

# The XP Customer Role in Practice: Three Studies

Angela Martin  
Faculty of Information  
Technology  
Victoria University of  
Wellington, New Zealand  
[angela@mcs.vuw.ac.nz](mailto:angela@mcs.vuw.ac.nz)

Robert Biddle  
Human-Orientated  
Technology Laboratory  
Carleton University  
Canada  
[robert\\_biddle@carleton.ca](mailto:robert_biddle@carleton.ca)

James Noble  
Faculty of Information  
Technology  
Victoria University of  
Wellington, New Zealand  
[kjx@mcs.vuw.ac.nz](mailto:kjx@mcs.vuw.ac.nz)

## Abstract

*The Customer is the only non-developer role in eXtreme Programming (XP). The Customer's explicit responsibilities are to drive the project, providing project requirements (user stories) and quality control (acceptance testing): unfortunately the customer must also shoulder a number of implicit responsibilities including liaison with external project stakeholders, especially project funders, clients, and end users, while maintaining the trust of both the development team and the wider business. In this paper, we report on a series of case studies of the Customer role in XP projects. We have found that Customers have a pressured and stressful role, leading to issues of sustainability.*

## 1. Introduction

The Customer is the only non-developer role in eXtreme Programming (XP). The Customer's explicit responsibilities are to drive the project, providing project requirements (user stories) and quality control (acceptance testing): unfortunately the customer must also shoulder a number of implicit responsibilities including liaison with external project stakeholders, especially project funders, clients, and end users, while maintaining the trust of both the development team and the wider business.

In describing the XP perspective, Beck says:

*XP is a starting line. It asks the question, "How little can we do and still build great software?" [2]*

The customer role is critical in making decisions about "what to build", and in the minimalist philosophy of XP, the following are recommended for the customer role [2]:

- The customer is an integral part of the team and should be on-site with the team
- The customer writes user stories and then discusses each requirement directly with the programmers

- The customer is responsible for all business decisions including prioritising user story development
- The small 2-3 week iterations allow the user to evolve their requirements based on concrete working software
- The customer regularly tests the software to confirm it works as expected

XP explicitly assumes that the customer knows the domain well and is able to make decisions and as such does not provide "how-to" advice on gathering, expressing and prioritising requirements.

This paper builds on our initial research [10], which reviewed the customer role in a single project, and found that that the customers had the most pressured and stressful role in the project, requiring significantly more effort than the development team members.

In this paper we firstly recap our initial research case, Project Endeavour. We then present two further in-depth case studies that outline how the on-site customer role has been implemented. Finally we compare and contrast the three cases and present our conclusions.

## 2. Research Method

Information Systems Development (ISD) methodology researchers [7, 12] have expressed a growing concern that existing ISD methods do not meet the needs of today's business and software development environments. Studies [12, 10] in this area have begun to explore practices in natural settings in order to begin to address these issues. Given this trend, we have used interpretative in-depth case studies to explore our research questions within their natural setting, software development projects.

We used semi-structured in-depth one-on-one interviews to collect the data for this paper. Three XP projects are explored; we interviewed a total of 20 people across the three projects and in all cases have covered the spectrum of core XP roles including the customer, programmer, coach and tester. All interviews were taped and later transcribed in detail. The interviewees were

asked to validate both the transcriptions of the interview and the interpreted findings. Please see the technical report [9] for further information on the research method.

We use a number of quotes from the interviews to illustrate our findings in this paper; names have been avoided or invented to preserve anonymity.

### 3. Project Endeavour

Project Endeavour is an outsourced custom development project, the purpose of the project was to build a content management system. The project involves three organisations:

- KiwiCorp, a large New Zealand company, is the customer organisation
- DevCorp, a large consultancy, is the development vendor
- BureauCorp, a large software services company, is the infrastructure vendor

The data that forms this paper was collected in a series of interviews with both KiwiCorp and DevCorp project participants, near the completion of the project [10].

#### 3.1. Project Overview

Project Endeavour is seen as a success by both KiwiCorp and DevCorp, and part of that success was attributed to the use of an agile development method; XP. Project Endeavour had been attempted by KiwiCorp previously, but each previous attempt was unsuccessful. In fact, the KiwiCorp customer representative notes that when her manager was handed this project to ‘sort-out’:

*“We felt we could probably only do it if we used [DevCorp] ... because we had such a lot of confidence in them based on previous experience ... [later in the interview] ... we knew what they were capable of, ... they could actually deliver what they said they could ... a key value of working with [DevCorp is that] it was a joint effort”*

– Customer, KiwiCorp

The project was approximately 15 months long, and at its peak the project team had 11 full-time experienced team members.

**3.1.1. Prior Exposure to XP.** All of the team members were new to XP. DevCorp had recently used XP successfully on a similar project. However, none of the team members on the earlier XP project transitioned to this project. There was strong support in the team for adopting XP as evidenced below.

*“I’d had good reports about it in [project name], I’d read the book and the book made a lot of sense, it was common sense. [and later in the interview] I was willing to give it a go myself but the other thing was the team was willing to give it a go, if the team had pushed back, I probably would have folded”*

– Project Manager, DevCorp

The team gained an understanding of XP by reading the XP books, sharing XP experiences with other practitioners of XP and also by regularly reviewing their progress with the method throughout the project.

#### 3.2. Focusing on the Customer

**3.2.1. Who was the Customer?** The identified customer on the project was a librarian from KiwiCorp. She had extensive domain experience of both the functionality required for the system and in “getting things done” in KiwiCorp. The customer was responsible for determining what to build, ensuring it was built, and managing the outsourcing relationships. The customer was supported on this project by:

- A KiwiCorp senior manager, who assisted the customer to prioritise the requirements and to drive decisions affecting the project through at a senior management team level.
- A KiwiCorp acceptance testing team, who were responsible for testing the stories.
- A DevCorp analyst who was responsible for writing the first draft of the user stories based on the initial functional specification and conversations with the customer.

**3.2.2. What Happened?** Initially the project was a traditional waterfall project and was divided into three phases, planning (deciding what to build), development (building the application) and implementation (user acceptance testing, training and roll out). The planning phase focused on gathering requirements using standard workshop techniques and involved a series of user workshops that were attended by the business users.

At the end of the planning phase it was decided to use XP for the development phase. The requirements gathered during the planning phase were used as a basis for the XP user stories. The XP iteration process was followed, except that the customer was not always on-site with developers, but she attempted to spend 50% of her time at the same location as the developers.

The implementation phase was retained as this approach meshed with the existing practices of KiwiCorp and BureauCorp.

**3.2.3. How were the Requirements Gathered, Expressed and Prioritised?** The user stories allowed the customer to adapt or evolve the initial requirements as they watched the system develop and learnt what does and does not work and/or matter. Both the customer and development team members commented on the approach:

*“If we’d had to come up with requirements to the [nth] degree right at the beginning, I think it would have been very difficult because I hadn’t worked with intranet content management systems so a lot of the time I was sort of learning as I went and once ... I saw the bare bones thing, it was much easier to make more detailed decisions for the next level”*  
– Customer, KiwiCorp

The team did not follow the standard XP process for documentation because the user stories were based on a functional specification. KiwiCorp formally accepted the functional specification at the completion of the planning phase. No formal change control was implemented to track the changes between the user stories and the functional specification. Neither was there a formal agreement that the user stories superseded the functional specification. The DevCorp team noted this approach, particularly in an outsourcing arrangement, can result in a scope debate between the organisations [11]. However, on this project no scope debate occurred.

The customer noted the issue with user stories and communicating the requirements with conversations, was an issue of knowledge transfer. For example, the end-user training material development required an understanding of the system typically obtained from the functional specification, but the functional specification no longer reflected the system functionality. No documentation existed that specified the system to the level of detail required by the trainers. Cockburn [4] discusses this issue and suggests it is best to determine a need for the documentation rather than simply producing it. This suggestion indicates new adopters of XP should consider their knowledge transfer requirements (conversations or documentation) and plan accordingly.

The other change to the process was that the customer did not write the user stories and instead the stories were based on the functional specification. The project manager emphasised the time it took the development team to write user stories. He went on to discuss the potential impact of relying on the customer to create the stories:

*“I know strictly speaking that’s the clients job if you read the books but I actually did that to sort of streamline the process a little ... I don’t think the client would ... have had enough stories prepared on the day iteration*

*kicked off so we’d have delayed start[ing] the iteration and I think ... delay[ing] the iteration is [a] slippery slope”*  
– Project Manager, DevCorp

The project manager did not clarify his concern further. Possible interpretations include:

- The time available to the customer *on this project* was insufficient to include this task which is supported by the customer’s comment that she needed more time to effectively contribute to requirements development and testing
- The difficulty maybe the technical skills required to write usable user stories for developers as suggested by Beck & Fowler [3]

It is interesting to note that no one interviewed noted that the development team writing the user stories and the customer reviewing and confirming the stories was an issue or deficiency in the process.

Finally, one of the key roles of a customer is to make business priority decisions about what functionality is delivered when. During the interview the customer demonstrated her strength of feeling regarding prioritisation:

***I hate it, I hate that word prioritise***  
*[and later in the interview]... Usually I’ve done [the prioritisation] myself and in conjunction with [my manager], when **I can’t bear the responsibility** anymore*  
– Customer, KiwiCorp

The customer also learnt the importance of not assuming all functionality would be delivered on the high level plan, and so in hindsight should have started prioritising from the start of the process:

*“We thought we’d bitten off what we could chew but we worked out that our eyes are bigger than our stomachs – it works for IT too ... so I guess one way of doing it would have been to ... do the prioritising right at the time – what can we actually leave till last? And actually leave that till last.”*  
– Customer, KiwiCorp

This point concurs with the XP philosophy of delivering functionality in business priority order. However, this study also suggests that it is a learning process for the customer.

**3.2.4. How Sustainable is the Customer Role?** The customer was in no doubt about her preference to use XP in the future:

*“Overall – I love this approach to development and I’d certainly like to use it again in any future projects I am involved in”*  
– Customer, KiwiCorp

However, when considering the sustainability of the role she notes:

*“I think we needed some extra roles basically. We probably needed about three of me ... it's been my life for about a year ... look at these grey hairs ...”*  
– Customer, KiwiCorp

Although the customer would have liked to devote 100% of her time to being the on-site customer and working with the programmers, she also needed to work with the end-users and business stakeholders in KiwiCorp. Representing thousands of end users is a significant task and the time commitments should not be minimised, neither should the importance of being on-site with these users. Finally approximately 50 % of her time was spent managing technical integration issues between DevCorp and BureauCorp.

### 3.3. Summary

Project Endeavour is a successful implementation of XP, in an outsourced environment, where both organisations are keen to use XP on their next project. This project is of the recommended size and the spirit of XP was retained, although the implementation of some practices varied.

The customer on this project was responsible for all business decisions, including the prioritisation decisions. Although she was not located on-site with the development team, she spent approximately 33% of her time with the team, and did not rely solely on documents to communicate the requirements. The customer recognised more time with the team, allowing them the opportunity to just pop over and chat whenever they needed to, would have been ideal. However she, in her customer role, also needed to work directly with the users and managers she was representing in her organisation as well as managing the two outsourced vendors on this project. Finally the customer did not write the user stories, a DevCorp analyst always wrote the first draft, from that point on the customer modified the story.

One of the key issues uncovered in this case was the sustainability of the on-site customer role.

## 4. Project Discovery

This section tells the story of Project Discovery, a project undertaken by RavenCorp. RavenCorp is a start-up company based in the United States, specialising in the development of scientific programs.

### 4.1. Project Overview

RavenCorp started in mid 2000, with the intention to move a piece of scientific software developed in an academic environment into a commercial environment; this software is codenamed Discovery. Discovery will be their key product and at the time of the interviews was still under development with a team of eight staff.

Discovery, a piece of scientific software, is comprised of two main components, a complex graphical user interface and a set of complex scientific computational algorithms. The software product is due to be released in the first quarter of 2004 and at the time of the interviews was on track for completion. Initial external client feedback, based on product demonstrations, is positive.

**4.1.1. Prior Exposure to XP.** The founders of RavenCorp were not software developers, and although they had significant experience in developing simulation based programs for scientific research, they had very little formal software background or experience. RavenCorp recognised, as the focus moved from proving the concept to building a commercial product, that a software development process would be beneficial to the development of a quality commercial product.

At the end of the first quarter of 2002 RavenCorp decided to move to XP. This decision was made by the founding members of the organisation. RavenCorp followed what they considered to be best practice guidelines in moving to XP, including bringing an experienced XP coach full-time onto the project. However, it soon became apparent that a few of the existing developers could not work in XP and as a result they left the company.

**4.1.2. The Environment.** RavenCorp created a specific XP working area for Project Discovery, bringing all of the team members together in one area. All of the team also retained their individual offices as well as the communal working area; however, the team seldom use their offices. The environment is a friendly environment with a pleasant buzz and has a traffic light to show the state of the current build, a wall given over to progress charts and story cards, and a collection of toys for the programmers. A project wiki also exists which provides external visibility as well as a virtual communal workspace.

RavenCorp senior management works closely with the project team and as such, a trusted and productive working environment appears to exist.

### 4.2. Focussing on the Customer

**4.2.1. Who was the Customer?** The identified customer on the project, at the time of the interviews, was one of the founding members of the company and is a scientist. He

had been involved in developing the original academic software and as a result had a close professional relationship with the founding directors of RavenCorp.

The customer has been on the project, full-time, since the founding of the company four years ago, and throughout this period he has had the role of the customer. He has also had the role of programmer, particularly within the first two years of product development. Although he no longer has the role of programmer, he has, however, become the development manager within the company.

The customer was supported in his role by:

- One of the founding directors of the company, who was responsible for the initial vision of the product and working with potential buyers to better understand their requirements.
- A scientist with a similar background to the customer, who was responsible for using Discovery to produce product demonstrations for potential buyers. This responsibility ensured he tested the system as an end-user on realistic production scenarios and provided this feedback to the project team.
- A small team of research scientists, who were responsible for researching the patented computational algorithms that form the back-end of the product.
- An experienced software tester, who implemented an automated acceptance testing environment and worked with the customer and the research scientists to develop a suite of acceptance tests.

Finally it should be noted that until recently he has shared the role of customer with another founding scientist, who has recently left the company. They worked together closely and each had key areas of functionality they were responsible for.

**4.2.2. What Happened?** The product development cycle at RavenCorp can be divided into three stages, planning the product, evolving the product and releasing the product. This section covers the activities that occurred at each stage.

*Planning the Product.* Discovery is a new product, there is no existing product to replicate and much of the product is based on cutting-edge research discoveries made by RavenCorp scientists. Planning the full scope of the product at the start of the project was not feasible or potentially even desirable. For the first three years, including the first year of the team using XP, very little long-term release planning occurred, instead the team focused on evolving the product based on their discoveries, iteration-by-iteration.

The customer over the last year has spent a significant portion of his time planning the scope of the product for a 2004 release date. Although there is a significant portion

of discovery in this project, the release planning could have occurred earlier in the process, the customer noted:

*“In the early days ... there was always too much to do ... we were basically working to have enough stories for the next iteration ... let alone have enough stories where you could actually have sort of a larger view ... [of] a few iterations or a release.”*

– Customer, RavenCorp

His solution was to work the release planning process in parallel to the iteration planning process, rather than incorporating a small intense release planning process into the team’s schedule. The customer works on gathering and expressing the release user stories and publishing them to the project wiki. He holds regular one hour sessions twice a week with the programming team to review and estimate the release user stories. This information allows him to review and prioritise the scope of the release with the RavenCorp stakeholders. This approach has meant that over the last year he has managed to slowly “get-ahead” and establish a release plan. His intimate knowledge of the domain, as a research scientist, has potentially been a factor in the success of this strategy.

*Evolving the Product.* Initially the team followed a vanilla XP iteration process, with the customer deciding on and writing the iteration stories a couple of days before the iteration planning meeting. The release planning process has allowed this process to evolve so that the time spent in iteration planning meetings has been drastically reduced:

*“In the early days the meetings were... like four hour meetings and they just dragged on and I think everyone would lose their patience at a certain point”*

– Customer, RavenCorp

The small regular meetings allow the customer and programmers to discuss the stories in small batches, resulting in the iteration planning meeting simply being a refresher discussion, followed by the tasking process.

One of the other variations to the process was the involvement of the research scientists in the acceptance testing process within each iteration. The research scientists developed the algorithms in the product and were in the best position to accept the implemented algorithm. An automated acceptance-testing framework built by the team allowed the research scientists to write tests to test their algorithms. All other aspects of the vanilla XP process are followed by RavenCorp.

*Releasing the Product.* Discovery, at the time of the interviews, was still some way off being released as a shrink-wrap product. However, RavenCorp did release the product regularly, every 2-3 iterations, to the internal RavenCorp stakeholders, including the end-user

representative and the founding director. These releases allowed the team to receive end-user feedback, as the end-user representative used the release to create product demonstrations.

**4.2.3. How were the Requirements Gathered, Expressed and Prioritised?** The customer is an expert in his domain, and he is supported by a team of RavenCorp domain experts, so determining what to build has not been the difficulty for the customer:

*“We are building a scientific application and my domain experience is such that I understand what the application is supposed to do ... [and later] I think that for me, coming up with the ideas for “what we need to get done” or “what this story should be about” ... [has] not been too challenging for me ... [however] one of the things that has been frustrating for me is how detailed to make the stories”*

– Customer, RavenCorp

He has worked with the programmers for the last two years to evolve the level of detail that he places in the user stories. However, he notes the appropriate level of detail is still a concern when writing the story. He struggles with finding the level of detail and decomposition that ensures the programmer is able to engage and understand the purpose of each story, while ensuring the customer does not make design decisions in how he decomposes the functionality. Finally, one aspect of story writing he has confirmed is critical to the quality of the product is the inclusion of exceptions and error conditions for each story.

The customer noted that one prioritisation issue the team encountered, in hindsight, was spending too much time in the early iterations on infrastructure functionality rather than implementing business functionality. It is tempting to get a solid technical foundation under the team prior to moving onto business functions. Delivering business functionally quickly, however, engages the stakeholders in the process early to shape the product. It also ensures, when the hard trade-off decisions are being made later in the project, that business functionality takes precedence over the technical infrastructure.

**4.2.4. How Sustainable is the Customer Role?** One of the ways to assess the sustainability of the customer role, is the simplistic measure of how many hours the customer is working:

*“For the entire time I have been at the company ... I’ve always worked at least 70, probably more on an average of 80 [hours a week] ... I don’t even mind it, its like what I do ... I don’t worry so much about how much*

*time I’m putting in because it doesn’t even really register for me, but yeah, all these different responsibilities can be a little bit wearing.”*

– Customer, RavenCorp

The level of domain understanding, and potentially the personal attributes, required to be the customer, also make the customer a valuable resource for the company. As a result the customer is too valuable a resource for the company to be able to simply focus on the product:

*“... so it’s really been too much responsibility ... on myself and [the other customer] ... I think that was a large contributor to the problems we have had in the past ... we weren’t getting the stories written up at a fast enough pace ... well I’ve got to write the stories and think about product planning but I’ve [also] got to manage the team and I’ve got to think about this and that ... that’s just too much to deal with ... balancing all those things can be difficult, [it would] definitely be nice to be able to just focus on one thing like just focus on the product”*

– Customer, RavenCorp

The customer also noted that part of the time commitment is due to the interactive nature of the on-site customer role:

*“It may take you an entire afternoon to do something that’s like an hour of work because you’ve been interrupted and had you know 20 minute discussions a number of times through the afternoon with the developers and its all valuable stuff ... then you look back and you’re like man did I get anything done today ... so it is a bit deceiving in that respect I think”*

– Customer, RavenCorp

So given the interactive nature of the customer role, the customer typically structures his day so that during the normal 40 hour working week he is available to the programmers, and the rest of his work occurs outside of these hours. Interestingly, this means that many of the development team, including the coach, may not be aware of the workload on the customer.

## 4.3. Summary

XP has worked well with the “research and development” nature of Discovery. In fact, Project Discovery is another example of a successful implementation of XP. Once again, the project falls within the recommended size of projects and the team has

implemented the values, principles and practices with little variation from the standard XP outlined by Beck [2].

The customer is available on-site to the team, and in fact prioritises this interaction above his other duties. He works with the key stakeholders in order to provide a single voice to the development team, and be in a position to make all of the required business decisions. He undertakes all of the typical customer tasks including writing stories and testing stories, although he has some help with the later task.

One non-typical aspect with this customer is that he has at times also held a programmer role on the project. This dual role implementation blurs the boundaries of the business and technical decision making process. However, it seems likely that other XP projects may encounter this need, particularly in start-up organisations and also products where the final users of the product are programmers.

The RavenCorp product manager fulfils all of the requirements of the on-site customer role. However, the time commitments required of him to fulfil his on-site customer role requirements may not be sustainable for other product managers in a similar situation.

## 5. Project Atlantis

This section tells the story of Project Atlantis, a successful project undertaken by EagleCorp. EagleCorp is a software company in the United States that specialises in products that support application development and application development management.

### 5.1. Project Overview

The project started in mid 2002, with the intent of creating the next-generation of EagleCorp's flagship product, Atlantis 7.0. Atlantis 7.0 would be a complete re-development of the existing product, Atlantis 6.x, and would move the product onto a new technical architecture.

The interviews to collect the data for this case study occurred just as Atlantis 7.0 was moving into internal QA. At the time the project had been underway for approximately 18 months and had 16 experienced full time staff. All indicators were that the project was considered a success, particularly with regard to the quality of the product.

**5.1.1. Prior Exposure to XP.** EagleCorp had not used XP previously. The decision to move to XP was made after the architecture and the product feature set had been defined, approximately 6 months into the project. EagleCorp followed what they considered to be best

practice guidelines in moving to XP, including; the use of a local XP mentoring company, for both initial training and ongoing advice.

EagleCorp attributes the success of Project Atlantis, at least in part, to the predictability provided by XP. Firstly, the predictability of dates. The two-weekly iteration cycle allowed EagleCorp to plan and execute their development significantly more accurately than they had in the past. Secondly, the predictability of quality. The quality focus provided by test-first development and rapid customer acceptance testing have enabled EagleCorp to increase the quality of Atlantis 7.0. Finally the predictability of flexibility, otherwise known as "embracing change":

*"We as a business [would] be foolish if we said; no we can't take any new things. [XP enables us to ensure] what we are releasing is going to meet our customer's needs as of today, not ... what they were several months ago"*

*– VP of Product Development, EagleCorp*

**5.1.2. The Environment.** EagleCorp created a specific XP joint working area for Project Atlantis. The managers moved out of their offices and the programmers moved out of their cubicles into this collaborative working area. The working area contained desks on wheels and whiteboards. The desk on wheels allowed the programmers to personalise their working environment as well as allowing the programmers to easily re-orientate their work area to work collaboratively in pairs. The EagleCorp executive team endorsed these changes, which were not standard EagleCorp practice, as well as supporting these changes financially.

Another aspect of the environment was that the responsibility for the product definition, quality and implementation was clearly divided across three line managers; a product manager, a quality assurance manager and an engineering manager respectively. It was also evident that each of these managers was trusted by the executive team. For example, when asked about the importance of his buy-in to XP succeeding, the Vice President of Product Development responded:

*"To me the buy-in was easy because of the trust relationship that I've built up with [the engineering manager], I've seen how he works, I've seen how he attacks problems. Him coming to me and saying, okay, we want to use XP for this, that's almost good enough for me."*

*– VP of Product Development, EagleCorp*

This trust was also placed in each team member by the direct line managers. For example, no-one was placed on the team who was not acceptable to each team member. Any team member was able to state they would not work

with the proposed individual and no reasons were required. A strong team culture was in evidence, particularly within the engineering component of the team. A standard practice was to all head out together for lunch at around 12:30, other less regular social events also occurred, such as trips to the movies.

## 5.2. Focusing on the Customer

**5.2.1. Who was the Customer?** The identified customer on the project, at the time of the interviews, was a product manager. The product manager had over 15 years of software development experience covering a spectrum of roles including, programming, application support, testing and product management. The majority of this experience is in the same area as Atlantis, that is, products that support developers and development managers. The product manager had only been with EagleCorp approximately 12 months prior to joining the Atlantis team.

The product manager was responsible for the product definition and vision as well as the detailed functional requirements. He, however, was supported on this project by:

- An experienced *user interface designer*, who developed the user interface guidelines and mock-ups. A decision was made by the senior management in EagleCorp to add an experienced user interface designer to the project as one of key improvements to Atlantis 7.0 was to be the user interface.
- An experienced *technical architect* who was responsible for all non-functional requirements, including design and code quality. He also played the role of programmer when required.
- An experienced *technical writer* who was responsible for writing the user guide.
- An experienced Quality Assurance (QA) team who was responsible for the final full system and integration testing of the product.

Finally it should be noted that at the start of the project there was a product management team, consisting of 4 product managers. Approximately 4 months after the decision to move to XP, this team was reduced to 1 product manager as part of an organisation-wide staff cutback.

**5.2.2. What Happened?** The product development cycle at EagleCorp can be divided into three stages, planning the product, evolving the product and releasing the product. This section covers the activities that occurred at each stage.

**Planning the Product.** The first activity that occurred on this project, for the product manager, was the definition of the product. The outcome of this phase was a product

plan; an agreed and prioritised set of functionality for Atlantis 7.0. Running in parallel to this activity was a technical stream, focused on deciding and proving a new technical architecture.

Once the product plan was agreed and the architecture defined, EagleCorp decided to move to XP. The team had an XP release plan, courtesy of this initial work, and so immediately began XP at the iteration planning stage.

Approximately four months into the twelve month development process, this release plan was revisited. At this point enough data had been gathered to recognise that the original release plan was unachievable. EagleCorp needed to make a decision to cut scope or increase the timeframe. To assist this decision, the team stopped the XP iterations in order to assess and estimate the remaining functionality, this process took approximately a week. The EagleCorp executive team chose an option that involved both cutting scope and increasing the timeframe.

**Evolving the Product.** Over the course of the project the iteration process evolved. The resulting refined process meant that for the product manager and the user interface designer, the two-week iteration started approximately a week before the iteration planning meeting. During that week the focus was on developing the user stories and user interface mock-ups to present at the iteration meeting.

At the iteration planning meeting, the product manager would walk through all of the user stories and user interface mock-ups with the programmers and elaborate as required to meet the programmers' questions. Once the programmers felt comfortable that they understood the user stories enough to estimate them, the programmers estimated all of the user stories. Finally the group agreed the scope of the iteration based on the product manager's prioritisation decisions and the previous iteration's velocity.

During the next two weeks, the programmers would implement the stories. If at any point they required clarification of the user story they would discuss the user story with the product manager for functional clarifications, or the user interface designer for user interface clarifications. At the end of each day the product manager, user interface designer and engineering manager would review and test each of the completed user stories. Accepted stories would be marked as completed; the engineering manager would discuss the remaining stories with the programmer pair.

At the next iteration planning meeting the previous iteration would be reviewed for any issues and the process would be evolved.

**Releasing the Product.** EagleCorp has a Quality Assurance (QA) team that focus on full integration testing of the software products. The QA team was overloaded

throughout the development process with Atlantis 6.x testing commitments, and as such was only brought into the process near the end of development. As a result the QA team was not present at either the iteration planning or daily acceptance meetings, and missed these undocumented “conversations”, which has increased the difficulty of their job. However, the QA team is pleased with the initial quality of the product and is keen to be involved in the “conversations” throughout the next release and adapt their QA processes to XP.

Another aspect of releasing a product is the development of user documentation. The technical writer noted the XP process significantly aided her task. She was able to write the user manual based on both the user stories and working software. In a typical waterfall process she must write a user guide based on a specification that is quickly out of date. She then faces a significant rework phase once a working product is available for her to explore, typically only a week or two prior to shipping the product.

**5.2.3. How were the Requirements Gathered, Expressed and Prioritised?** As outlined in the previous section, the first step facing the product managers was to define the Atlantis 7.0 feature set. The inputs into determining the release feature set included the existing product functionality as well as specific customer requests. The product manager described this activity that took a few months to complete:

*“We [did] a lot of cross-functional, cross-decision making meetings. We talked to the executive team, we [worked] with the rest of the organisation to define requirements ... we had ... broad-brush organisational agreement [to the prioritised feature set].”*  
– Product Manager, EagleCorp

The resulting feature set was a product plan, and it listed approximately 50 features with an associated priority of 1, 2 or 3. Priority 1 features were to form Atlantis 7.0, with the remaining features to be implemented in later versions. Atlantis 7.0 included some new features but did not include all of the features of Atlantis 6.x.

Once the product plan had been agreed, the product managers turned their attention to developing the user stories:

*“One thing that helped us [write good user stories], is that [around] the same time [we started writing user stories we] started with [the] persona driven approach ... from our UI designer, so we ... had hypothetical users of the system, so we [insisted that all users’ stories] must be based on these three ... identified ... primary users. So that’s helped us [to move from] I want this and that*

*feature, [to explaining] what this person wants [to do in the system]”*  
– Engineering Manager & Coach, EagleCorp

The “persona” approach stems from a method for user interface design suggested by Cooper [5]. The personas of the three primary users of Atlantis were based on market and sales research, and included details such as the end-user’s name, age and job role. This technique ensured the prioritisation process focussed on the features required by the primary personas. This technique also resulted in a good size piece of concrete functionality for a programmer to implement that could be tested, although this outcome was not immediate, but evolved as the product manager and programmers worked together.

In parallel to the writing of user stories was the development of the user interface mock-ups. The user interface designer developed a user interface guideline document that included detailed items such as the space between fields, as well as a high level picture of the user interface metaphor. The user interface designer then worked with the product manager, defining the user interface mock-ups for the user stories within the current iteration.

In the above description it may appear as if the product manager worked in a vacuum. However, the product manager throughout this process interacted with a mixture of internal and external stakeholders.

In the case of internal stakeholders, such as the sales department, application support department and executive team, the product manager presented the product as it evolved in a regular monthly meeting. This meeting met a dual purpose of both reporting progress as well as providing a venue for the internal organisation members to evolve and/or re-prioritise functions.

In the case of external stakeholders, EagleCorp established a Customer Advisory Council (CAC), comprised of their key set of accounts. The CAC typically meet once a quarter. Two of these meetings will be face-to face meetings over a 23 day period, and the other two meetings will be 1-2 hour “webinars”. Important to these meetings is the emphasis on two-way communication:

*“...for me, from a PM perspective, [these sessions are] very valuable because [they] help me understand what some [of our] key customers are doing, some of the key issues that they have ... which parts of the application [they] don’t ... use, which parts of the application need major changes before [they] could find them valuable ... [although not all of our customers are represented] it helps me get a very good understanding of ... where some of the real major holes are and how I should ... spend my engineering*

*dollars [to build] the next generation [of] tools”*

*– Product Manager, EagleCorp*

The ability to present Atlantis 7.0 as it currently stands at each of these meetings, as well as part of an on-site customer demonstration, greatly enhances EagleCorp’s ability to obtain concrete feedback on the product direction from the end-users, thus completing the two way communication loop with Atlantis external end-users.

**5.2.4. How Sustainable is the Customer Role?** There is a consensus in the team, from all perspectives including senior management, programmers and the customer, that the on-site customer practice is a drastic improvement over the traditional document-centric processes. However, the changes that have caused this improvement have also had some unintended, and perhaps unrecognised, impacts:

*The iteration processes are quite exhausting. Every two weeks you’re basically doing another product release. ... So it’s a very, very intense process to keep up and maintain... [From] the first year of doing it I think it’s worked really well but, from a longevity perspective, I don’t know how long it will be possible to keep [this pace] up*

*– Product Manager, EagleCorp*

The product manager goes on to compare the workload to that of a traditional process, where a document is “handed over the wall”. In the traditional process although there are “peaks” there are corresponding “troughs”, when the responsibility is passed to the development team. In XP, the consistent pace results in no “troughs” for the customer. Not only are there no troughs, but the type of work is significantly more intense than writing a document:

*“... You know, you can work 40 hours on a document or you can spend four hours in an intense meeting where you are getting hammered with questions, you know, you’re getting issues you’re having to face and deal with in a four hour meeting and feel more tired from the four hour meeting than the 40 hours of document development. So it creates a different kind of pressure I think on the PM. It makes ... again the iteration meetings and the requirements to deliver every time you go into these iteration meetings, ups the pressure, ups the intensity and just being prepared for these things is the key. So, you know, you’re also acutely aware of the fact that you have this large group of people who are sitting in this meeting costing the organisation a lot of*

*money. If you haven’t got it right, something isn’t thought through correctly...”*

*– Product Manager, EagleCorp*

So although there has been a significant improvement as the result of the on-site customer role, the practice of sustainable pace has been sacrificed.

### 5.3. Summary

Project Atlantis is an almost perfect example of an XP implementation, it is the recommended size and it retains the spirit of XP, its values and principles, as well as implementing the practices of XP.

The product manager fulfils the role of on-site customer. He worked on-site with the development team, wrote the user stories, accepted the user stories, and worked with both internal and external stakeholders to ensure he was able to make the business decisions for the project. The small iterations allowed the product manager to use the working software to both report progress to internal stakeholders as well as obtain feedback and evolve the product with both internal and external stakeholders. EagleCorp tailored XP in three significant ways that impacted the XP customer role.

The first significant modification was the introduction of user-interface design principles into the process. Additional documentation was introduced, including a user-interface guideline as well as mock-ups of all of the screens. Finally, all of the user-stories were modified to implement the persona user interface technique, which they discovered had the pleasant side effect of providing a good decomposition technique for user-stories.

The second significant modification was the introduction of supporting roles for the customer, including the user-interface designer, a technical writer, an acceptance team and an architect. All of these specialists assisted the customer to define and refine the product based on their specialist skills.

The third significant modification was the introduction of structured techniques to manage the interaction with internal and external stakeholders. These techniques included regular meetings ranging from “webinars” to “face-to-face” meetings.

However, despite the success of the project and the initiatives to support the customer, there remains a concern over the long-term sustainability of the on-site customer role.

## 6. Discussion

In the three studies presented the project characteristics vary, but we have found that the customer role is consistently under significantly more pressure than the developers or other participants in the project.

The commercial settings vary. Endeavour was a new internal management system, outsourced by a large established company, Discovery being a new scientific system, developed as a product by a start-up company, and Atlantis being a new version of an IT development system by an established company in that area.

The alterations made to XP also vary. Endeavour had already done some work on requirements determination before committing to XP. The Discovery customer had helped found the company, worked as a developer early on, and later was also the development manager. Atlantis had already designed a new technical infrastructure before committing to XP, and would involve their QA team only late in project.

The kind of person playing the customer role also varied according to the commercial nature of the project. In Endeavour, the customer was a domain expert from the client organisation. In Discovery, the customer was a founder of the start-up and, as typical in start-ups, had to play several roles as the company took shape. In Atlantis, the customer was the product manager, and hence in close touch with existing and prospective purchasers and users of the new product. In each of the cases, the customer seemed well-prepared to play the role, and seemed a sensible choice for the particular kind of project involved.

In each of the cases, XP appeared to be working well enough, and the projects all seemed to be looking successful, with some credit being given to the XP process. Of course, projects do not always go smoothly, as we discuss in one of our other studies [11].

Each of the customers did use some strategies to cope with the workload. For example, the Endeavour customer was supported by a more senior manager, an acceptance team, and an analyst involved early on. The Discovery customer was supported by one of the company founders, a research scientist involved in demonstrating the product to potential buyers, a team of scientists verifying the software, and an experienced software tester. And the Atlantis customer was supported by a UI designer, a technical architect, a technical writer, and a QA team. In other words, each of the customers was assisted by what initially seemed a sufficient support structure.

But in the experience of the person in the customer role, there was a conspicuous commonality. In each of the cases, the customer was under stress, and committing long hours, to fulfil the role. In each case, the customers coped, but the question of sustainability is clear.

## 7. Related Work

Members of the agile community have begun to raise some of the issues potentially associated with the XP Customer Role. Deursen [6] noted that XP provided a

“bag of tricks” for developers but little systemic guidance for a customer to succeed. To begin to address this need, Deursen ran a workshop at XP2001 that aimed to collate a number of guidelines for successful customer involvement. The workshop group agreed that developing user stories with a “so that” section of the story assisted the customer to evolve their long-term goals of the system. The workshop group also agreed that the customer role typically can not be held by one physical person, rather a customer team is required that includes specialist expertise such as acceptance testing as well as domain area specialists, as it is difficult for one person to represent such a diversity of users, from system administrators to senior managers. Another customer team suggestion was for a programmer to pair with the customer, in order to provide effective support for the customer. It was noted that it would be essential for the customer pair to facilitate customer/programmer communication within the team rather than blocking this communication. In all of the three cases presented in this paper, an informal customer team was established, and all of the teams included acceptance testing support. However, the customer remained overloaded. EagleCorp’s implementation of personas extends the concept of the recommended “so that” section and was found to be successful, not only in helping the customer to identify requirements but also as a communication mechanism for the programmers.

Other agile methods such as Agile Modeling (AM) [1], Lean Development [13], Adaptive Software Development (ASD) [8] and Scrum [14] include more practices to address requirements gathering and analysis, and it may be possible to incorporate some of these practices to address the short-comings of the XP Customer Role. For space reasons, we consider only one of these, ASD, in depth.

ASD is based on complex adaptive theory and describes the mind set and principles required to succeed in developing software iteratively and incrementally. ASD recommends the following for requirements:

- There are a set of artefacts (project vision, data sheet and specification) that ensure a shared project vision exists.
- Collaboration techniques are used to evolve requirements: JAD sessions, Customer Focus Groups and finally post-mortems or process improvement reviews.
- A collaboration facilitator role is introduced to focus on “thinking” about and planning collaboration rather than simply “letting it happen”.

There is significant cross-over between ASD and XP, including:

- The user stories and product specifications all ensure there is a centralised list of requirements

- The on-site customer role and the JAD/CFG sessions all aim to ensure the developers do not need to deal with conflicting requirements, including the prioritisation of requirements
- The small and regular iterations ensure there is opportunity to learn and to evolve requirements.

ASD concentrates on outlining learning and collaborative techniques and theories for all project members, and introduces a collaborative facilitator role that specialises in collaborative and learning based communication, including the communication between customers and developers. XP appears to under-emphasise the challenging job [10] facing the *on-site customer* and in gathering and prioritising the requirements for the project. ASD begins to address this issue by recognising the need for structured collaborative techniques and roles in this area.

## 8. Conclusion

At the end of Arthur Miller's "Death of a Salesman", the speech that passes for Willy Loman's eulogy includes the idea that:

*"For a salesman, there is no rock bottom to the life. He don't put a bolt to a nut, he don't tell you the law or give you medicine. He's a man way out there in the blue, riding on a smile and a shoeshine."*

An XP customer is in much of the same situation as Miller's salesman. Customers don't write code or refactor it – developers do that; they don't make the key decisions about the project – in any large business, rather more important managers will do that; nor do they tackle the key technical or process problems with the development – the XP Coach does that. Rather the customer "sells" the requirements of the business to the development team and then "sells" the products of development back to the business.

One of the key principles of XP is that "business people make business decisions, technical people make technical decisions". So phrased, this sounds like an equal balance. On the other hand, the customer must be the business's main interface to the development team. This requires the customer to keep in touch with the ongoing evolution of the business – XP allows development to keep track of rapidly changing business requirements, but the conduit for communicating these rapidly changing requirements to development is once again the customer. Simultaneously, the customer must maintain the project's credibility, and funding within the wider business environment. In all but the smallest business, the customer will have to answer to other business stakeholders – managers, employees, boards etc.

Compounding this effect, as the team member closest to the rest of the business, the Customer is often saddled with additional administrative responsibilities, often hiring team members, finding working space, providing equipment and infrastructure services, liaising with internal or outsourced systems management facilities both for the development and the final production or deployment environments, right down to apparently trivial tasks like arranging sick leave, and buying index cards.

XP practices support the development team in making the technical decisions that are their responsibility (Spike Solution, Once and Only Once, Merciless Refactoring, Do the Simplest Thing That Could Possibly Work). XP includes very few practices that actually support the customer in their role – *other than proscribing how they interact with the developers*. Furthermore, the practices – primarily the Planning Game and interactive nature of XP – effectively tie the Customer to meet the technical rhythms and needs of the development. The customer is supposed to provide stories or tests or advice or clarifications to the development team instantly, whenever they are required. In practice, it is far too easy for *anything* that is not to do with programming, or that is not explicit covered by XP's practices, to be defined as a "business" requirement, and thus, to become the sole responsibility of the customer. As in Project Discovery, it is easy to see why a Customer may spend forty hours a week working with the development team, and another forty hours dealing with everything else.

The existing XP Customer practices appears to be achieving excellent results, but they also appear to be unsustainable, and so constitute a great risk to XP projects, especially in long or high pressure projects. We believe that our research shows that, even when most of the relevant XP practices have been followed, the customer role is difficult and requires serious consideration. XP has focused on building effective development team practices: we now need to turn our attention, given the pivotal nature of the customer role, to exploring the processes that will support the XP customer.

## 9. References

- [1] Ambler, S., *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, John Wiley & Sons, 2002.
- [2] Beck, K. *eXtreme Programming Explained: Embrace Change*, Addison Wesley, 2000.
- [3] Beck, K., & Fowler, M. *Planning Extreme Programming*, Addison Wesley, 2001.

- [4] Cockburn, A. *Agile software development*, Addison-Wesley, 2001.
- [5] Cooper, A. *The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Sams, 1999.
- [6] Deursen, A., (Ed). *Proceedings of the XP Workshop on Customer Involvement in Extreme Programming*, Sardinia, Italy, 2001. <http://www.cwi.nl/~arie/>.
- [7] Fitzgerald, B. “Systems Development Methodologies: The Problem of Tenses”, *Information Technology and People*, 13 (3). pp. 174 – 185.
- [8] Highsmith, J. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House Publishing, 2000.
- [9] Martin, A. *A case study: exploring the role of customers on eXtreme programming projects*, CS-TR-03-1, School of Computing and Mathematical Sciences, Victoria University of Wellington, Wellington, 2002.
- [10] Martin, A., Noble, J., and Biddle, R. Proceedings of the Fourth International Conference on eXtreme Programming and Agile Processes in Software Engineering, Giancarlo Succi (Ed.), “Being Jane Malkovich: a Look into the World of an XP Customer”. *Lecture Notes in Computer Science 2675*, Springer-Verlag. 2003.
- [11] Martin, A., Biddle, R., and Noble, J., Proceedings of the Fifth International Conference on eXtreme Programming and Agile Processes in Software Engineering, Jutta Eckstein & Hubert Baumeister (Ed.), “When XP Met Outsourcing”, *Lecture Notes in Computer Science*, Springer-Verlag, 2004.
- [12] Nandhakumar, J. & Avison, D. “The fiction of methodological development: a field study of information systems development”, *Information Technology & People*, 12(2). pp. 176-191.
- [13] Poppendieck, M & Poppendieck, T. *Lean Software Development: An Agile Toolkit*, Addison-Wesley, 2003
- [14] Schwaber, K., Beedle, M. *Agile Software Development with Scrum*, Prentice Hall, 2001