Successful User Experience in an Agile Enterprise Environment

Melissa Federoff and Catherine Courage

Salesforce.com, The Landmark at One Market St., San Francisco, CA 94105
ccourage@salesforce.com, mfederoff@salesforce.com

Abstract. Salesforce.com is a leader in the enterprise Customer Relationship Management (CRM) marketplace. In 2006, salesforce.com's Research and Development (R&D) organization transitioned over 30 product teams from a waterfall development process to an agile one. The R&D department is responsible for producing all products offered to salesforce.com customers. After the transition, it was clear that User Experience (UX) team members were dissatisfied. When asked 6 months into the rollout if agile was making their work life better, only 24% agreed. This paper discusses how the team and management responded to this data and as a result reached an 85% satisfaction rate a year later.

Keywords: Agile, waterfall, enterprise, user experience, design, research.

1 Transition from Waterfall to Agile Development Process

R&D teams at salesforce.com are functionally organized into program management, user experience, product management, development, quality engineering, and documentation. Prior to agile, these teams leveraged a waterfall development process (Figure 1).

Program management oversaw projects and coordinated feature delivery across the various functions. Product management created business requirement documents that specified what was to be built. User experience produced and evaluated feature prototypes. Development wrote technical specifications and coded based on prototypes. The quality assurance team tested and verified the feature functionality. The documentation team documented the functionality. The system test team tested the product at scale. These functions were performed in a serial fashion.

As the company grew and the salesforce.com application gained complexity, development became increasingly unable to accurately estimate time and scope for new features using the waterfall method. Product teams suffered from feature creep, redesign work, extended development times and compressed testing schedules during development cycles. In October 2006, it had been almost a year since salesforce.com's last major release - a release that had been rescheduled five times.

In an effort to increase the number and accuracy of releases, the R&D organization decided to move all 30 plus development teams from a waterfall development process to an agile one. Agile is a philosophy toward software development that was

M.J. Smith and G. Salvendy (Eds.): Human Interface, Part I, HCII 2009, LNCS 5617, pp. 233–242, 2009.

established in an Agile Manifesto [2]. There are many tenets to this approach; its core lies in methods that are adaptive and people-oriented. One common agile methodology is 'Scrum' which is the process salesforce.com decided to espouse during the company's self termed "big-bang rollout" in 2006.

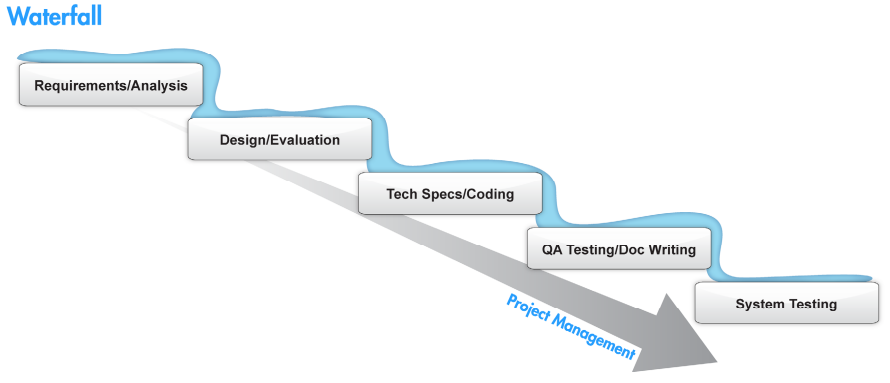

Fig. 1. A visualization of the Waterfall process used at salesforce.com

With Scrum, projects progress via a set number of time-boxed iterations called sprints (typically 2 - 4 weeks in duration). The goal at the end of each sprint is to have fully functioning code that has been tested and that could be released (Figure 2).

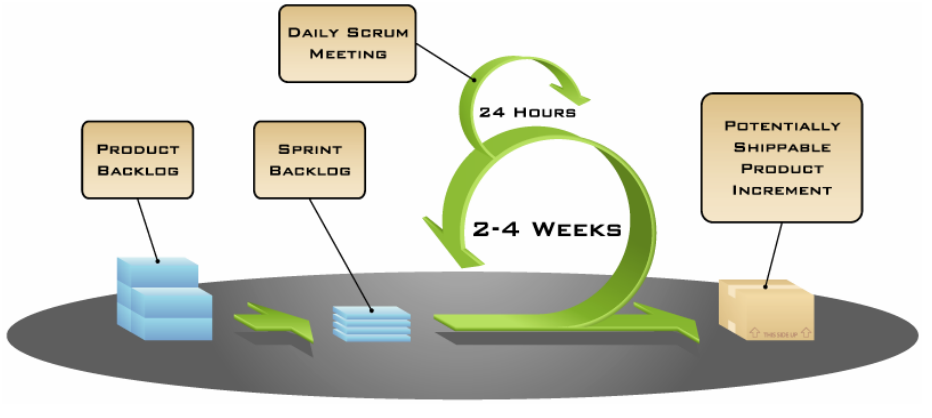

Fig. 2. A visualization of the Scrum process, taken from Mountain Goat Software (2005)

The procedure of the Scrum method at a high-level is as follows [3]: A Product Backlog is created by a product owner (a product manager typically fills this role). This backlog is a list of all desired changes to the product for the release.

At the start of each sprint a planning meeting is held with all members of the Scrum team (product management, user experience, documentation, quality assurance, development, project management). During this planning meeting the product

owner prioritizes the product backlog and the Scrum team selects a chunk of the backlog that it can complete during the coming sprint.

Items selected for the sprint are then moved from the Product Backlog to the Sprint Backlog. Because the time-frame is short this allows the team to effectively estimate and commit to work that it can reasonably accomplish. During the sprint, all items in the sprint backlog need to reach completion: at salesforce.com this means they will be designed, validated with users, coded, tested and documented.

During the sprint the Scrum team conducts a brief daily meeting called the Daily Scrum, which helps the team stay on track and creates visibility into any barriers to work completion. Program Managers take the roll of Scrum master which entails facilitating all meeting and planning sessions and ensuring that any roadblocks are removed.

This process repeats for a pre-defined number of sprints, and then work is released. Salesforce.com completes three sprints and then releases new features to customers quarterly.

1.1 Impact of the Transition to Agile on the R&D Organization

In the transition to agile, some successes were seen immediately. The most significant change for salesforce.com was that the R&D organization met the February 2007 release date. At the one year anniversary of moving to agile, the rollout team calculated overall improvements. Changes included a 61% improvement in mean time to release for major releases, a 94% increase in feature requests delivered in major releases, and a 38% increase in feature requests delivered per developer.

1.2 Initial Impact of the Transition to Agile on the User Experience Team

While moving to agile was clearly a success for the R&D organization at salesforce.com, the initial transition period was far from smooth for the User Experience team. In 2006, the vast majority of literature on the topic of agile did not include guidance on the inclusion of User-Centered Design (UCD) processes [4]. In agile literature, 'design' typically referred only to coding or system design [1]. The User Experience team was left with many unanswered questions; some included:

- How can we properly identify the target users and ensure that their needs are met within agile's just-in-time process?
- How can we accomplish holistic design when agile teams are planning and building features in a piecemeal fashion?
- How do UX professionals succeed across multiple teams when agile espouses one team per person?

The unhappiness of the UX team was illustrated clearly in the results of a survey sent out to the R&D organization about 6 months after the rollout. Eighty percent of total respondents felt that agile was making their Scrum team more effective. However, when looking at the data by functional area, only 30% of User Experience team members agreed with this statement. When asked about several other impacts of agile, the same trend was seen (Figure 3).

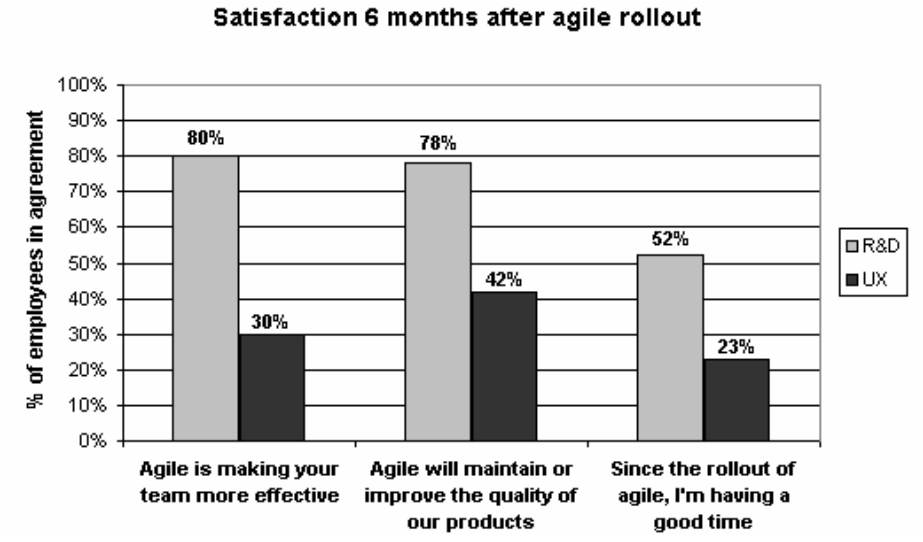

Fig. 3. UX satisfaction with agile as compared to all of the R&D Department

Some of the reasons cited by User Experience for their dissatisfaction with agile included:

- Assigned to too many teams
- Spending too much time in meetings
- Not enough time to complete work
- Lack of focus on the big picture

User Experience team members supported four product teams on average, while developers and other disciplines only supported one, as dictated by the Scrum process. For team members that are only assigned to one Scrum team, attending planning meetings, daily Scrums and retrospectives is a reasonable time investment. For those on multiple teams, attending all of those meetings meant having little time to do anything else, such as research or design work.

Tighter timelines were also mentioned. In Scrum, teams select what they will work on at the beginning of a sprint and then build those items to completion over the course of that sprint; in the case of salesforce.com that is a one month period of time, start to finish. These aggressive timeframes can cause UCD to be compromised, since investigating and iterating designs often takes longer than weeks, particularly for complex enterprise applications. In this initial phase, the User Experience team struggled to find ways to achieve the success that they had reached using waterfall. The R&D division made significant progress using agile, though, so the UX team had no choice but to adapt and evolve.

2 Strategies for Success

The road to agile success within an enterprise space was not an easy one for the User Experience team at salesforce.com. The key factors that drove the success included:

- A New Resource Plan
- Design Transformation
- Getting Usability RITE

2.1 A New Resource Plan

Management made the decision to reduce the number of Scrum teams per User Experience team member from four or more to a maximum of two. The process of determining which teams would receive assigned resources involved the UX managers meeting with the VPs of the different product areas to determine the priorities of the new features. UX management took these priorities and assessed them against the complexity of the features (e.g., would the feature require a brand new user interface). Assignments were made accordingly. The Scrum teams that did not have an assigned UX resource became responsible for working as a team to create the user interface as best they could.

This solution to de-support approximately 35% of teams was a situation that no one was satisfied with. Certainly, the UX team members were pleased with their new, realistic workload, but the decision to strand some Scrum teams was not accepted. Ease of use is a core value at salesforce.com. Placing the burden on the Scrum teams put that value at risk.

In the spirit of agile, UX team members brainstormed an alternate solution, and the concept of Office Hours (OH) emerged [5]. The UX team members determined that by giving two hours of their time per week, they would be able to assist Scrum teams that did not have assigned UX resources without spreading themselves too thin.

One of the goals was to make OH easy for everyone involved. In the spirit of agile, OH slots are self-service. Scrum teams are not scheduled by UX team members, but instead, Scrum teams are encouraged to make use of the time and schedule it when most appropriate for them.

Scrum teams are asked to bring user stories, design objectives, and a design artifact to OH to maximize the effectiveness of the session. A typical agenda looks like this:

- Project review – < 15 min – Cover the user stories and design objectives
- Artifact presentation – 10 min – Show the UX team member(s) the proposed design artifact
- Discussion – 35 min – Converse about the artifacts and objectives, and assess the design

The OH program has been successful on several fronts: job satisfaction is higher among UX team members; Scrum teams with minimal user experience needs are supported very well with OH; teams that might never have received any support are receiving some (e.g., departments outside of R&D such as Marketing). However, the program is a stop-gap and complex features do suffer if not given full design support. We have noticed a slight decrease in the overall consistency of deliverables for

complex features, and have found OH slots to lack adequate time to address Scrum teams' needs fully.

Overall, at salesforce.com, the need for the Office Hours program is decreasing over time. When management supported the cut of Scrum teams supported by each UX member, they also supported headcount for rapid growth of the UX team. The UX team in two years has gone from less than ten, to more than 30 full-time staff. This larger team can cover more product areas, so fewer Scrum teams are left without an assigned design resource. The Scrum teams that are not given a resource now are typically the ones that have simplier features and therefore are served well through the OH program.

2.2 Design Transformation

Fitting the complex design of enterprise features into the agile process proved to be a major challenge. Developers were frustrated that they did not have UI designs that they could start to code in the first sprint and designers felt the one month sprint model did not leave them with enough time to create and evaluate their design. Also, in waterfall, the designer could focus on the design in its entirety. In agile, the UX team was forced to take the design and break it into pieces that could be tackled by development in one month segments. This can be difficult for complex enterprise features such as analytics or workflow which are often impossible to design and evaluate as component parts.

Some of the approaches that have improved the process include:

- Moving to parallel development and design [7], [8]
- Working a release ahead
- Utilizing interactive prototypes for usability testing and communication of designs to developers
- Design Studios

Parallel Development and Design. Parallel development and design (Figure 3) is beneficial in the enterprise applications space because features are typically too big to be designed, usability tested, iterated, validated, built, translated and documented in one sprint. Sometimes single sprint design and validation is actually possible to achieve, when the feature being built is an enhancement of something that already exists. However for new features or for features that require new interface components to be designed, sometimes it is not possible to accomplish a finished design in that time period. Parallel tracks between design and development can help solve this problem (see Figure 4).

At salesforce.com, when a new feature or interface component needs to be designed, the UX team member will often choose to work in parallel with development. The UX professional will create and verify designs during the first or second sprint, while developers are working on back-end features that require little to no user interface. Designs are then handed off to development to be completed in the final sprint(s) of that release.

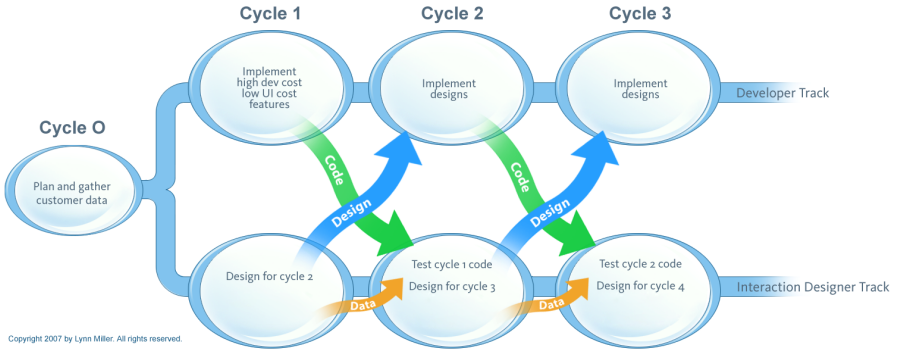

Fig. 4. Lynn Miller's depiction of parallel tracks at Alias [7]

Designing a Release Ahead. For very large scale projects that do not have a precedent in the product or for large features where the design cannot be chunked into sprints, the UX team will work on designing and iterating the prototypes an entire release ahead, while the development team is working on minor feature enhancements or features that do not require front-end design.

Interactive Prototypes. The prototypes that are created by UX become the means for expressing designs to the development teams. They essentially replace the need for written specifications. This method is not only more efficient, it is often more effective at conveying detailed interactions that are hard to describe in words and sometimes left open to interpretation.

Design Studio. We have recently had success with the Design Studio approach [9]. This process gets the entire Scrum team together for a single-day participatory design session. During this design session, all attending members are able to critique concepts and contribute to design that will be embarked on during the course of the sprint. Some key advantages are:

- Rapid exploration of design alternatives and the creation of an initial design
- The entire Scrum team, rather than the UI designer alone, owns the design and, as a result, feels very invested in the design
- Provides an opportunity for User-Centered Design education
- Provides the Scrum team an alternate way to view the functionality proposed. The resulting design from the exercise can help the team take a more realistic and informed look at whether the functionality they have proposed is feasible for the release.

2.3 Getting Usability RITE

Once the UX team has designs ready for feedback, they are swiftly taken into the lab for testing with users who represent those who will actually use that product. The UX team conducts an extraordinary amount of formative testing due to the nature of our business. Since salesforce.com is an enterprise company with millions of existing customers, we

have to be extremely diligent about making changes to existing paradigms, to ensure that these changes are worth the learning curve to the user. Also, due to salesforce.com's software-as-a-service model, all of our customers are on the same version of our product at all times. We do not have the luxury of A/B style testing (testing where multiple versions are released at the same time to see which is has the best impact). As a result, it is most effective for us to do most of our testing in a formative way, prior to release. Luckily, since we have this existing customer base, we also have a huge pool of users to pull from easily when we need to conduct testing quickly.

Since moving to agile, the UX team at salesforce.com has only on rare occasion conducted a standard usability test. For formative testing, Rapid Iterative Test and Evaluation (RITE) is almost always the method utilized to evaluate designs. RITE-style testing is based on the principle of iterating the design as you test each participant. This is in contrast to a traditional usability test, where changes are only made after a full set of participants has evaluated the design [6]. In RITE, the designer is empowered to change the interface if: (1) the participant is believed to be a representative user for that feature; (2) the problem is believed to be understood and; (3) a solution is proposed. RITE testing acknowledges that initial designs will be flawed, but that a successful design will be achieved through user input and iteration.

RITE and agile development go hand in hand. RITE is fast, iterative, inclusive of the team, collaborative, and in the end produces a prototype that acts as a proven design specification for development. No time or resources are wasted; every hour is utilized; every problem is acted upon. At salesforce.com, the UX team often makes changes after seeing one participant have a problem. Sometimes those changes are even made during the session.

In the field of usability, many researchers have a difficult time getting product teams to want to attend usability tests. Typically, by the third or fourth user, everyone, including the researcher, knows what most of the problems are, so there really is little value in continuing to attend. In RITE, the lab setting changes from a stagnant place where problems are repeatedly rediscovered, to a dynamic environment where problems are solved in real time. The lab becomes an active design space.

During UX training at salesforce.com, certain mindsets are taught to enable designers and researchers to engage in RITE successfully. Designers are encouraged to embrace the following principles:

- Collaboration: Use the lab to discuss ideas
- Focus: Don't do anything other than watch and iterate
- Flexibility: Be willing to try new ideas at a moment's notice
- No ego, no blame: Find out what's wrong, don't validate that you're right

Likewise, researchers are taught to support RITE by valuing the following:

- Collaboration: Keep attendees involved and inspire brainstorming
- Flexibility: Be willing to change your protocol to support design iteration
- Report Immediately: Do continual lightweight reporting that keeps the team iterating

Using these principles and combining that with the evolution of our prototyping tool set over time, the UX team has found a way to be able to iterate designs as fast as agile requires.

3 Success!

With the advent of these changes, UX team members have reported greater job satisfaction. The most recent agile survey showed a significant improvement in how much UX team members are enjoying their jobs (Table 1).

Table 1. A comparison of job satisfaction survey responses from UX team members in March 2008 and March 2007

Survey question and responses: Since the rollout of agile, how much fun are you having?	**% that agreed in March 2008**	**% that agreed in March 2007**
The best time	24	0
A good time	62	23
Not much fun	14	30
A terrible time	0	46

The recent survey also revealed improvements in UX team members' attitudes toward agile (Table 2).

Table 2. Comparison of survey responses from UX members in March 2008 and 2007

Survey question	**% that agreed in March 08**	**% that agreed in March 07**
Is agile making your team more effective?	73	30
Agile will maintain or improve the quality of our products.	86	46

The job satisfaction of UX members reflects the team's belief that UCD is being achieved again at salesforce.com. All customer-facing features undergo usability testing and design iteration until all serious usability issues are resolved, just as they did within waterfall. This transition from standard development to agile wasn't easy for UX, but with adaptation, the team has experienced great success.

References

1. Beck, K.: eXtreme Programming Explained: embrace change. Addison-Wesley, Reading (2000)
2. Beck, K., et al.: Manifesto for Agile Software Development (2001), `http://agilemanifesto.org`
3. Cohn, M.: Agile Estimating and Planning. Prentice Hall PTR, Englewood Cliffs (2005)

4. Fowler, M.: Put Your Process on a Diet, Software Development, v.8, 12. Expanded version (December 2000), http://martinfowler.com/articles/newMethodology.html
5. Leszek, A., Courage, C.: The Doctor is "In" – Using the Office Hours Concept to Make Limited Resources Most Effective. In: Proceedings of the Agile 2008 conference 2008 (2008)
6. Medlock, M.C., Wixon, D., Terrano, M., Romero, R., Fulton, B.: Using the RITE Method to improve products: a definition and a case study. Usability Professionals Association, Orlando, FL (July 2002), http://www.microsoft.com/downloads/details.aspx?FamilyID=3b882eb1-5f06-41d9-baba-d39ad13bc3ff&displaylang=en
7. Miller, L.: Case Study of Customer Input For a Successful Product. In: Proceedings of the Agile 2005 conference (2005), http://agileproductdesign.com/useful_papers/miller_customer_input_in_agile_projects.pdf
8. Sy, D.: Adapting Usability Investigations for Agile User-Centered Design. Journal of Usability Studies 2(3), 112–132 (2007),\ http://www.upassoc.org/upa_publications/jus/2007may/agile-ucd.html
9. Ungar, J., White, J.: Agile user centered design: enter the design studio - a case study. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, CHI 2008, Florence, Italy, April 5-10, pp. 2167–2178. ACM, New York (2008), http://doi.acm.org/10.1145/1358628.1358650