Large Scale Agile Transformation in an On-Demand World

Chris Fry and Steve Greene
Salesforce.com

Abstract

Salesforce.com has recently completed an agile transformation of a two hundred person team within a three month window. This is one of the largest and fastest "big-bang" agile rollouts. This experience report discusses why we chose to move to an agile process, how we accomplished the transformation and what we learned from applying agile at scale.

1. Introduction

This report describes our successful agile transformation. In three months we have moved thirty teams from waterfall development to agile development. We have focused on creating self-organizing teams, debt-free iterative development, transparency and automation. This report covers the background of the project, the results, lessons learned, advice for others and conclusions.

Within the last six months we have benchmarked our progress, completed two major releases and continue to deliver potentially deployable code each month. In our latest organizational survey 87% of our technology staff believe that their scrum team is self organizing and 80% believe that our new development methodology is making their team more effective. We are continually trying to improve our organization and agile has provided a framework for continuous improvement.

2. Project Background

Salesforce.com is a market and technology leader in on-demand services. We routinely process over 85 million transactions a day and have over 646,000 subscribers. Salesforce.com builds a CRM solution and an on-demand application platform. The services technology group is responsible for all product development inside Salesforce.com and has grown 50% per year since its inception eight years ago, delivering an average of four major releases each year. Before our agile rollout we had slowed to one major release a year. The agile rollout was designed to address problems with our previous methodology:

- Inaccurate early estimates resulting in missed feature complete dates and compressed testing schedules.
- Lack of visibility at all stages in the release.
- Late feedback on features at the end of our release cycle.
- Long and unpredictable release schedules.
- Gradual productivity decline as the team grew.

Before the agile rollout the R&D group leveraged a loose, waterfall-based process with an entrepreneurial culture. The R&D teams are functionally organized into program management, user experience, product management, development, quality engineering, and documentation. Although different projects and teams varied in their specific approaches, overall development followed a phase-based functional waterfall. Product management produced feature functional specifications. User experience produced feature prototypes and interfaces. Development wrote technical specifications and code. The quality team tested and verified the feature functionality. The documentation team documented the functionality. The system test team tested the product at scale. Program management oversaw projects and coordinated feature delivery across the various functions.

Our waterfall-based process was quite successful in growing our company in its early years while the team was small. However, the company grew quickly and became a challenge to manage as the team scaled beyond the capacity of a few key people. Although we were successfully delivering patch releases, the time between our major releases was growing longer (from 3 months to over 12). Due to fast company growth and lengthening of our release cycles, many people in R&D had not participated in a major release of our main product. Releases are learning opportunities for the organization. A reduction in releases meant fewer opportunities to learn. This had a detrimental affect on morale and on our ability to deliver quality features to market.

3. Our Transition Approach

An original company founder and the head of the R&D technology group launched an organizational change program. He created a cross-functional team to

address slowing velocity, decreased predictability and product stability. This cross-functional team redesigned and rebuilt the development process from the ground up using key values from the company's founding: KISS (Keep it Simple Stupid), iterate quickly, and listen to our customers. These values are a natural match for agile methodologies.

It was very important to position the change as a return to our core values as a technology organization rather than a wholesale modification of how we deliver software. There were three key areas that were already in place that helped the transition: 1) the on-demand software model is a natural fit for agile methods; 2) an extensive automated test system was already in place to provide the backbone of the new methodology; and 3) a majority of the R&D organization was collocated.

One team member wrote a document describing the new process, its benefits and why we were transitioning from the old process. We led 45 one-hour meetings with key people from 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 buy-in for change. This open communication feedback loop allowed everyone to participate in the design of the new process and engage as an active voice in the solution. Two key additions to the initial paper were a plan for integrating usability design and clarification on how much time we needed for release closure sprints.

At this point, most literature recommended an incremental approach using pilot projects and a slow roll out. We also considered changing every team at the same time. There were people in both camps and it was a difficult decision. The key factor driving us toward a big-bang rollout was to avoid organizational dissonance and a desire for decisive action. Everyone would be doing the same thing at the same time. One of the key arguments against the big-bang rollout was that we would make the same mistakes with several teams rather than learning with a few starter teams and that we would not have enough coaches to assist teams every day. One team in the organization had already successfully run a high visibility project using Scrum [1]. This meant that there was at least one team that had been successful with an agile process before we rolled out to all the other teams. We made a key decision to move to a "big-bang rollout" moving all teams to the new process rather than just a few.

We started by sending a large group of people (initially program and functional managers) to Certified ScrumMaster training and buying agile books for the office. Three key members from the cross-functional team developed a consolidated presentation and training deck that included concepts from the current methodology, Scrum [1], XP (eXtreme Programming) and Lean methods [2]. We facilitated two-hour agile training sessions for every team. In addition, we provided Certified Scrum Product Owner training and Agile Estimating and Planning [3] training on-site. We also created an internal, wiki-based website as a repository for all our information and as a valuable reference for team members transitioning to the new methodology.

Our cross-functional rollout team was run using Scrum and focused daily on making the rollout successful. The team created a global schedule for the entire product, provided expertise, coaching and guidance, removed systemic obstacles to change, monitored success, and evangelized our agile vision throughout the organization.

Some of the key wins since the rollout have been:

- Focus on team throughput rather than individual productivity
- Cross-functional teams that now meet daily
- Simple, agile process with common vocabulary
- Prioritized work for every team
- A single R&D heartbeat with planned iterations.
- User stories & new estimation methods
- Defined organizational roles – ScrumMaster, Product Owner, Team Member
- Continuous daily focus on automated tests across the entire organization
- Automation team focused on build speed & flexibility
- Daily metric drumbeats with visibility into the health of our products and release
- Product line Scrum of Scrums provide weekly visibility to all teams
- R&D-wide sprint reviews and team retrospectives held every 30 days
- Product Owner & ScrumMaster weekly special interest groups (SIGs)
- A time-boxed release on the heels of our biggest release ever
- Reduction of 1500+ bugs of debt
- Potentially release-able product every 30 days

Although we are still learning and growing as an organization, these benefits have surpassed our initial goals for the rollout. Some areas that we are still focusing on are: teamwork, release planning, bug debt reduction, user stories and effective tooling.

4. What we learned

The key takeaways from our rollout were to: 1) have executive commitment to the change; 2) create a dedicated rollout team to facilitate the change; 3) focus on principles over mechanics; 4) focus early on automation and continuous integration; 5) provide radical transparency and 6) leverage external agile training and coaching. These topics are expanded below.

Ensure executive commitment to the change. Executive commitment was crucial to implementing massive change. There were several key points in the transition where boundaries were tested. Without executive support the transition might have failed. For example a key executive decision was to stick to an aggressive release date, regardless of the content of the release. Although many teams argued throughout the development cycle for more time to add more features the entire executive management team stayed committed to the release date and the move to the new methodology. Their ability to hold firm reinforced the agile principles of delivering early and often, reducing waste and made it clear that we were doing a time-boxed release.

Create a dedicated, cross-functional rollout team. Another key to our success was a dedicated, fully empowered agile rollout team built from a cross-section of the organization. Each area of the organization nominated members for the group. We had members from quality engineering, development, program management, product management, user experience & usability, documentation and executive management. This team was empowered to make decisions, used the new methodology and held its meetings in a public space. This team provided accessibility, transparency and shared ownership of the transition. The team also reached out to industry experts and other similar software companies that had adopted agile techniques.

Focus on principles over mechanics. Focusing on the principles of agile rather than the mechanics also helped people understand why we were moving to an agile process. The principles from the lean movement [2] also were key to communicating the value of changing current behavior. If teams were feeling that something was not working "the way it should," they could refer back to the values and reject anything they thought did not correlate with our core values. We focused on the following agile values: communication, empowered teams, continuous improvement and delivering customer value early. We published them on a handout that was distributed to the entire technology organization.

Focus on automation. An extensive automation suite and build system already existed to support the transformation. This was extremely helpful because we had a continuous integration system in place and a value system around automated unit and functional testing within the entire development organization. We improved this system during the rollout but did not have to create it from scratch. Everyone focused on code line health, driving down end-to-end test times and working together in a single, integrated codeline. We were required to make substantial efficiency improvements to the automated build system to allow much more frequent check-in/build/test runs. These quick runs were critical for the short development test cycles.

Provide radical transparency. During our rollout, transparency in everything that we did was a key to our success. We held all of our daily rollout meetings in a public place so anyone could see how the rollout was progressing. We visually displayed our task board on a public lunch room wall where everyone had access to the information. We over-communicated vision, information, guidance and plans to everyone. We implemented "daily metrics drumbeats" sent to the entire R&D team describing the health of the release in terms of automation results, test execution results, system testing results, open bug counts, and deployment activities. This bias to sharing information with everyone was critical in our ability to adapt on a daily basis to ensure our success.

Leverage existing agile training. The last key contributor to our success was sending a large set of people (approximately 25% of the R&D organization) to professional training and hiring external, experienced consultants to assist team members, ScrumMasters, Product Owners and functional managers with the process. This provided a foundation in agile principles and allowed us to scale the rollout team to provide support for all the teams. The external training and coaching exposed everyone inside the organization to agile success stories, lessons learned and best practices from other companies. This exposure aided and drove adoption. Several teams started innovating on their own by moving to two week iterations, focusing on team deliverables and experimenting with different physical and virtual task tracking methods.

5. What we would do next time

Although we have achieved many of our initial goals with our agile rollout (time boxed releases, self-organizing teams, automation, visibility), we think other teams could benefit from doing certain things

earlier than we did. These things are: 1) involve more individual contributors early; 2) train product owners early and with more intensity; 3) get outside coaching earlier; 4) work on automation early; 5) give key executives concrete deliverables around the rollout; and 6) be more clear about what the agile rules are. These are expanded below.

Involve more individual contributors early. Initially you may not get feedback from key employees. Current culture and attitudes that promote "the way we do things here" are powerful anti-change agents. One great way to involve everyone in your organization up front is to run an open space meeting. There are many ways to run a meeting like this but one way is to have everyone put their top three issues on note cards or sticky notes, group them into topics and then self-organize around a few key themes. Form a set of task groups that nominate leaders to drive resolution of the issues in the office with someone from the agile rollout team coordinating. We held these sessions later in the transition, doing them earlier would have helped.

Train Product Owners earlier and with more intensity. Throughout our initial rollout we heard from many experts that the Product Owner role was key to the success of our agile transformation. Although we intuitively understood this we didn't truly understand the significant changes that the Product Owners would experience in their role. They were required to prioritize and plan the release, needed to move to a more communication based paradigm and were directly involved with the day-to-day functioning of their teams. Early immersion and training of the Product Owners in agile principles, product backlog creation, user story design and estimation & planning is key to the success of any agile team. Also, beyond initial training, continuous Product Owner coaching throughout the rollout is necessary to ingrain the new process into the culture.

Get outside coaching earlier. Several of the outside coaches we brought in were able to quickly recognize ways to more quickly enable and coach our teams. They also recognized common patterns that we could correct and brought in lessons learned from other organizations transitioning to agile. Their experience helped drive adoption. Because they were external to the organization some people were more comfortable receiving constructive advice from our outside experts.

Prioritize build and test infrastructure early in the process. Automation is key to any agile methodology and making sure that you have automated and integrated builds is a key way to give visibility to the entire organization. Salesforce.com has invested in a large JUnit based set of functional and unit tests with integrated reporting and failure triaging.

Give key executives concrete deliverables around the rollout. Executives were key to our success. Giving them small or large tasks related to the agile rollout brings them into the organizational change program and helps them stay grounded in what you are doing.

Be more clear about what the agile 'rules' are. Self-organization can mean anything to anyone. Allowing teams to self-organize (as opposed to assigning tasks) is critical to real commitment and engaging the passion of team members. Avoiding partial credit by properly defining done is another aspect of self-organizing. It's important to be clear about your definition of done, what decisions are within the purview of the team and which ones are not. This helps the team to understand what flexibility they have to reach their sprint goals. It's also important to coach executives and functional managers to make changes at sprint boundaries, rather than within the sprint.

6. Advice for others

This section describes our advice for other people embarking on an agile transformation. Our advice is:

- Create a dedicated, fully empowered, cross-functional rollout team
- Don't be afraid to change the entire company at one time
- Get professional help
- Encourage peer to peer coaching
- Focus on getting several teams to excellence
- Create a company sprint heartbeat
- Decide early which tool you will use to manage the rollout (we built our own discussed below)
- Encourage radical visibility and over-communicate
- Be inclusive
- Be patient and expect to make mistakes

These topics are expanded below.

Create a dedicated, cross-functional rollout team. This team will become central to managing change and communicating within the organization. They will provide accessibility to everyone in the organization when issues arise and responsibility to address them. We suggest using your new process to run this team. Make sure you over-communicate changes.

Don't be afraid to change the entire company all at one time. Many people will tell you to experiment with a pilot project first then slowly rollout the process to

other teams. It is possible to change the company all at once and this can lead to significant benefits. It reduces cross-talk between teams functioning in the old way and the new way and helps move the entire organization to the new process.

Get professional help. External coaches have done it before and will see the roadblocks coming before you do. They can also help you learn from other organizations that have gone through similar transitions. They can provide new areas for you to consider adding into your process like user stories or estimation and planning.

Encourage peer to peer coaching. Discover who on your teams has the ability to master agile methods early or has experienced success with them in other organizations. These people can provide invaluable coaching and will see obstacles that you may miss.

Focus on getting several teams to excellence. Your intuition is often to focus on the teams that are struggling the most. By focusing on creating a few successful teams you will build momentum and create examples of what you can accomplish with the new process. We witnessed many of our lagging teams improve tremendously just by improving the other teams around them.

Create a company sprint heartbeat. We developed a one-month sprint cycle early on and had all teams in the same cycle. This allowed all of our sprint reviews to be coordinated on monthly boundaries, allowing our stakeholders and teams to be present at all of the reviews and feel the momentum building across the entire product line each month. Since, many teams have adopted shorter 2-week inter-sprint cycles, however we still maintain our monthly organization-wide sprint review cycle.

Decide early on the right tool. We are "dog fooding" our own platform to create an agile tool to manage development. Spreadsheets quickly became unmanageable and using our own product to build our product has great side benefits.

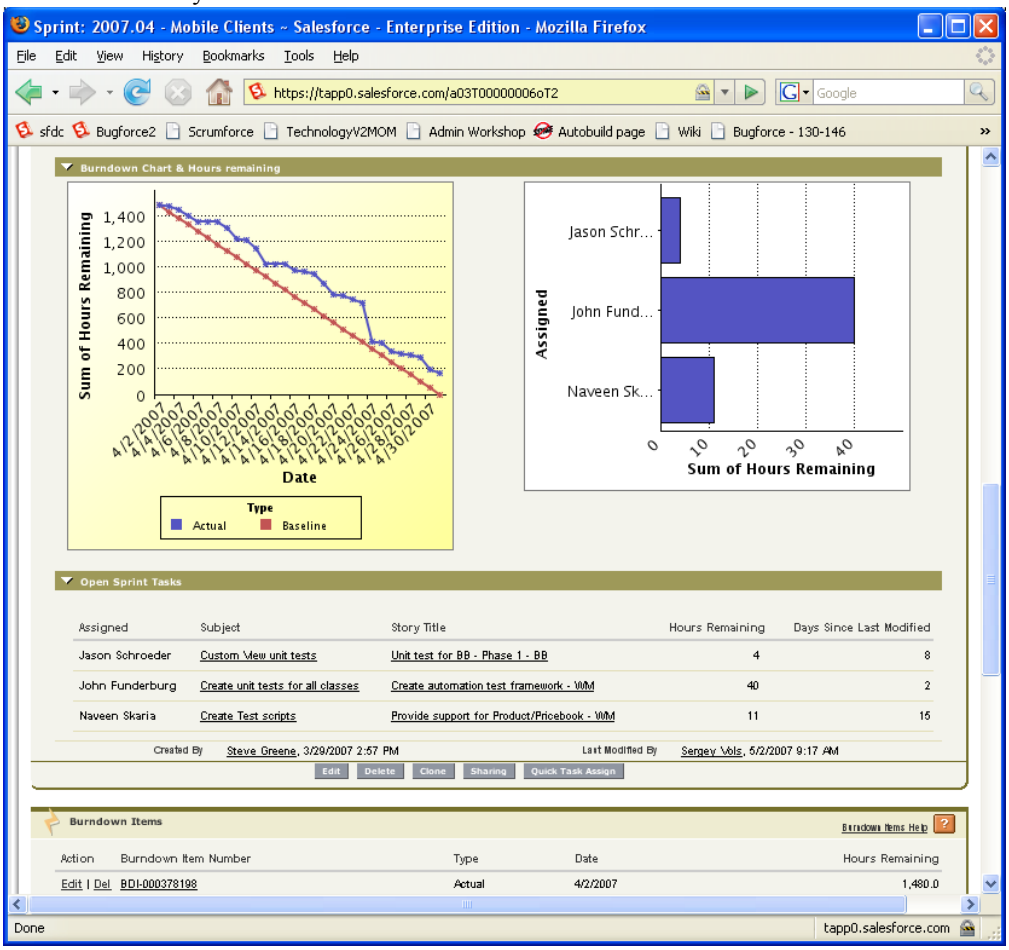

Figure 1: Our internal agile tool (ScrumForce)

Figure 1 illustrates our ScrumForce tool which every ScrumMaster, Product Owner and team member use to manage their work. Functional managers use built-in reporting to manage their teams and releases.

The tool itself is built using the Salesforce.com platform and gives every developer a reason to use our application every day. It provides drag and drop prioritization of user stories, user story management, task management and burndown chart creation.

Encourage radical visibility and over-communicate. Coming up with a tag line like "radical visibility" helps people overcome the inertia and fear of sharing information widely. Change is hard and often everyone is busy and not reading their email. So having multiple channels of communication and providing the same message over and over helps. When you think that your teams understand a new method or process, repeat your communication.

Be inclusive. Extend invitations to reviews to your entire technology teams. Sometimes help comes from people you don't expect but have a passion for a certain area. By casting your net wide you can encourage visibility and participation.

Be patient and expect to make mistakes. Encourage a culture of experimentation. You aren't going to get everything right, so set the expectation that you are going to make a few mistakes. Reward everyone on the team for experimentation: don't create a punitive environment around making mistakes.

7. Conclusion

This report places on the record a large, successful, big-bang transformation from waterfall to agile. If you are considering transitioning your organization consider moving all teams at the same time rather than staggering the rollout.

8. Acknowledgments

We would like to thank Mike Cohn, Pete Behrens, Peter Morelli, Andrea Leszek and Tom Poppendieck for providing comments on drafts of this report. They improved the original version, any mistakes are the authors.

9. References

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

[2] Poppendieck, M. and Poppendiek, T., *Implementing Lean Software Development*, Addison-Wesley, Boston, MA, 2006.

[3] Cohn, M,, *Agile Estimating and Planning*, Prentice Hall PTR, 2005.