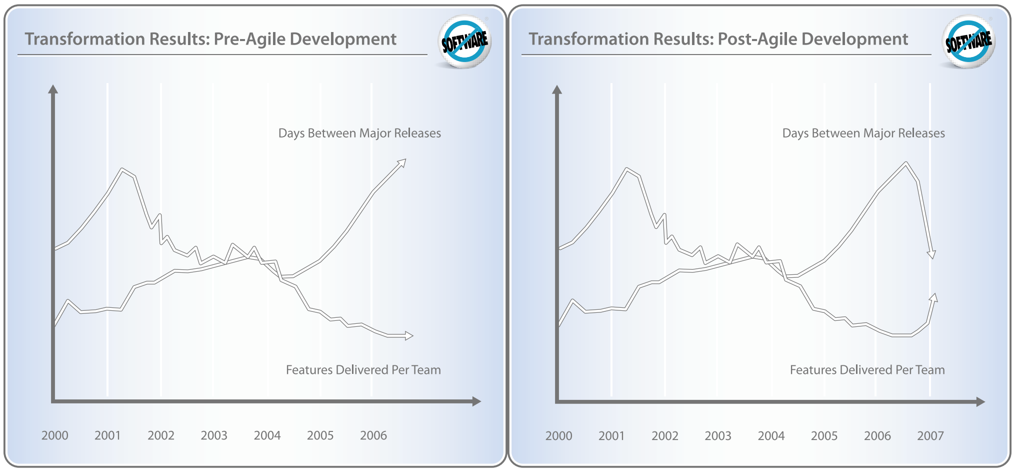

Figure 1: Pre- and post-agile development at salesforce.com.

Salesforce.com's R&D organization has benefitted in several ways from its transition to agile development:

- Increased delivery rate and created a process that makes customers and R&D happy
- Increased time to market of major releases by 61 percent
- Achieved a Net Promoter Score of 94 percent, a good indicator of customer satisfaction
- Convinced 90 percent of the R&D team to recommend the methodology to colleagues inside and outside the company
- Increased productivity across the organization by 38 percent, as measured by the number of features produced per developer (a side benefit not anticipated as part of the original goals)

Some salesforce.com customers have followed this example and benefitted as well. For example, CODA used agile development in developing CODA 2go, a new software-as-a-service (SaaS) accounting application built natively on the Force.com platform. Product Director Debbie Ashton reports that the agile approach encouraged the development team to break the process into smaller, more manageable stages, leading to more accurate delivery forecasts and faster time to market. She also credits the approach for making development more transparent, allowing stakeholders to see and influence product functionality earlier and resulting in features that map more closely to requirements. In turn, the team gained a better appreciation of how requested changes are prioritized, while being "more agile" in its ability to change focus.

Agile Development Benefits: Predictability, Transparency, & Energy

For salesforce.com, the agile development methodology provides three main benefits:

- Predictability through "timeboxing"
- Transparency
- Energy, fueled by creativity and job satisfaction

Predictability

Salesforce.com's number-one goal in moving to agile development was a more predictable release cycle. In conventional "waterfall" development, an organization tries to predict what functionality can be delivered 6 months to 1 year out—and makes promises it cannot keep because of large unknowns in the cycle.

Agile development's answer to this dilemma is counter-intuitive: predictability comes when releases are "timeboxed," keeping the date, resources, and quality control fixed while keeping functionality decisions flexible. Salesforce.com now segments its releases into smaller pieces, with a narrower scope that can be completed in 1 month or less. Compare this process with a waterfall process, in which priorities are set at the beginning of the new cycle and never benefit from customer feedback until months or even years later.

Shorter cycles can incorporate customer feedback quickly into the next major release. Shorter cycles have also increased the importance of maintaining a priority list and focusing, with laser-like attention, on the items at the top of that list. Customer feedback allows constant reassessment of the list, and, if necessary, reprioritization for the next cycle. In this process, it helps to have a strong conduit to the user community, which the team has built with salesforce.com's IdeaExchange customer forum.

The agile development value of promising only what you can deliver is also evident in the "sprint reviews" held at the end of each iteration cycle. These sessions let teams show the organization what they've accomplished. Teams only show work that is ready to ship; work in progress is not factored in.

Transparency

Force.com platform division head Steve Fisher is fond of saying that agile development has given him the best handle on the state of an R&D team in his 25-year career. The key to this breakthrough is transparency made possible by a set of engineering practices that includes continuous integration, unit and functional test automation, and daily meetings and information "radiators" that continually show the progress of every team, displayed on a wall for all to see.

Salesforce.com has also made transparency an external value, thereby building trust with the customer base. The company now provides customers with readouts of the state of the system over the last 30 days at http://trust.salesforce.com. This practice contrasts with the usual approach of hiding problems from the outside world. With Salesforce CRM, if things go wrong, customers can see any problems and how they are being resolved.

Energy

Agile development has helped recapture the spirit of creativity and excitement that marked salesforce.com's days as a startup. As companies grow, the centralized decision-making process that has served them well sometimes begins to strangle creativity. Agile development helps decentralize the process by pushing responsibility and decision-making to the development teams.

Continuous improvement—an agile development cornerstone—also creates a better work experience because everyone from developers and QA engineers to documentation writers can improve how they work individually and contribute to how the team functions. Productivity is its own reward. And when you give smart, capable people a tangible sense of accomplishment, job satisfaction rises.

Bringing Agile Development to Salesforce.com

In making the dramatic switch to an agile development methodology, salesforce.com technical personnel began with an important premise: that this methodology was true to the company's core values—keep it simple, iterate, and listen to your customers. In a way, agile development recaptured the spirit of the startup salesforce.com once was.

The Challenges of Converting to Agile Development

To begin the process, a newly formed cross-functional team vetted the initial proposal in 45 1-hour meetings with key stakeholders at all levels of the organization. Feedback from each stakeholder was incorporated into the proposal document, which was presented to the technical executive team at an offsite.

Once the team had the green light, it took a "big bang" approach: converting the entire organization overnight. The move was not without risks. Skeptics argued that several teams might repeat mistakes that could be caught by a few starter teams. They were also concerned there wouldn't be enough coaches to assist the teams every day. On the other hand, a big-bang rollout would avoid organizational dissonance and send a clear signal that salesforce.com was taking decisive action.

An early step in the transformation was to send a large group of engineers to Certified ScrumMaster training and to seed the office with agile development books, while the cross-functional team developed a presentation and training deck. All 30 teams participated in 2-hour training sessions; a wiki-based Web site was created to house everything related to agile development.

The biggest hurdle wasn't the mechanics of agile development, but the mindset of the team members. The methodology required a different way of thinking, which can be difficult if you've been developing for a decade or two. People weren't shy about letting the team know what they thought, with comments like these: *"We spend more time talking about scrum than we spend time working on Salesforce"; "Stop trying to implement scrum and look at how many releases we can really do in a year"; and "The lingo is ridiculous."*

The Tipping Point

The tipping point in favor of the adoption of agile development came about 4 months into the transition. As people started to master the mechanics, they voiced fewer frustrations. The rollout team received fewer negative comments. The release schedule became more predictable and regular. Gradually, the entire group realized that agile development was worth the effort.

Since the initial rollout, salesforce.com has made continuous improvements to agile development by dividing the process into four phases: rollout, adoption, excellence, and expansion, as shown in Figure 2.

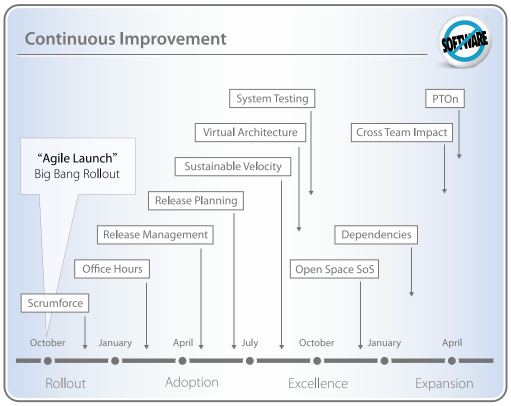

Figure 2: The agile development process at salesforce.com.

An early milestone in continuous improvement was the development of Scrumforce, an internal backlog management tool built in just 3 weeks on the company's Force.com platform using the Force.com Apex programming language and the Visualforce user interface development tool. Customized to manage the work of any agile development team, Scrumforce has been a key factor in salesforce.com's ability to manage agile development at the enterprise level.

Lessons Learned

While bringing agile development to salesforce.com, the team discovered key elements for success and identified changes that would have made the transition easier:

Keys to success:

- Make sure your executives commit to the change
- Focus on principles over mechanics
- Focus on getting several teams to excellence
- Focus on automation
- Experiment: be patient and expect to make mistakes; when the heat is on, however, stick to your guns

Potential changes:

- Involve more individual contributors earlier
- Train product owners earlier and more thoroughly
- Get outside coaching earlier
- Provide key executives with concrete deliverables around the time of the rollout
- Be clearer about defining the "rules" related to agile development

Computing and Agile Development: A Great Combination

If salesforce.com's experience has demonstrated anything, it's that cloud computing is the ideal environment for agile development. It lets you get valuable functionality to your customers quickly, collect immediate feedback, and make rapid changes based on that feedback. These fast development cycles, an inherent advantage of cloud computing, are impossible to implement in the traditional development model because of the high cost of distribution.

Leveraging the Force.com platform, salesforce.com's R&D team not only produces four major releases a year, but makes incremental improvements on a monthly, weekly, and even daily basis. Customer feedback is now woven into the process at every stage—an email sent or idea posted on the IdeaExchange today may drive tomorrow's functionality.

Salesforce.com has the advantage of having the R&D team develop in a single, shared code base. As a result, the team supports only the current production release and next release—not years of legacy releases. That means that each day, hundreds of people check into the same code base. When someone makes a change that may break existing code, everyone immediately knows about it, so there's no time wasted in the merge and integration processes later in the cycle.

With agile development in place, salesforce.com will continue to expand its use to every team within the company. And beyond its walls, salesforce.com's message is one of synergy: the combination of the Force.com platform and agile development add up to more than the sum of the parts.

For More Information

Contact your account executive to learn how we can help you accelerate your CRM success.

Corporate Headquarters
The Landmark @ One Market
Suite 300
San Francisco, CA, 94105
United States

1-800-NO-SOFTWARE
www.salesforce.com

Latin America
+1-415-536-4606

Japan
+81-3-5785-8201

Asia/Pacific
+65-6302-5700

Europe, Middle East & Africa
+4121-6953700

salesforce.com
Success. Not Software.