Adding Usability Testing to an Agile Project

Gerard Meszaros
clearStream Consulting
gerard@clrstream.com

Janice Aston
Canadian Pacific Railway
Janice_Aston@cpr.ca

Abstract

Usability testing based on paper prototypes and early versions of the software were added to the agile development process for the second application release resulting in a significant reduction of usability related rework. The paper prototype became a tangible representation of the project vision that was used in many ways that contributed to the resounding success of the project.

1. Introduction

This project started off like many other development projects at Fortune 500 companies; an urgent business need was identified and the IT department was engaged to estimate the cost of filling that need. This paper describes how we deviated from the tried and accepted company practices for developing software, first by using agile methods [1], and second by introducing usability testing into said agile methods. Along the way we gained insights into what works and what doesn't and chase off a few sacred cows in the bargain.

2. Background

2.1 Company Background

Based in Calgary, Alberta, Canadian Pacific Railway (CPR) is a Class 1 North American railway providing freight transportation services over a 14,000-mile network that extends from the Port of Vancouver in Canada's west to the Port of Montreal in Canada's east, and to the U.S. industrial centers of Chicago, Newark, Philadelphia, Washington, New York City and Buffalo.

Canadian Pacific Railway was founded in 1881 to link Canada's populated centers with the vast potential of its relatively unpopulated west. This incredible engineering feat was completed on Nov.7, 1885 - six years ahead of schedule - when the last spike was driven at Craigellachie, B.C.

Consistent with over 120 years of traditional engineering project experience, most Information Technology projects at CPR embrace traditional waterfall or document-driven development methods. In an attempt to improve its IT delivery capabilities, CPR has recently started to experiment with agile methods. Usability testing has not typically been a part of the CPR culture or development process.

2.2 Project Background

The project was initiated to provide a web-service-based mechanism for exchanging rates for movement of railcars between class 1 railroad carriers. CPR already had an application that Account Managers used for building quotations; it calculates the rates for the CPR portion of the shipment route based on a set of pricing rules. That application was built using VB6 and MS Access and was considered too fragile to consider extending. As a result, we decided to build a separate application specifically for interline price quotations. The application consists of four major parts:

1) A web service that responds to Rate Request messages from other class 1 railroads
2) A user interface for building Rate Requests to be sent to other class 1 railroads
3) A user interface for maintaining the translation data between company-specific and industry standard formats.
4) A user interface for searching and displaying previously prepared quotations.

The user interface portions of the application were built as a browser-based application using ASP.Net. For the first release we chose to build the web service that received the Rate Request messages from other roads and provided the CPR rate in a Rate Reply message. We deferred construction of the user interface used by Account Managers to build Rate Requests to be sent to the other railroads until the second release because it was less urgent and seemed more complex.

We chose to reuse the existing rate database to avoid having the business manually synchronize data between the two applications. We used DTS database scripts to copy the data from the old application's database into our own database once a day. Rate administration would continue to be done using the existing application's rate admin user interface. Account Managers would continue to use the existing application to manage quotations and would only use the new application when they needed to request a rate from another railroad (in release 2.)

We made sure that the development team was collocated and largely full time. The security expert was part-time on the project and the only "subcontractor" was the outsourcer of the IT network infrastructure. The business members of the team were part-time and were located on another floor in the same building.

We built the application using the eXtreme Programming (XP) methodology modified to align with more traditional document-driven processes employed by the various internal support groups and external Interline Partners. Functional testing was conducted by the project team which included several part-time business resources. After testing within CPR, a twelve week "interline testing" phase was required with each of the other class 1 railroads.

The CPR project manager had a strong track record for delivery and wanted to try using agile practices. She engaged ClearStream Consulting to provide mentoring on project management and detailed agile development practices. ClearStream provided an "agile coach / project management mentor" and an "agile .Net development lead".

2.3 First Release Without Usability Testing

We focused the first release of the application on delivering the web service to respond to Rate Request messages from other roads. Time was of the essence so we launched the development team with a first round of user stories to start developing a rudimentary first version of the web service. Subsequent iterations introduced more and more rating rules. Some of the rules required translation of industry standard terminology for things like railroad equipment types and stations to the CPR-specific terminology used in the existing rate database. We initially chose to load these tables from spreadsheets to avoid front-end loading the project with data administration user interfaces. As business resource availability was limited and most functionality was considered to be "back-end", we elected not to solicit feedback from the end-user community.

As it turned out we required quite an extensive UI to manage the data, as well as to search for and view historical quotations. This UI was grown organically as individual user stories were developed. It was quite late in the release schedule when we were finally able to get them to try out the administration and searching/viewing user interface. Because we hadn't put a lot of thought into the overall UI design during the initial project planning phase, the end users found a significant number of usability issues that needed to be resolved. This resulted in an additional month of development that was neither planned nor budgeted. Despite this setback, we were still able to hit our "go live" date because the changes did not prevent the 12 week "interline testing" phase from starting. The start of the second release of the project was delayed because our developers were busy fixing the administration user interface.

3. Learning From Experience

3.1 Retrospectives

A key part of the agile methodology is getting feedback and making changes based on it. To this end we held iteration retrospectives [2] at the end of each two week iteration; we used these sessions to fine tune our processes. We also held release retrospectives at key points in the project. We held the first of these retrospectives after the initial development was completed; the second was held after the production installation and the third after the final software release.

One of the key concerns expressed in the initial release retrospective was the level of rework required once the business had started using the administration user interface during acceptance testing. We attributed this to the fact that we had built the application story by story without any overall vision of how the user interface should operate or what it should look like. Another conclusion was that we had started the project without any significant domain knowledge and had been finding our way by feel much like the five blind men trying to describe an elephant.

Based on this feedback we decided to engage the business Subject Matter Experts (SME's) in upfront whiteboard-based user interface design sessions. Our Agile coach had taken part in Jeff Patton's tutorial on

Agile Usability [3] at a recent agile conference and suggested we build a “paper prototype”[4] and use it to do “Wizard of Oz” testing.

3.2 Paper Prototyping

To prepare for the “Wizard of Oz” testing, we captured the results of the white boarding sessions in a more tangible and persistent form by transferring the doodles onto paper. Because we didn’t have a UI designer and we wanted to protect the development team from this “added” work, the project manager and agile coach did the work of building the prototype. Neither of us had any experience doing either UI design[1] or paper prototyping so we just made it up as we went.

We cut out screen shots from the existing application to make a backdrop on 11x17” paper. We then made up, cut out and pasted various controls onto the backdrop. Many of the screens required data grids and we found it easier to populate the grids in MS Excel and scale the grids to match the sizes of controls. We didn’t worry too much about matching the actual width of the screen so many of the pages had content narrower or wider than the “browser frame” on the backdrop.

The two of us spent a total of approximately two days preparing the prototype. This included creating a number of “pick lists” and “pop ups” that would illustrate how the application would behave when certain controls were activated. Building the paper prototype was a lot of fun. “It has been a long time since I was able to play with scissors and glue and I’ve never been paid for it before!”

4. Usability Testing

The concept of doing usability testing once an application has been built is reasonably obvious even if some of the detailed practices are less than well known. Doing usability testing on an application that hasn’t even been built yet might seem foreign to those uninitiated in the ways of User/Usage Centered Design but is, in fact, pretty easy to do. The paper prototype played a central role in this.

4.1 Wizard of Oz Testing

Once we finished building the paper prototype, we were ready to do the “Wizard of Oz” testing.[2]

Preparation for Usability Testing:

We sold the idea of usability testing to the business people on the project based on the experience in release 1. They agreed that it would be better to solicit that type of feedback earlier in the project. One of the business people took the lead in coming up with sample tasks for the participants to carry out using the paper prototype. They defined three tasks in the form of mocked up “e-mails” from customers requesting rate quotes. They also picked the users to participate in the testing. Fortunately, there was a railroading conference planned for Calgary and we were able to take advantage of having several out of town users be able to take part in the testing. We paired up the users and encouraged them to discuss what they saw so we could get better insights into their thought processes. This also allowed us to get feedback from twice as many users.

Usability Test Session

We had the business lead conduct the usability sessions with some coaching from us. He introduced the application in very broad terms (business goals, application purpose) but did not provide any detailed instructions. This was deliberate because we wanted to see how “intuitive” the design of the application was.

The development team took part in the testing in several roles. Some members acted as the “computer” by laying the various sheets of the paper prototype in front of the user in response to the user actions. We told the users that the computer was an ancient 286 PC and that response times would be much better on their own desktops! This usually elicited quite a chuckle and kept the mood of the sessions fairly light.

One member of the development team played the role of “help system”. Whenever the users asked for

[1] While we have no *formal education* in designing user interfaces we certainly have a lot of experience using them and can recognize bad user interfaces. Like a lot of users, we don’t know the rules for designing a good UI but we know what one looks like when we see it!

[2] *In the climax of The Wizard of Oz, Dorothy and her friends discover that the fearsome wizard is merely a large mechanical puppet controlled by a benevolent huckster. Dorothy's dog Toto fearlessly pulls back a curtain in the wings on stage left exposing the real "Wizard". "Pay no attention to the man behind the curtain", he roars into the microphone while setting off another series of flash pots. http://www.stopdubya.com/Editorials/12-20-04.htm*

help this person would tell them what the particular button or field did.

Other members of the development team played the role of observer [5]. They took notes on the kinds of difficulties the users encountered, the comments they made to each other and any questions they asked of the "help system".

Usability Test Follow-up

After the three 1-hour usability testing sessions were completed we brought all the subjects together for a group debrief to see if they raised any other comments or concerns in a group setting. We then compiled all the observations into a single list based on the area of the application being used.

Some of the comments resulted in defining new user stories that we added to the feature backlog at the appropriate priority; others resulted in changes to the training plans. We also provided the complete list of comments along with the planned resolution to all participants as a way to get further buy-in and to influence what they told their co-workers about the soon to be deployed application.

4.3 Results of Wizard of Oz Testing

As a result of the session we ended up adding several features that we originally considered out of scope for the project. One common theme was the need to integrate the solution into their current workflow minimizing duplicate data entry. Since this was so critical for the end-users we ended up removing some less valuable features and adding several features to improve the integration between the new application and the existing application. The usability testing helped us to validate assumptions that turned out to be erroneous. Had we proceeded on our original assumptions we would have missed the mark with our end-users and compromised business acceptance of the application.

5. Role of Paper Prototype in Planning

After the first round of usability testing was completed we posted the paper prototype as a UI story board. We linked screens together creating a virtual workflow using lines and post-it notes indicating what user action caused a transition. Then we taped up all our story cards around the appropriate UI screens as close as possible to whatever visual element they introduced or modified. This process created a checkpoint to ensure all stories in our feature list appeared in the paper prototype as well as identify missing stories from our feature list. Missing stories were added and prioritized.

5.1 Iteration Planning:

The story board became an integral part of our planning process. Prior to the Iteration Planning Meeting (IPM) we conducted business planning meetings where the business would refine features and business rules. They would start by walking thru the workflow on the story board adding additional information on the story cards. When we had enough information for a story it became a candidate for selection at our IPM. We moved the story card off the UI story board onto the Iteration Planning board, sequencing all stories in priority sequence. In the IPM the business would describe the stories allowing the development team to seek clarification as required. Occasionally the stories would be "rejected" at the IPM as further information was still required and the story card was replaced on the story board. The stories selected in the iteration were then moved to the Iteration Task Board where the developers broke down the stories into tasks to deliver the story.

During the iteration, developers and business SME's would often use the UI story board to discuss issues. Business SME's also used the story board to discuss business process changes with end-users. Through various discussions around the story board a united vision was obtained by all team members.

6. Usability Testing Phase 2

We conducted a second round of usability testing with the end-user after the core user interface stories were completed. We conducted this testing using the emerging application and it helped focus the prioritization of the user stories and fine tune the original design. We had planned for several "usability" stories to be generated from this round of testing. This ensured that we budgeted and resourced the work accordingly. We allocated about a 1.5 person-month's worth of work for these usability stories and decremented this "slush fund" as usability stories were defined based on the feedback received. The project burn down chart helped the business understand how much development time was budgeted and the business continually scrutinized all usability feedback by the amount of value to be realized from the proposed changes vs the effort required. ("No, that improvement is not worth 4 points to us so let's drop it; I'd only bother with it if it was a 1.")

7. Benefits

The value that we derived by the usability testing was far greater than anticipated.

7.1 Tangible Vision for Development Team

The paper prototype we created for the initial usability testing became our UI story board. It helped everyone understand the vision for what the project needed to achieve. It provided the "big picture" that was missing when only the "feature list" was used to communicate the requirements.

User Stories were related to the UI story board and many requirements and design session discussions gravitated to the wall that contained the story board.

7.2 Tangible Vision for End Users

The business people on the project and the end-users who were consulted or used as usability test subjects had a much better idea of what they were getting. On several occasions the business people on the project brought other end-users to the UI story board to walk them through some scenario despite that fact that it was two elevator rides away from where they were located.

7.3 Iteration Planning Mechanism

The UI story board became an integral part of our planning process but more importantly helped ensure the overall success of the project. It provided an important cross-check of the functionality described on the feature list. If something on the UI didn't have a story card related to it, we were missing a story. If a story card could not be related to the UI story board it often indicated spurious or deprioritised functionality. It was a lot more meaningful to look at than the feature list.

7.4 Increased End User Acceptance

End-user acceptance was strong as we had listened to and acted on their feedback. We received excellent feedback from end-users after the second phase went to production. One end-user (not included in the usability testing) sent us a note one hour after go-live indicating how much they LOVED the system. Given that we had worried a lot about whether the users would be willing to use the new system, this comment was very gratifying and indicated that we had wildly exceeded our own expectations.

7.5 Faster, Better, Cheaper

Overall we were able to deliver on time, under budget and exceeded our business partners' expectations; it doesn't get much better than that! Our business partners are now rabid believers in using agile methods for developing software. They cannot imagine how they could have possibly specified the system they received in a traditional document-driven development process.

8. Conclusions and Surprises

Emergent Design doesn't work very well for user interfaces. Some Design Up Front[3] seems to provide better guidance to the development team and provides earlier opportunities for feedback.

We had no trouble selling the idea of usability testing to the business people on our project. This was likely due to the pain they had experienced in the first release and may have been harder to sell before that pain had been felt. Doing the usability testing within the context of the project did not require getting permission from anyone outside the project team (e.g. the development process/tools group, the project management office, etc...)

Both paper prototyping and wizard of oz testing were easy to do despite no formal training in any aspect of it. Common sense prevailed and proved adequate.

The paper prototype proved to be useful in many more ways than just for the usability testing.

There was value in doing two different kinds of retrospectives. Biweekly iteration retrospectives gave us the ability to fine tune the process each iteration. Release retrospectives, held at major project milestones, were useful in seeing the larger trends that were occurring while still being frequent enough to effect process changes within the project. The last of these will act as the overall project retrospective and feed into the "Project Closeout Report".

The project burn down chart helped the business focus on how much work was left to do and to make the tough decisions to keep the project on time and on budget. Usability testing complemented this by ensuring that all the work was accounted for earlier in the project and it prevented last-minute essential scope creep.

[3] As opposed to BDUF: Big Design Up Front

9. Next Steps:

In order to capitalize on this success, for the 3rd phase of the project we have started to build our UI story board and to begin usability testing with end-users during the initial planning phase. We are also considering engaging actual customers in the process as well to provide feedback on the Customer Portal version of the application. This will help to ensure that we have a clear vision during the planning and estimation phase.

We are also using the UI story board to validate requirements and refine our feature list during this initial planning phase. We built a prototype based on “our” understanding of the requirements. We had our business SME’s walk through the paper prototype with typical business scenarios validating our thinking. This helped to clarify the envisioned process and identify where further clarification was required.

10. References

[1] Agile Alliance, *Agile Manifesto*, Agile Alliance, Snowbird, 2000.
[2] Kerth, Norman, *Project Retrospectives*, Dorset House, New York, 2001.
[3] Patton, Jeff, *Agile Usability*, Tutorial Notes from Agile 2005, Denver, July 2005.
[4] Snyder, Carolyn, *Paper Prototyping*, Morgan Kaufmann, San Francisco, 2003
[5] Kuniavsky, Mike, *Observing the User Experience*, Morgan Kaufmann, San Francisco, 2003