

Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes

Paul Hodgetts
Agile Logic, Inc.
phodgetts@agilelogic.com

Abstract

Most significant software processes involve a wide range of disciplines, from programming to testing, and from documentation to database development. Unfortunately, agile processes are typically presented from the point of view of programmers, with the other disciplines often left feeling excluded and disenfranchised.

One such discipline is that of user experience design (often abbreviated UED), a discipline encompassing several key specialties including user research, interface design, visual design and usability testing. UED activities span the full lifecycle of product development from early requirements analysis to construction and testing, spanning both large scale system issues and detailed components, with its work products forming key inputs and deliverables of many development activities.

In this experience report, I discuss my coaching experiences integrating sophisticated UED practices into the agile process initiatives of several organizations. My background is initially that of a programmer and later that of an agile process coach, and I'll explore my journey understanding UED practices and how they map to popular agile processes, mainly Scrum [1] and Extreme Programming [2]. I'll chronicle the teams' struggles to come to grips with the often programming-centric orientation of agile processes, and their ongoing efforts to integrate their UED best practices into the incremental, collaborative world of agile processes.

1. Introduction

Over the past five and a half years, as an agile coach I have had to opportunity to assist dozens of teams in adopting agile approaches on their projects. Many of my earlier experiences, like those of much of the agile community, centered on technical development teams consisting mostly of programmers. As the transition to agile processes became more

widespread, my adoption efforts moved from the single technical teams to larger teams and across entire organizations.

One of the primary challenges for agile processes as they have matured has been the incorporation of a wider range of non-programming disciplines into the core agile teams. Some of the first disciplines to receive attention were those of testing and architecture, and more recently the disciplines of database development and user experience design (often abbreviated UED) have been more deeply explored. In this experience report, I discuss my coaching experiences integrating sophisticated UED practices into the agile process initiatives of several organizations.

I began my career in software development as a programmer, and later advanced to the roles of designer, architect, team lead, and then on to management roles. Along the way, I had increasing exposure to individual UED practices, but not to the overall discipline of user experience design.

When I later became a coach for agile teams, I began to encounter several teams from organizations with sophisticated UED disciplines. I quickly realized that my understanding of the UED discipline was insufficient to effectively coach the UED practitioners through their agile transitions, and I thus began a concentrated effort to gain greater UED knowledge.

On the other hand, the UED practitioners' knowledge of their discipline was quite deep. Many of them had become UED practitioners at the start of their careers, and their practical experience applying UED theory and concepts on their real-world projects was extensive.

When their organizations began to transition to an agile approach to development, many of the UED practitioners were challenged by the new ideas and practices of agile development. Most were accustomed to practicing their UED discipline within a more sequential, phased process framework, and it was difficult to adapt to the agile frameworks' heavily collaborative, iterative and incremental approach.

The experiences described in this experience report are thus a combination of my journey understanding UED practices and mapping them to agile process frameworks, and the UED teams' journeys in integrating their existing UED organizations and practices into the new incremental and collaborative world of agile processes.

1.1. Subjects of the Experiences

Several organizations are the subjects of the experiences described in this report. These organizations are briefly described here. Unfortunately in some cases, contractual obligations prevent the identification of the specific organizations involved, or of proprietary details of their experiences.

The Internet Start-Up. From 1999 through 2002, Escrow.com, a provider of business-to-business e-commerce solutions, engaged in nearly a half-dozen significant development projects, most of which were conducted using agile processes. Many of these projects experienced notable successes [3].

Escrow.com employed a wide range of UED practices, some relatively sophisticated, on the development of their on-line platform. The UED organization within Escrow.com was not, however, highly developed, and the UED practitioners were for the most part simply members of the overall development team.

The On-Line Service Provider. Beginning in 2004, Yahoo!, one of the world's largest on-line service and portal providers, began introducing agile processes to development projects under a coordinated pilot project program.

As could be expected from the highly user-centric nature of their products, Yahoo! utilizes a wide range of very sophisticated UED practices. Yahoo! also has a very well-developed UED organization consisting of many dozens of UED practitioners, generally organized around specialized activities and a hierarchical UED management structure.

The Financial Services Provider. Beginning in 2005, a provider of advance financial services platforms transitioned their core development organization to an agile approach.

The Financial Services Provider employs a more limited range of UED practices on their projects, primarily those addressing interaction and visual design. Their UED practitioners are organized into a separate and distinct group within their overall development group, led by a thin but actively involved functional management structure.

2. The Scope of User Experience Design

As mentioned in the introduction, one of my primary coaching challenges in these experiences was to gain a sufficient understanding of the UED process and practices. The UED practitioners I met helpfully directed me to the works of Larry Constantine [4] and Alan Cooper [5]. The most helpful to me, however, was a relatively small, but significant book by Jesse James Garrett [6].

Garrett has developed a multi-layered model for describing the overall scope of UED activities (see figure 1) that provides a powerful framework for understanding the scope of the various UED practices.

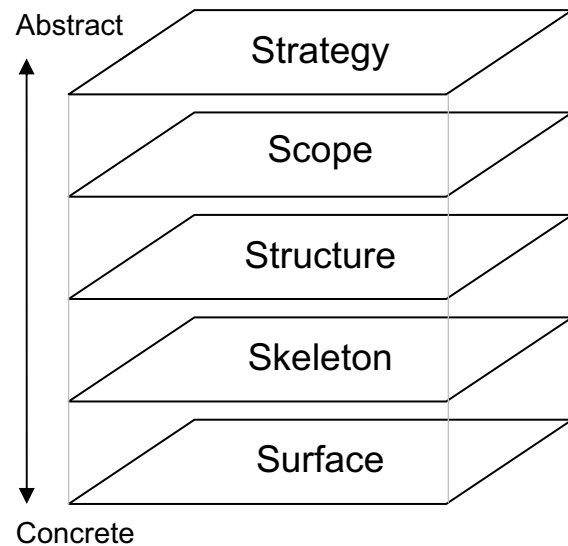


Figure 1 – The Jesse James Garrett Model. Based on the core diagram from [6].

At the most abstract level, called the “strategy plane,” we find UED practices that help define user needs and site objectives. These include usability and user research, and specific techniques such as user testing and persona development.

At the next level, the “scope plane,” we find UED practices that help define the functional specifications and site requirements. These include contributing to the writing of artifacts such as use cases.

In the middle level, the “structure plane,” are the UED practices that define the overall patterns and policies of the system, including interaction design and information architecture.

At the next level, called the “skeleton plane,” are the UED practices that define the more specific form of the system. These practices include interface design, navigation design and information design.

Finally, at the most concrete level, we find the UED practices at the “surface plane.” These define the visual presentation of the interface, and include the practices of visual design.

Interestingly, the practices of post-implementation usability testing are outside the scope this model, although Garrett discusses validating the results of these activities by gaining feedback from such testing.

Although at first I had some knowledge and expertise in a few of the common UED practices, I had little understanding of an overall conceptual model such as this. One of my greatest challenges as a coach was moving beyond my impression of UED practices as primarily concrete visual design, and appreciating those that contribute to the more abstract and fundamental definition of the system.

A useful parallel for me was to relate the “planes” of UED practices to a similar continuum of code-based practices that drive the system definition from functional needs at the most abstract, through requirements, architecture, design and code at the most concrete.

Of course, most UED practitioners are likely to be well-versed in these concepts. But since many agile coaches share a similar programming background as mine, it may be useful that as coaches we better understand the overall scope of UED activities to better coach UED practitioners through the challenges of an agile transition.

3. Organization and Culture

Making broader generalizations from a handful of experiences with UED groups can naturally be suspect. However, I found a few organizational and cultural attributes to be common among the UED groups with which I worked, and these attributes also seem common from the discussions I have had with other agile coaches. These attributes present both key challenges as well as opportunities when transitioning to an agile UED approach.

3.1. Separation and Silo Organizations

Agile processes rely heavily on the close collaboration of the core team. Many agile processes suggest the core team should include all of the necessary resources to deliver a completed product every iteration, and that these resources should collaborate very frequently.

In the cases of the On-Line Service Provider and the Financial Service Provider, there were dozens of UED practitioners organized into their own distinct

group within the overall organization. In both cases, the organizational hierarchy of UED management extended to fairly high levels, such as Senior Vice-President or Director. Although the UED practitioners were considered members of the project teams, most UED work was conducted within the UED organization, with more traditional project management practices ensuring the UED work was coordinated with the overall project.

In the case of the Internet Start-Up the UED practitioners were few and already integrated with the overall development organization. These UED practitioners worked as part of the core project team.

Agile processes are centered on an adaptive style of project management. One such need for adaptation is in adjusting the expertise of the core project team to accommodate the actual needs of each iteration. Ideally, the existing team members can provide the needed expertise (more on this below), but on occasion the composition of the team needs adjustment.

Where a separate UED organization exists, my experience is that team composition adjustments can be problematic. Approval up the management hierarchy is often required for personnel assignments and adjustments, and such approval is often delayed when the focus of the UED organization is primarily on the UED group, without the strong project team focus of agile processes.

Agile processes rely heavily on self-organizing teams, where project management is performed at a much more local level than in more traditional processes. In an organization with a strong management hierarchy, the interests of oversight and control on a project can interfere with the team’s ability to self-organize.

One such example occurred when a team was having difficulty integrating UED practices alongside the programming practices during iterations. The reaction of the UED management was to place a manager into the team to help resolve the issues, even though the manager did not actively participate in the completion of tasks for the iteration. Although the manager’s participation did help resolve some of the issues, their participation also relieved the team of some of their responsibility to learn how to self-organize, and seemed to inhibit the team’s progress towards becoming a self-reliant team.

In general, a strong separate UED organization provides a seemingly opposite and countering force to the agile project team focus we desire. While a center for UED discipline is of course important to provide the needed best practices, tools and standards for the discipline, a strong organizational and management hierarchy can be problematic.

3.2. Specialization within the UED Discipline

One of the goals of agile processes is to form a team of people who are flexible, with a general range of skills to take on a variety of tasks each iteration. This allows the team to become more productive and efficient with fewer resources.

In the case of the larger UED organizations with which I've worked, I've encountered an increasing degree of specialization among the practitioners, for example those that only perform visual design. Smaller UED organizations, likely out of necessity, often are made up of practitioners who practice a wider range of activities.

Where increasing specialization exists, I've found that teams encounter difficulties in coordinating and completing their iteration goals.

Specialization forces the team to pre-assign tasks to individuals, thus hindering the team's ability to dynamically allocate tasks on a day-to-day basis as new circumstances emerge. The pre-allocation of tasks also requires additional planning time that often extends the length of each iteration's initial planning activities.

Specialization also drives teams to break down their iteration tasks to accommodate each individual's specific practice. This can result in finer-grained tasks and more points of hand-off between individuals in order to complete an overall feature. The finer-grained tasks produce larger. More complex iteration plans, and the increased points of hand-offs create more opportunities for misunderstandings and missed work.

I have found some degree of specialization common in many organizations and projects. For example, there are fundamental differences between activities such as testing and database schema design that seem to require some degree of specific expertise. But I have observed that increasing levels of specialization within disciplines seems to create more issues than it solves. Within the UED discipline, it seems that more flexibility is usually possible.

It is interesting that I've encountered few UED practitioners in large organization who are truly only capable of performing a single type of activity. Yet the culture of the organization is such that many practitioners seem to resist taking on types of activities outside of their identified specialty. My observation is that the practitioners are often responding to the expectations of their management, which is reinforced by policies such as job categorizations and performance reviews that compel the practitioners to focus on single specialties in order to advance their careers.

4. Fitting UED Activities into Agile Iterations

Both Scrum and Extreme Programming are agile processes that strongly advocate completing the entire range of end-to-end activities for a targeted product backlog item or story within the time box of a single iteration. The most common iteration lengths vary from two weeks to a month in duration, although some teams operate on cycles as short as one week.

In my experiences both adopting agile processes and coaching teams through agile transitions, I've found challenges in completing all of the code-producing activities, from architecture through design, programming and testing, within a single iteration. I observed some key parallels between code-producing activities and UED activities that helped the teams formulate approaches for integrating the UED activities into agile iterations. These approaches are shown in figure 2, and discussed in detail below.

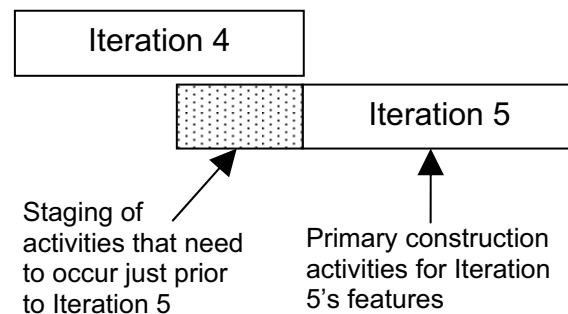


Figure 2 – UED Activities and Iterations

4.1. User Research

User research typically focuses on collecting data on the users' needs and characteristics, and produces a variety of work products from reports to prototypes to personas. This data is most often used early in the overall product lifecycle to help drive the system requirements. There are strong parallels between the UED activity of user research, and other requirements gathering activities such as RAD workshops.

In agile processes, requirements development is performed as part of developing the individual items or stories that comprise overall product plan or product backlog. Agile processes seek to perform requirements development incrementally, first producing general descriptions of features and capabilities and then progressively refining those descriptions through a series of conversations between the domain experts and the development team.

Each iteration begins with as general as a description of a story as possible, with the refinement occurring during the iteration. From my experience, however, a certain minimal level of definition is required at the start of an iteration in order to avoid delays while the development team waits for sufficient understanding of the story. This initial understanding is typically gained during a prior iteration, or in some cases before any iterations start.

From my experiences with UED groups, we have reaffirmed that user research is an important contributor to the understanding of features and stories. User research can contribute both early in the lifecycle to the initial understanding, as well as during an iteration to help refine the understanding of stories currently under development.

Many other user research practices, such as surveys and interviews, can be conducted as needed either prior to or during development, and can be integrated into agile iteration cycles to support incremental development.

One aspect of user research that was found to be difficult to integrate into agile cycles is the practice of conducting user or usability sessions and field studies. Conducting a user session often involves scheduling specialized facilities that are in high demand, as well as making appointments with the subjects of the studies sufficiently in advance, and thus these sessions typically have lead times of several weeks. Despite the best efforts of the teams, user sessions still had to be conducted at least one iteration in advance, even though some of the detailed information gained would have benefited from a more just-in-time approach.

4.1. Interaction Design and Information Architecture

Interaction design and information architecture contain elements that both contribute to the overall understanding of the system, as well as to the detailed construction of individual features.

Where the goal of a specific interaction design or information architecture activity is to produce structures or policies that affect the design of many features, the teams found it appropriate to perform a wide but shallow set of these activities just prior to constructing the first feature affected. From my experience, this is an approach similar to that I've used with larger-scale architectural issues, such as the partitioning of a system into major layers and services. In both cases, however, it is important for the team to continually gain feedback on these earlier decisions

and adjust the structures and policies based on the feedback.

Where the goal of a specific interaction design or information architecture activity is to produce structures or policies that are limited to a single feature, or perhaps to a small set of features, the teams found it effective to perform these activities as part of the development of those features during an iteration. While some natural dependencies still occurred between these design activities and the construction activities, the teams were easily able to coordinate these local dependencies within each iteration.

4.2. Interface Design and Navigation Design

Interface design and navigation design consist of practices I initially thought contributed only to the construction of features. Through discussions with some patient UED practitioners, I learned that some interface design and navigation design activities actually share many similarities with interaction design and information architecture.

Some interface design and navigation design activities are intended to produce interface structures and navigation patterns that apply to many pages or screens throughout the system. Through experience, the teams found that although these structures and patterns had to apply to every page or screen developed, it was not always necessary to design the common structures or patterns prior to constructing the first page or screen. In fact, we found that the first page or screen could serve as an effective prototype for many of the structures or patterns.

In other cases, however, we found that failing to consider some aspects of the structures or patterns across a wider range of features resulted in expensive interface refactoring in later iterations (see also the discussion on modular UI design below). Unfortunately, our experiences to date have not resulted in any solid recommendation regarding when to choose a local or broader scope for interface design and navigation design.

Where the interface design and navigation design activities produce structures or patterns that apply to only a single or limited set of features, the teams found it highly effective to limit these activities to the iteration in which the feature was scheduled for development.

4.2. Visual Design

Similar to interface design and navigation design, visual design activities may appear to be limited to

construction activities. Although most visual design activities produce designs limited to a few pages or screens in a system, some create designs that apply across many.

In the area of visual design, however I found many more techniques and tools that support the creation of decoupled interface components (see the discussion on modular UI design below). In practice, the teams found few visual design aspects that could not be adjusted more dynamically when these tools and techniques were effectively applied. Thus it was very rare for any visual design activities to be conducted outside of the iteration where the corresponding features were developed.

4.2. Estimating Creative Activities

Every UED group that I coached encountered similar difficulties planning detailed tasks for some of the UED activities within iterations. Agile processes ask the team to create a plan for each iteration consisting of all the detailed tasks they will need to perform in order to complete the development for the targeted features. Part of the planning activity involves estimating the time required to complete each task.

For many UED activities, the detailed development tasks are relatively straightforward, with a clear path to completion. However, some UED activities are highly creative, even artistic, in nature, and they have a much less certain path to completion.

As the teams explored these creative tasks, we addressed two primary questions – what it meant to be complete, and how long would it take to get there?

Creative activities often have quite fuzzy notions of their completion criteria. The creation of basic designs is often followed by an indeterminate period of refinement. The teams found it useful as a UED group to realistically discuss how much refinement was required for each type of creative task and when that refinement was needed in the product lifecycle. Through these discussions, they were often able to see when the refinement efforts were no longer producing benefits proportional to the additional effort, and where they could postpone and often batch together some final refinement efforts for later in the project when the value of the additional refinement could be more effectively compared to the value of other remaining development work.

Creative activities are also often difficult to estimate. Particularly when something very new is being conceived, some amount of “noodling around” is often required before an idea produces results. Through working with the teams, I suggested that we always time box these truly creative tasks with a fixed

estimate representing the most time they would want to waste if this task could not be completed as envisioned. During the daily status meetings, if the time box has been exhausted before the task is complete, a discussion is triggered to examine the progress that has been made and how much more time would be required. The team can use their judgment to decide if more time will be allocated, or if a different approach is needed.

5. Incremental UED Development

A key agile strategy is to incrementally develop systems. Incremental development can address both “breadth,” by progressively adding features to the product, as well as “depth,” by progressively adding sophistication and refinement to features. A common longer-term agile strategy is to opportunistically alternate between building breadth and depth, perhaps by first building a single feature in depth to gain top to bottom feedback, then building additional shallow features to gain feedback on the boundaries of the system, and then switching back to building more features in depth.

5.1. Applying UED Activities Incrementally

As discussed in section 4, UED activities such as user research and interaction design can contribute to a broader understanding of the scope of the product. These activities address a wider range of product features, and often define more abstract structures and policies.

Other UED activities, such as visual design and interface design, contribute to either a detailed understanding of specific features or to the actual construction of the product.

Each of the UED teams experienced challenges in identifying activities, or the portion of the activities, that produced broader abstract deliverables, and those that produced specific concrete deliverables. My observation was that most UED practitioners were accustomed to approaching each activity in its entirety from start to end, rather than breaking an activity into parts that could be progressively completed over time.

Through some effort, the teams were able to identify the portions of the UED activities that contribute to breadth, and those that contribute to depth. The breadth-based activities were typically associated with defining the broader understanding of either the overall product feature set or of strategies and policies governing the construction of sets of features. These activities were typically scheduled to

occur earlier in the lifecycle for the corresponding feature sets, perhaps one or more iterations prior to the actual construction of the features.

The depth-based activities were usually associated with the construction of specific features. These activities were usually scheduled to occur later in the lifecycle for the corresponding features, most often during the same iteration where the feature was scheduled for completion.

5.1. Modular User Interface Design

Section 4 alluded to the importance of modular designs as an enabling technique for incremental user interface development of smaller, targeted feature sets. This approach has strong parallels to the techniques utilized in code-producing activities, and well documented in texts discussing agile design and refactoring [7, 8]. Modular designs have been proven to produce software structures that are more easily changed to meet the evolving needs of a progressively emerging feature set.

As I learned more about the specific UED activities, I discovered that the activities of interface design and visual design have a number of well-known techniques for achieving varying levels of modular interface designs. For example, the use of CSS-based styling information allows many aspects of the appearance and even the layout of the interface to be changed without necessitating changes to the HTML code of the pages. Where the teams made effective use of these techniques, they enabled the execution of agile strategies such as the incremental design and refactoring of the user interfaces.

For larger-scale elements of the user interface, including the work products of interface design and navigation design, I found only a few specific solutions isolated to individual projects. In these cases, the UED practitioners, in conjunction with the programmers, had created software frameworks that allowed the interface and navigation elements to be altered without requiring extensive reprogramming of the underlying code. These frameworks seemed to enable the teams to “rewire” the navigation flow with a low enough cost to allow for more frequent changes during iterations.

Other than these specific framework solutions, I am not aware of more generic techniques that are being applied to larger-scale interface and navigation designs, although I propose that many of the techniques that have been applied to code-based artifacts, such as modular design, frequent small-scale refactoring and powerful tools, may apply here as well.

6. Steps towards Solving the Puzzle

The Internet Start-Up has undergone several reorganizations, merger and sales, and to the best of my knowledge its agile development organization no longer exists. The On-Line Service Provider and Financial Services Provider continue to utilize agile processes on many of their projects, although their adoption efforts are each less than a year old and must be considered as works-in-progress.

Nevertheless, each experience has generated significant successes in integrating user experience design practices with agile processes. As a result of these successes, a few key strategies have emerged:

Forming the Whole Team. It was of vital importance that the UED practitioners both viewed themselves as part of the project community and conducted their activities in support of that view. While transitioning from a more separate, silo-based organization proved difficult and painful, large benefits in productivity and efficiency were gained by doing so.

Striving for Collaboration. In addition to integrating UED practitioners into the project community, it was also extremely important to enable and encourage close collaboration between the UED practitioners and the other disciplines of the project team. Without the daily interaction between all disciplines, important gaps emerged in the overall work progress that hindered the efficient delivery of completed features.

Effective Scheduling of Activities. Perhaps one of the more controversial recommendations is to recognize those UED activities that require lead time, and to effectively schedule those that need to be staged an iteration or two ahead of the primary construction activities. While it may be possible to achieve the agile ideal of single-iteration completion, the teams and I have not yet been able to realize this within our current understanding and expertise.

Promoting Understanding of UED Activities. In several of the project teams, there existed a notable lack of understanding of the UED activities among the other disciplines on the project team. Just as I as a coach needed a deeper understanding of UED practice, fostering a greater understanding of UED practices across the entire project team has enabled teams to work more collaboratively and effectively.

Utilization of UED-Wide Patterns and Standards. Just as discipline-wide design and coding standards help agile programming teams gain greater efficiencies, similar UED-wide patterns and standards have enabled similar gains for UED teams. In the teams where such patterns and standards were lacking, the inefficiencies were noticeable, but in the case of

the Internet Service Provider the positive effects of such standards were evident [9].

7. Remaining Challenges

As previously mentioned, several of the subjects of these experiences are ongoing projects. Although many significant lessons have been learned, several challenges remain to be addressed.

Long-Running and Dependent UED Activities. Despite the described successes in staging some UED activities to occur just prior to the construction of features, some aspects of handling long-running activities such as user research sessions remain problematic. Activities that require interaction with groups outside of the project community, notably content creation, copy editing and legal reviews, remain difficult to schedule and ensure completion within the teams' iterations.

Culture and Organization. Although several teams have made significant progress integrating the UED practitioners into the project community, some aspects of integration continue to present difficulties. In particular, the UED practitioners are often physically co-located with their discipline, not with their project community. In the cases where co-location with the project community was achieved, noticeable benefits have resulted, but wider-scale co-location has met with resistance from the overall UED and company organization.

Understanding and Respect. The frequent interaction between all of the disciplines on the teams has enabled an accelerated rate of cross-training. Although this has fostered an increased level of mutual understanding and respect across the disciplines, each discipline often remains focused more on their individual interests as opposed to the overall project interests. Particularly in times of stress, team members can revert back to positions of self-interest and blame, rather than adopting a collective team approach towards resolving issues.

8. Retrospective

As I mentioned to in the introduction, my biggest challenge was to gain a greater understanding of user experience design activities to effectively map them to the frameworks of agile processes. While I feel I have come a long way in solving this challenge, and have been able to adapt and integrate both UED practices as well as agile practices, an even greater understanding may help to solve some of the difficult remaining issues. Ideally I feel should learn UED practices

sufficiently to actually practice them on real projects, but unfortunately there is often not enough time to do so under the demands of coaching agile projects.

Nevertheless, significant progress was made with the teams from the Internet Start-Up, the On-Line Service Provider and the Financial Services Provider. A great deal of this progress was due to the hard work and perseverance of the team members who, faced with the daily pressures of life in-the-trenches of an agile project, were still able to frequently take a step back and consider how their practices could be improved. More importantly, they had the courage to try new approaches even when the changes carried no guarantees for improvement.

It is my hope that others may recognize these experiences in their situations and perhaps gain some new ideas and approaches to help them integrate UED practices into their agile process efforts.

5. References

- [1] Ken Schwaber and Mike Beedle, *Agile Software Development with Scrum*, Prentice Hall, 2002.
- [2] Kent Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000.
- [3] Paul Hodgetts and Denise Phillips, "Extreme Adoption Experiences of a B2B Start-Up", *Extreme Programming Perspectives*, Addison-Wesley, 2003, pp. 355-362.
- [4] Larry Constantine and Lucy Lockwood, *Software for Use*, ACM Press, 1999.
- [5] Alan Cooper and Robert Reimann, *About Face 2.0*, Wiley, 2003.
- [6] Jesse James Garrett, *The Elements of User Experience*, AIGA/New Riders, 2003.
- [7] Robert C. Martin, *Agile Software Development*, Prentice Hall, 2003.
- [8] Martin Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.
- [9] Erin Malone, Matt Leacock and Chanel Wheeler, "Implementing a Pattern Library in the Real World: A Yahoo! Case Study," *ASIS&T IA Summit*, 2005, <http://leacock.com/patterns/>.