

Even though a "Dev Day" sounds like a unit of time, it cannot be associated with calendar days easily. Because resources were not dedicated and Dev Days are not strongly associated with calendar days this project **was** able to deliver on schedule without significant overtime on anyone's part.

- 1.4. Discuss the **value** of the project? How did projected benefits compare to actual (if measured) benefits?

This project was the first of its kind in the division of the firm so value was received in a number of ways in addition to product delivery.

Value as a pilot: Much of the value derived came as a result of the quest to understand how a Scrum project fit into the company's existing product development lifecycle.

- We learned that four distinct groups; Product Management, User Experience Design, Quality Assurance, and User Acceptance Testing, exist largely outside of the available window for iterative development. We also learned that there is strong interest from these groups to be included within this iterative development process. Thus, it should be possible to more completely unify these groups into cross-functional teams in new Scrum development efforts.
- We learned that our QA resource allocation model makes it difficult to bring testers into the project and next to impossible to bring in test platform hardware. These will be primary focus areas in our next Scrum project.
- We learned that we have serious environmental stability problems that must be addressed before significant test automation practices can be put into place.
- We learned that we had a broad base of support for change to a Scrum model. This support came horizontally from all functional groups and vertically from individual contributors through several layers of management.

Value as an alternative to the current product development cycle: This project was positioned as a viral messenger about the value of Scrum:

- We began with a hesitant and inexperienced customer. She now wants to do another Scrum project.
- We invited many people to the retrospectives that had never heard of Scrum. As a result, some of the Product Owners and internal customers want to know more about how Scrum delivers early business value. At least one other customer representative has requested that she be assigned to a Scrum project.

Value to the business:

- We delivered our product on time with high quality measurements.

Value to the customer:

- Service representatives are able to enter contact reasons into the system with fewer mouse clicks and keystrokes.
- System administrators are able to add new contact reasons more easily.

1.5. Discuss the project's **size**. How many people were on the project team? How were they organized into teams?

The primary team consisted of nine people organized into one team. Each member of the team was functionally highly specialized per the existing corporate development model. This specialization prevented the sharing of responsibilities to a large extent. All members of the team had at least 5 years of industry experience but almost no experience within the company.

1.6. Describe the project's **teams**. Were the teams cross-functional and self-organizing? Were the teams collocated in an open space? Were the teams physically separated within one location, or located in more than one physical location??

The team was geographically distributed across three sites in California, Texas and Nebraska. It was cross-functional and self-organizing. Despite the geo-dispersion, lack of a team-room at each site, and the impossibility of obtaining dedicated resources, we managed to maintain extremely tight and close collaboration. This was a key to our success. To do this we adopted several primary communications practices:

- A daily 15 minute "standup" conducted via teleconference.
- A synchronized and dedicated time period (1 hr) each day in which everyone on the project committed to be available and working on this project and no other.
- A group Skype chat that ran all day, every day, for immediate questions and concerns.
- A wiki that published user stories, task burn-downs, defects, release schedules, test platforms, team contact info and the like.

1.7. Tell us about the project's initiation. How was the project initiated? How was the team trained to use the Scrum process?

The initiation of the project and its initial phases were outside of the Scrum process. The "Product Council" approved the project and a customer representative was assigned. She worked with the Product Owner and User Experience Design to draft the Product Requirements Document and the User Interface storyboard. By the time a project manager (me) was assigned, this work was complete. Scrum was introduced at the beginning of the technical design phase. Care was taken to include the customer, the Product Owner, and the user interface designer into the project team despite the fact that they maintained that their work was largely complete. It was a very wise decision.

Three of the nine people on the team were previously involved in Scrum projects at other firms. All of the team was new to the company and half had never worked through one of the corporate standard development cycles. I held two one-day long training sessions in Austin and Omaha to introduce the team to Scrum concepts. This training included much CSM-like content and several hands-on exercises. That along with in-situ coaching turned out to be sufficient for us to be successful.

1.8. Discuss project **reporting**. How did you report progress to management? To customers?

We had no dedicated team room to post information radiators. Instead, we hosted a twiki on the corporate website. This twiki posted the release backlog, user stories, burn-down charts, defect and quality links, team contact information, and encompassing release calendar.

In addition, management required a brief project status to be sent to the manager of the Program Management Office once each week. This report adhered to the corporate standard format.

Our customer was part of the development team. She participated in daily Scrums and groomed and prioritized user stories with the Product Owner frequently. She was always available to the development team for questions. We reported results to her through daily Scrums, Sprint reviews, the twiki, and the weekly PMO report.

1.9. How was **change** handled? What difficulties were surfaced by Scrum that had to be resolved? How were these resolved?

The customer and the Product Owner worked together to agree on changes to the release backlog before Sprint planning meetings. Since the Sprint planning meetings focused on this release backlog, changes were largely transparent to the development team. New user stories were not injected into ongoing Sprints.

Task additions/changes/deletions to implement a user story were discussed by the team, estimated, and collectively accepted or rejected. I, as ScrumMaster, reflected the changes into the task list and task burn-down chart as decisions were reached.

We had one scope change decision that created a difficult situation for us. During normal Sprint testing we found a relatively serious defect that existed in production code. The question became whether to immediately fix the defect in production in the context of the project itself or to spin that repair off into another project context. Because fixing the defect in production would require a significant amount of unplanned additional work and infrastructure, and because the internal customer was ok with deferring into another project context, the latter course of action was adopted. (The value of customer presence in the team paid off handsomely on this one issue)

Several other change requests initiated extended discussions between engineering, the User Interface Designer, the Product Owner and the customer. In all but one instance, the customer and Product Owner ultimately decided not to change the backlog. In one instance, due to the technical difficulty of

implementing a particular user interface element, a decision was made by the team to rework the design. In the end, there was general acceptance that the modified element was not only easier to implement from a technical perspective, but also created screen elements that were easier for the service agents to use.

- 1.10. Discuss **management**. What was the previous role of the ScrumMaster? Who took on the role of Product Owner? To what degree were they successful in fulfilling their roles?

I, as ScrumMaster, am a classically trained Project Manager with PMP certification through the Project Management Institute and CST certification through Scrum Alliance. I am currently a member of the corporate PMO that (largely) implements our traditional lifecycle model. With my urging, the PMO agreed to implement this project as a Scrum Pilot. Previous to this project I was ScrumMaster on an 11 month (12/06 – 10/07) project at another firm, I managed other traditional waterfall projects, and I held past roles of software engineer and architect.

The Product Owner had previous experience with Scrum at another company but was not certified by the Scrum Alliance.

The customer in this project also played an important product ownership role. She had no Scrum experience at all and had never heard the term before my initial training session. (She is now a convert)

The Product Owner and Customer did a good job of preparing, grooming and prioritizing the backlog. The backlog was always ready for Sprint planning and both had excellent domain knowledge. Because they were always available to the project team, issues were raised and resolved quickly. I grant a large portion of our success to the commitment and ability of these two individuals.

As ScrumMaster of a first time Scrum project team, I spent a lot of time coaching, preparing burn-down charts, dealing with corporate burden, and assuming many of the petty tasks that took developers away from developing. One measure of our success was that I (the ScrumMaster) was overseas and completely incommunicado during the final Sprint, yet the team successfully delivered the product.

- 1.11. Discuss **engineering**. What environmental factors or software engineering practices had to be changed?

The post-Project retrospective identified the development environment as one of the major problems in the project. There were two major environmental problems:

First, the standard corporate lifecycle did not allow the allocation of test resources to a project until commencement of the QA phase. QA management was actually willing to work around this restriction (they did assign a QA Engineer to the project at the start), but they were not in a position to allocate test hardware to the project, because all were being utilized to test a previous release cycle. We were

able to resolve this problem because we also requested a member of the User Acceptance Test group be assigned to the project from the beginning. In the standard corporate lifecycle, UAT is separate from QA and comes into the project very late in the delivery cycle. The UAT member of our project was able to provide the needed hardware platform early for functional, regression, integration, and User Acceptance testing. Having a UAT member assigned to the team from the start also gave us a great deal of insight into build and deployment issues.

The second area where we had major problems was in build and deployment to test environments. The company's legacy development environment is so complex and convoluted that we cannot consistently build and deploy our product from day to day, even without new code commits. This issue was never resolved and caused us a great deal of grief over the course of the project. It is worth noting though, that we began dealing with these build/deploy issues on day one of the project and did manage to cobble together workable test environments during the course of development. Other concurrent (traditional lifecycle) projects didn't begin to address build/deployment until near the beginning of the QA cycle. All of those projects were late to exit development. Our project exited development on time despite the difficulties.

Another engineering practice that deviled us was the bottlenecks caused by functional specialization. This project was often gated on the completion of work by a single web developer. We simply did not have the capacity to add another web developer or address cross-training issues in this new Scrum team and still deliver our product as committed. This will certainly be the focus of a future Scrum project.

- 1.12. Tell us about **stabilization**. For how long did the software have to be stabilized before it could be released? How did you structure this stabilization process?

The company's Release Management group mandates three stabilization periods of fixed duration at the end of feature development and two subsequent integration periods. This product required little to no stabilization because it was tested and debugged during development and fully demonstrated at the end of each Sprint.

- 1.13. Discuss **success**. To what degree was the project successful? To what degree was Scrum instrumental in the project's success?

The project delivered expected capabilities on time and with few defects. The delivered features were tweaked slightly and improved as a result of stakeholder feedback received during Sprint reviews. As mentioned previously, the project was highly successful in introducing Scrum to a large group of divisional employees. Some of those employees are now requesting Scrum for their next project.

The post-project review identified collaborative aspects of the project as instrumental in success. Specifically, the presence of all functional disciplines in the team was noted as beneficial. Combined

with daily stand-ups and an always-open Skype dialog, we resolved difficult issues quickly and built a strong team dynamic.

The Scrum practice of a Sprint review kept the team focused on software that was demonstrably functional. Traditional lifecycle projects in this same company do not deliver functional code for months after project initiation. Product owners and customers were overjoyed with early working versions of the product.

- 1.14. Discuss the **Scrum Process** on this project. To what degree was the Scrum process implemented "out of the box?" To what degree did you have to modify the Scrum process for this project? For each modification, how did you formulate the modification so that the basic inspect/adapt mechanisms continued to function? What parts of Scrum couldn't be implemented, or failed, and why?

Ways in which we conformed to "out-of-box" Scrum:

We did implement most Scrum practices out of the box. Specifically we had a Release planning meeting, five two-week Sprints, Sprint planning sessions, Sprint reviews and Sprint retrospectives. Each day we conducted a virtual "daily standup" via teleconference.

We did create, groom and manage a release backlog. User stories were estimated in planning poker sessions. We selected the highest priority user stories off of this list for implementation in each Sprint. We defined acceptance tests for each user story. We measured velocity and comforted ourselves early on that our release plan was reasonable. During the Sprint, we created a task list for each user story. These tasks were estimated in hours and plotted as a task burn-down chart. This chart provided a visual picture of task completion trends for the Sprint.

The traditional Scrum roles of ScrumMaster, Product owner, and team were defined in this project. In addition we injected a customer into the team. Each role was filled with an individual that upheld the Scrum spirit of the role.

The ScrumMaster coached the team on Scrum practices, removed impediments, insulated the team from non-Scrum corporate practices, and facilitated the project.

The Product Owner and customer constructed and groomed the backlog. They were always present at Sprint planning sessions, during daily stand-ups, and during Sprint reviews and retrospectives. Their domain expertise was valuable to success.

The team itself picked which user stories to implement in each Sprint, committed to deliver, and self-organized to meet the Sprint delivery deadline.

Ways in which we did not conform to "out-of-box" Scrum:

We did not publically publish a release burn-down chart. Nevertheless, we did carefully measure Sprint velocity and use that velocity empirically to measure performance against our

release plan. We knew at the end of the first Sprint that our release plan was reasonable. We knew at the end of each successive Sprint that we were on track to finish as scheduled.

Another way we deviated from acceptable Scrum practices was that we were unable to have a complete backlog of user stories ready and a full release planning session before the first Sprint began. Instead, we pulled user stories for the first Sprint from a prioritized list of one's previously prepared, then, we completed the release backlog and release planning during the first week of the first Sprint. This did not materially affect the project because the most important stories for the first Sprint were ready for the start of the first Sprint.

Most of the practices we tried to implement that failed are more closely associated with XP than Scrum. We certainly were not successful in implementing test-first, automated unit test frameworks, or continuous integration. As mentioned, co-location was not possible, but group collaborative tools helped mitigate that problem. Because of build, deploy and platform stability issues, we were not able to QA code in the same Sprint in which code was developed. Instead, we QA'd code in the subsequent Sprint. While not ideal, this model was pretty successful in meeting our quality objectives.

2. How do you cause the accuracy of Product Backlog estimates to improve? To what degree does their accuracy matter?

Our product backlog estimates will improve the next time we estimate a project with this team. In this project, the team was new to the company, to the Scrum process, and to the concept of collectively estimating story points through planning poker.

Notwithstanding that comment, we had sufficient estimation accuracy in this project to know early on that:

- Our release plan was achievable
- We were on track at the end of each Sprint
- We needed to properly set customer expectations that the lowest priority user story would probably not make the release (and it didn't).

3. How do you ensure that what a team commits to for a Sprint is what the team actually delivers?

I will break my response into two parts. First, I will consider the situation in which a team commits to more stories than they can implement. Second, I will consider the situation in which a team does not implement the stories they do select in an acceptable manner.

The answer to both concerns lie in the ScrumMaster's ability to coach and guide as well as the team's commitment, ability and maturity.

A capable ScrumMaster should know when the team is overreaching during Sprint planning based upon the teams demonstrated ability to accurately estimate and their measured velocity. If they are poor at estimating or overly optimistic in their assessment of the points they can discharge in a Sprint, then the ScrumMaster needs to step in with reality based guidance and coaching.

The team's ability and maturity also parameterize the formula. Teams staffed with capable individuals that care and have some experience will do well. Teams that do not have such characteristics must be mentored, coached and trained. It is the ScrumMaster's responsibility to see that this happens.

For any team, whether skilled or not, it is vitally important to communicate the value and importance of the commitment, provide the information needed to assess progress, and give responsibility and authority as a group to make the necessary decisions.

Despite it all, there will be times when the team discovers that it cannot make its commitment. If this occurs, it is important to work with the team to understand the reason for the failure. A 5-why analysis during the retrospective is one of several good ways to do this. The important point is that it is the team's responsibility to make this discovery (with assistance from the ScrumMaster). It is specifically not for management to perform a root-cause analysis that is initiated external to the team.

Teams will miss commitments. The simple statistics of a normal curve require that overestimation will occur 50% of the time. It is important for the team (and management) to understand this point and to work to reduce the estimate curve's standard deviation.

Another way in which a team can commit and not deliver is to commit to a set of user stories and then deliver features that do not meet customer expectations. A capable and experienced ScrumMaster is invaluable in this situation too. The ScrumMaster is responsible for making the team aware that acceptance tests are part of each story. The ScrumMaster must see that they are collected, that they are comprehensive, and that they are detailed to the proper level of granularity. The ScrumMaster must also ensure that an appropriate verification process is in place to validate that acceptance tests are met. Working with the team, the ScrumMaster can offer guidance in the adoption of Test First practices, automated unit test frameworks, continuous integration, FIT/FITness acceptance tests and other practices to ensure that delivered products do meet customer expectations.

4. What metrics do you use to track the development process? Which metrics have been changed, removed, or newly implemented as a result of using Scrum?

In this project we tracked the development process with:

1. Story point velocity to judge progress to release
2. A burn-down chart of task hours completed to track Sprint progress
3. A prioritized list of defects to track quality objectives

4. A system of individual time reporting to determine actual time on project (because we do not have dedicated people)

Practices 1 and 2 were implemented specifically for this project. Practices 3 and 4 predated this project.

5. What type of training, resources, or tools would best help you successfully employ Scrum in the future?

We have a number of needs:

1. Executives of our company need a more complete understanding of the benefits of Scrum along with the cultural and organizational changes that are necessary for successful implementation.
2. We need fundamental CSM training for our entire organization.
3. We would benefit from experienced, independent, 3rd party coaches mentoring us through a project or two.
4. A corporate organization that allows us to co-locate cross-functional teams would help. Barring that, the purchase of tools that enable distributed collaboration will help.

6. Describe the largest impediments you have encountered and how you have resolved it (or not!)

Previously described: The build and deployment infrastructure is impossible. The issue was not resolved nor will it be in the near term. (The company does recognize the gravity of the situation though and had teams working on this). We succeeded in spite of our inability to reliably build and deploy into a test environment by addressing the problems early in the project instead of waiting until the "start" of QA.

Obtaining test hardware during the development cycle was a big impediment to success. We solved it by obtaining the hardware from the User Acceptance Testing group instead of the QA group. This solution is not viable over the long term and was a successful workaround for this project.

The company's current product development lifecycle is rigidly formal with many external approvers. This is also a problem with no realistic short-term solution. We are able to carve out about less than one half of the cycle; from just after Product Council approval of the project, to just before QA "officially" starts to run a Scrum project. Part of our success comes from the fact that out-of-window groups like QA and User Acceptance Testing are eager to participate in the Scrum development cycle.

7. Describe how you have worked with other ScrumMasters to advance the use of Scrum within an organization and within the community.

This project was the first Scrum project in my division of the company. There are other isolated Scrum projects in the company and one division is almost completely working under a Scrum model. In addition to using this project as a model to raise divisional awareness of Scrum, I have collected the names of everyone in our division that has worked on agile projects and brought them together to form a self-aware, nuclear group, of agile/Scrum proponents. I intend to use this group as a pool of talent to staff pilot Scrum projects.

I have reached out to the ScrumMasters in other divisions and discussed our common challenges with implementing Scrum in our corporate environment. I have compared metrics with them. To my surprise, in some ways, our implementation was more advanced than theirs despite the fact that they had more funded, formal training.

As mentioned above I trained roughly forty people in two one day "Introduction to Scrum" classes I administered within my company. I subsequently trained a similar number on the specific skill of writing user stories in two three-hour sessions. Word gets around, and I have now been asked to fly to the West coast to provide additional training for another organization in our company.

I teach an IT Project Management course at Austin Community College and have been asked by the Dean of their Continuing Education Program to develop an Agile Project Leadership course for the college. This course will be offered for the first time in March 2009.

I speak publically in my community often. I have given three agile oriented presentations to the Austin Chapter of the Project Management Institute and I have a coming agile talk that I will present to the Austin Chapter of the American Society for Quality in February. For a complete list of my public presentations see: <http://www.killen.org/resume/presentations.htm>. I am a public storyteller and past member of the Texas Commission of the Arts Storytelling roster (see <http://TexasTales.killen.org>) and I hold an "Advanced Communicator Silver" award from Toastmasters International. (See <http://www.killen.org/resume/toastmaster.htm>).

I founded and am President of Agile Austin (www.agileaustin.org). Under my direct leadership, in our first year and a half of existence:

- The organization has grown from 0 to 150 paid members.
- The eMail listserv has grown from 0 to 300 people.
- The Linked-In network has grown from 0 to 400 people.
- We sponsor a free workshop roughly once a month addressing agile practices such as Writing User Stories, Estimation, Release and Iteration Planning and so forth. (They always fill up)
- We sponsor a quarterly luncheon for 20 to 30 Sr. agile executives to help them network and get answers to practical problems. (Executives of Borland and BMC participate in this luncheon)
- We sponsored an Open Space conference on Agile practices

- We meet once a month with outstanding speakers discussing agile topics
- We sponsor presentations by leading thinkers like Mike Cohn, Mary Poppendieck, and Luke Hohmann)
- We negotiate and offer 10% to 20% discounts to our members for 3rd party training
- We negotiate and offer free seats in those same training classes to our members.
- We give away agile books and software to those who attend our meetings.

In addition to founding the organization, I personally managed the establishment of the workshop program, the conference, the executive luncheons, the training discounts, distinguished speakers, and book/software giveaways.

In my capacity as President of Agile Austin, I reach out to ScrumMasters, agile executives, agile coaches, agile trainers, authors and consultants to network and learn from them.

When time allows, I blog at <http://killen.org/agileforexecutives/>.

Submission Process

Thank you for your interest in renewing/applying for Certified Scrum Practitioner Certification.

- Please forward your completed form via email to the following address:
practicingcertification@scrumalliance.org.