A Tale of Two Projects

Rod Coffin
Valtech
rod.coffin@valtech.com

Abstract

I had a unique opportunity to experience and analyze the adoption of agile practices on two projects at a major car rental company which, although sharing many similarities including staff, professed commitment to agile process, architecture, and programming environment, experienced two drastically different outcomes.

The first became bogged down and eventually went to production late, over-budget, with fewer features than expected, and was only reluctantly embraced by its user community. The second was ready for its first production milestone after only 9 weeks, was enthusiastically embraced by its users, and went on to become an unqualified success and a model for other software development projects at this company. In this report I will illustrate the similarities between these two projects, highlight the differences, and draw conclusions about the factors critical to the success of agile projects in large corporate environments.

1. Background

Valtech was hired by a major rental car company to help them transform their IT department to Java, Object-Oriented Programming, and Agile software development methodologies from Cobol and traditional waterfall driven development. The impetus for this was a desire to migrate away from aging and costly hardware systems, to be able respond more quickly to changing market, and to achieve higher levels of software quality. Over 3½ years facilitating this transformation Valtech has helped over 60 Cobol programmers and business analysts move to these technologies.

As a part of this enterprise-wide transformation, I was engaged as one of several Valtech consultants brought in to train and mentor a specific team of developers to learn these skills while involved in real project work. I initially functioned as a mentor on the team and eventually became the Valtech lead. Over the course of a year and a half this team developed two different systems. Each of these projects involved a mixture of classroom instruction, just in time training, and mentoring.

2. The Leasing System

2.1. Overview

The first system was to pay lease and concession (percentage of revenue) charges at all rental locations throughout North America and eventually worldwide. This project was specifically chosen by this company for the team’s first experience with Java and Agile because it was felt that this system was fairly independent from other systems involved in the company’s daily business. Notably this system was not chosen because it offered a compelling return on investment…in fact the cost savings produced by this effort did not justify the expense. This project was seen as an investment in the transformation of the enterprise.

2.2. Narrative

This development effort was jump-started with a one week long workshop in which participants engaged in analysis while receiving just-in-time training on OOAD skills. Participants in this kick-off week included developers, users, business analysts, and managers. All emerged from this experience with an initial understanding of the system and energized about the journey they were about to embark upon.

For the next roughly eight months the development team built up the system through a series of iterations of development interspersed with classroom instruction. This instruction was timed to be relevant to development currently in process. After this phase the consulting presence was reduced and the development team devoted 100% of their time towards development of the system.
Team members learned about Agile methodologies through formal classes, the project mentors, and brown-bags conducted weekly. The most influential of the books analyzed during the brown bags were “Extreme Programming Explained: Embrace Change” by Kent Beck [1] and “Agile Software Development” by Alistair Cockburn [2].

Requirement gathering was initially a whole team function, with users, business analysts, and users collaborating on the system vision and design. But after roughly six months a decision was made to have the business analyst team moderate between the users and developers. This decision was made by management for several reasons including that this was the group ultimately responsible for the system from an accounting perspective, to minimize the time required of the user community, and because they were located near the developers while the users worked in another building. The involvement of the users in the formulation of requirements was minimized from this point forward. They were most frequently consulted regarding questions of the business practices or in discussions about system defects.

Initially, stories were captured on note cards and affixed to the wall of a nearby conference room. Eventually the team felt this too unwieldy and moved to a spreadsheet which eventually became replaced through the use of an issue tracking tool to track and organize features and defects.

The team ran three week iterations that began with a planning workshop and ended with an iteration demo and retrospective. During the planning workshop the business analysts selected features to schedule into the iteration and assigned them priority. On several occasions it was remarked that the priority of individual features didn’t really matter because the system would be delivered at one time. The business analyst and developer teams participated in both of these workshops while the user team was only infrequently invited.

Every morning the team ran a short stand-up meeting where each team member discussed what they did since the last meeting, what they would do before the next, and raise any road blocks. Developers and business analysts attended these meetings but users did not.

Development progressed in this manner for quite some time before the first production release of the system occurred after approximately 15 months after project kick-off. Although the team was relatively good at meeting its iteration targets, production dates kept slipping because more and more features were “discovered” that were considered necessary before the first production release.

2.3. Outcome

The leasing system was a disappointment because it failed to meet scheduling, budgeting, and functional expectations. The initial expectation was that the system would be fully in production after 12 months of development. The leasing system was actually delivered in three primary phases: data-entry, lease payments, and concession payments. The data entry screens went to production after 15 months of development, the lease payments after 21 months, and the concession payments after 22 months. It should be noted that the pace of development and these deployment milestones were not synchronous. In other words, much of the functionality for lease and concession payments was developed when the data entry screens went to production. The total cost of the project exceeded initial expectations and the application under delivered on the user’s functional expectations for the system.

Despite its phased approach, this release schedule essentially constituted an all-or-nothing implementation. The system did not begin providing business value until the lease payments were implemented and concession payments followed very shortly thereafter. The data entry phase provided no value of its own but was seen as a prerequisite step to having all of the contracts entered that would be necessary for a complete switch-over from the existing system and the new one. In other words, real business value was not delivered until the second and third phases and even then the new system was not perceived as valuable by its users because it required them to change while providing no significant benefit.

3. The Traffic Violations System

3.1. Overview

After the first release of the leasing system I and three of the developers were pulled off the leasing system to develop the traffic violations system. This system was intended to process all citations issued on vehicles operated by the company. This project was chosen for several reasons including that the platform running the existing system was being phased out, the company desired to be able to adjust quicker to the changing marketplace (electronic toll processing, etc.), to reduce the cost of processing citations (through improved ease of use, OCR, etc.), and to reduce the amount of citations paid unnecessarily because proper
responsibility could not be determined. The last two directly contributed to a strong ROI for the project as they appreciably reduced cost. It also offered a fresh opportunity for the team to apply agile methods in the implementation of a software system and build upon their previous experiences.

3.2. Narrative

The basic method by which the traffic violation system was developed was very similar to the leasing system. But from the beginning the team was looking to build on its experience with the leasing system and revise its development processes to more effectively provide business value to the company.

Like the leasing system, development activity was structured in three week iterations beginning with a requirements workshop and ending with a system demo and retrospective. But unlike the leasing system, its users, developers, and managers all participated in these meetings.

This system was developed through the collaboration of two primary teams, simply the developers and the user community. During the development of this project the developers relocated to a conference room in the users building. Here they setup a shared project room environment with dry erase boards, static cling sheets, and an overhead projector. This was an effort to achieve higher levels of collaboration between and within teams and was in part motivated by the results of the leasing system where users did not feel actively included in requirements and design.

In the requirements workshops the users (typically represented by their supervisor and one or two end-users) set the priority for the iteration. Afterwards, the development team would estimate the prioritized features and commit to a certain amount of work for the iteration based on the amount of work accomplished in the previous iteration. During the requirements workshop the system would be explored and information captured on static clean sheets on the walls and on the team’s wiki site.

Each day developers and the user’s supervisor participated in a daily stand-up meeting similar to the one for the leasing system. Developers chose tasks to work on from the user’s list in a prioritized order and as questions arose they would ask a user to drop by and they would talk through the question, look at screens together, etc.

The traffic violations system also started with data entry screens, but these were designed in such a way that they directly fed into the live business process. They provided immediate value to the company by streamlining the way citations were entered into the business workflow. The first production release was ready after 9 weeks of development.

3.3. Outcome

The traffic violations system was a success because it surpassed scheduling and functional expectations, while still coming in on-budget. Such a success was so unexpected the company wasn’t ready and it took an extra three weeks before all the required signatory steps were in place in order to deploy the application to production. In the first two months of live production usage over 25,000 citations were entered using the new system. The application development team continued to deliver new releases to production approximately every six weeks thereafter. Every release was an exciting opportunity for the team of developers and users to provide value to the company and to improve the user’s ability to accomplish their jobs.

The success of the traffic violations system caused many people in the organization to take note and embrace the lessons learned in this project in other areas of the company. Notably management frequently invited guests (including users from the leasing system project) to the traffic violations iteration wrap-up meetings to encourage this style of development. Also, other projects teams restructured their physical arrangements into a shared project room as the traffic violations team had done.

4. Similarities

These projects make an interesting case study largely because they are more similar than not.

4.1. Staff

Both projects were staffed by developers from the same group who had received similar training. The application development team was comprised of seven developers led by a project manager. Although experience levels ranged from recent graduate to veteran developer all were new to Java and Agile. Several of the developers had extensive backgrounds in legacy technologies and the paradigm shift that this represented complicated the learning process. The entire team initially worked together on the leasing system before a subset of developers moved to develop the traffic violations system.

4.2. Methodology
Both projects employed a similar methodology, which was most closely based on XP. Notable XP practices held in common between the two projects include continuous integration, unit testing, the planning game, and coding standards. Both teams also held daily stand-up meetings and end of iterations demos. Neither project effectively automated acceptance testing or adopted pair programming.

4.3. Tools

Both teams used a wiki site for collecting notes, features, test data etc. and an issue tracking tool to track features, improvements, defects, etc. Both projects were built using a similar toolset upon a common architecture shared by many other departments throughout the company. They both also dealt with the same support organizations.

4.4. Domain

These projects were both dealing in a similar domain (car rental) although both were addressing different aspects of this domain. Both projects dealt with accounting issues instead of car rental operations and both were re-writes of an existing mainframe system.

5. Differences

In light of these many similarities the contrast of the outcomes between these two projects is striking.

5.1. Incremental Releases

Most significantly the traffic violations system adopted small, incremental releases to production. Early releases of this system were able to actively function within the processes of the business group by feeding into the existing system that was being replaced. This provided business value more quickly, reduced risk, bolstered the morale of the team, and gave the system’s users an early opportunity to use the system in a real environment.

Early on in the planning stages the team (developers and users) formulated an implementation roadmap. This roadmap laid out the sequence of releasable modules that would be developed. Releases were ordered based on business value and inter-dependencies. In fact, as the users gained experience using the new system in production, their conception of what they wanted this system to do and how they wanted it to do it changed and the plan was adjusted accordingly.

Conversely, the leasing system, although delivered in phases, actually went “live” at a single point in time. In other words, although portions of the system were deployed to the production environment ahead of time, no real processing was performed in this environment until the entire system was complete and all contracts were entered into the new system. This was the approach intentionally chosen by the user and business analysts groups. They felt that because of the nature of the system (highly interdependent) it could not be incrementally adopted into their business process. I sympathize with this concern but feel that this may have been possible had the business team been more willing to adjust their processes accordingly. The business team of this system was less willing to change their processes to accommodate a new way of doing business then the traffic violations team. I believe that this was influenced by several factors including the nature of the system, the personality of those involved, and their level of active involvement in formulating the system vision.

5.2. User Involvement

The traffic violations system more actively engaged their users in the development process and encouraged them to frequently provide feedback to the development team. Along these lines the team relocated to same facility as its users, invited the supervisor of the user group to its daily stand-up meetings, and gave the user community (represented by their supervisor) genuine control of the priority list.

Conversely, the leasing system involved its users less in the development process. Although in the early phases of this project users actively participated in requirements workshops, iteration planning activities, and wrap-up meetings their active involvement was quickly replaced with an intermediate process of a small group of business analysts facilitating the communication flow between developers and users.

But regardless of the reasons the user group began to feel increasing levels of isolation and lost a strong sense of genuine ownership in the system. This is understandable considering that as they lost input into the system being developed they stood to inherit the costs associated with implementing a new system without having input into how this system could benefit their day-to-day activities.

5.3. Shared Project Space

The traffic violations team also organized itself in such a way as to foster a “whole team” atmosphere. They chose to work in a shared project room rather than in a traditional cubicle environment as was the
case with the leasing system. The company purchased each developer a laptop for use in this environment. This project room was very quickly plastered with dry-erase cling sheets where most requirement and design discussions occurred. This room was additionally equipped with an overhead projector that facilitated prioritization efforts and gave the team a mechanism to look at code together, run end of iteration meetings, and requirements workshops.

5.4. Clear Business Objectives

The traffic violations system offered many compelling reasons to embark upon its development. Firstly, it stood to save this company hundreds of thousands of dollars per year in payments of citations that it should not have been responsible for. It also offered a strong potential to increase the productivity of the business team responsible for the payment of citations. Most notably by allowing thousands of citations that previously were manually entered to be scanned, parsed using OCR tools, and loaded directly into the system, the throughput of this organization was dramatically improved. And, perhaps most importantly, by building this system in an agile manner with an emphasis on extensibility, it could offer this company a competitive advantage by allowing it to quickly engage with vendors and municipalities to offer new services to its customers.

As mentioned previously, the leasing system offered less compelling reasons to invest in a redesign. Although the previous system offered only minimal features, and required many manual processes to supplement its shortcomings, the benefits of replacing these manual processes with a new more automated system did not justify the cost.

6. Evaluation of Outcomes

6.1. Incremental Releases

One of the key enablers for the comparative success of the traffic violations project was the willingness of its business team to change their business processes in order to accommodate incremental releases of the new system. This practice facilitated the success of the project for many reasons. Among them, it provided invaluable feedback allowing users to grow in their understanding of the system and for this understanding to translate into tangible changes to the product and development roadmap. The leasing systems had several moments late into development when fundamental assumptions were uprooted by the discovery of new requirements. These changes were embraced by the team, but their impact would have been much less had users been provided a working (albeit partial) system earlier and had the chance to validate their conception of the system.

The approach of small, frequent releases, also gave the business much greater control for managing risk. In addition to the risk of the system not meeting the true needs of the business group, technical risks were addressed early through this approach. These included tool selection and system integrations. The risk of implementing a partial system was managed by preserving the operational capabilities of the existing system. And business value was being delivered at each step of the way.

But much of the value that incremental releases provided was less tangible. The morale boost that early and frequent deliveries of software into production provided all parties involved helped to keep the project lively and progressing quickly. And the recognition by the greater company of the success of this project encouraged higher levels of cooperation from support group because everyone wants to be associated with a successful project.

Most importantly, the philosophy of small releases to production gave the prioritization efforts their value. If all of the features being prioritized go to production at the same time then there is little value in prioritizing them. But since the priorities the users of the traffic system chose translated into new software they could immediately benefit from, the prioritization activity became genuinely important.

6.2. Boldness to Change

But in order for an incremental approach to be feasible for the traffic violations project its business team had to be flexible with regards to their business processes. As often happens, this team had structured many manual processes around the system that was being replaced. Many of these processes were not intrinsically necessary, but were performed to compensate for shortcomings in the existing system. As modules of the new system went live the business users had to in turn change some of their behaviors. In a very real sense this willingness on their part to embrace change and evolve their processes alongside the new system was a large factor in the success of this project. And in fact, “embrace change” became the motto for the supervisor of the user team who saw the value of this approach and realized that embracing change was to the benefit of all involved.

This approach also required IT management to be bold enough to accept changes within their organizations as well. The most obvious change was that of physical re-location to the shared project room.
in the customer’s facility. This was a radical shift for this company. Managers were no longer physically near their teams. And this environment also challenged traditional concepts of rewarding seniority through cubicle size and location. Also, in order to provide developers the mobility necessary to work in this environment IT management purchased all the developer’s laptops so that they could work remotely.

Just as significant to these physical changes, IT management had to be flexible enough to accept less rigidly defined development plans. Instead of a traditional plan laying out a complete sequence of steps from start to finish, on both projects management accepted the concept of a high level roadmap and more detailed iteration plans that evolved through system development.

Management took these bold changes because they believed that they would lead to increased levels of productivity for the team. And specifically, they hoped to more fully engage the end-users of the system.

6.3. Collaboration

Another key differentiator for the traffic violations system was the collaborative environment between its users and developers. Users of this system felt (are indeed were) genuinely empowered to specify the type of system they desired.

As an example of this, when defects were discovered in the leasing system these were often presented as “show-stoppers” and were sometimes used as leverage to delay production implementations. Conversely, when defects were uncovered in the traffic violations system the users presented them in a forgiving manner and in some cases apologized for not having realized these quirks in the business process sooner.

7. Lessons Learned

Several lessons can be learned from these projects that can benefit other teams practicing agile software development in general and in large companies in particular.

7.1. Value Core Agile Practices

The differences between these projects largely resulted from the lack of application of a handful of core agile principles including early and continuous delivery of valuable software, collaboration between business people and developers, and welcoming changing requirements. Although these were raised as risks throughout the project, they were not valued enough by decision makers to effect real change. This case study should serve as a testament to the importance of a complete and real adoption of Agile and the consequences of not doing so.

Along these lines teams should watch for signs that a waterfall process is being superimposed onto their agile development schedule. Even if the developers are working in an agile approach, if other parties (users, managers, etc.) do not perceive the development process in the same light then you may encounter similar challenges to those documented in this paper.

7.2. Importance of Frequent Releases

Had the leasing system embraced more frequent releases then a large amount of frustration and disappointment could have been avoided. Users were frustrated at times that they weren’t receiving the system they needed to support their business processes. Developers were frustrated because whenever the users were afforded an opportunity to use the system many changes were uncovered. More frequent releases would have reduced the intensity of these interactions and uncovered change earlier in the process. This benefit could have been achieved even without production releases had users been afforded more opportunities to use development versions of the application.

The traffic system benefited from the application of this principle and as a result was able to produce a system that more closely aligned with the needs of its users, sooner, and in a more cooperative manner.

7.3. Importance of Collaboration

Many of the shortcomings of the leasing system stemmed from a breakdown in the collaborative atmosphere that was initially established. Had users been more involved throughout the project, the end result would have been a system that much more closely aligned with their actual needs. They would have had a greater sense of ownership and communication between the various groups would have been less tense.

7.4. Plan Staffing Levels to Support User Involvement

The decision to interject business analysts between the development and user groups was to some extent decided on because the users already had a full workload. Recognizing the importance of collaboration to a project’s success, we should have
allocated resources in such a manner as to make available a reasonable amount of their time to participate in the project.

7.5. Management Support

Key to the adoption of several missing agile practices in traffic violations system was the support of both IT and business management. Without their support of these changes the traffic violations system would not have been as successful as it was.

7.6. Impact of Project Selection

The traffic violations system more easily lent itself to agile methods then the leasing system. This was primarily because the nature of this system more readily lent itself to smaller production leases. This might have been possible with the leasing system, but more creativity and skill would have been required to do so. Had all teams involved had more experience with agile methods the impact of project selection might have been less pronounced.

7.7. Adopting Agile in Traditional Corporate Environments

To their credit, this company engages in an active process of observation, review, and feedback across projects. And this has been a key factor in the promotion the adoption of agile for them. The leasing system was successful in introducing this company to the concepts of agile software development. They applied the lessons they learned in this effort to the traffic violations system and the results showed how more complete adoption of agile principles could yield dramatically higher returns.

Also key to the adoption of agile at this company was that organizational change was undertaken at several levels. For example, just as the development teams on both systems engaged in weekly brown-bags, higher levels managers did so as well. This experience greatly contributed to the accepted of more radical practices in the successful traffic violations system. In more than one occasion, when management was asked to allow the develop team to deviate from company norm (for example, to relocate to their user’s site) the managers mentioned having read about such strategies in their own brown-bags.

8. References
