

When XP Met Outsourcing

Angela Martin¹, Robert Biddle² and James Noble¹

¹ Faculty of Information Technology, Victoria University of Wellington
Wellington, New Zealand
{angela, kjx}@mcs.vuw.ac.nz

² Human-Oriented Technology Laboratory, Carleton University
Ottawa, Canada
robert_biddle@carleton.ca

Abstract. Outsourcing is common for software development, and is the context for many projects using agile development processes. This paper presents two case studies concentrating on the customer role in projects using outsourcing and extreme programming (XP). The studies follow an interpretive approach based on in-depth interviews, and suggest some tensions between some contractual arrangements in outsourcing, and the XP process. In particular, one suggests XP worked well in the context of their particular outsourcing arrangements, and the other study suggests difficulty in aligning XP with a different set of outsourcing arrangements.

1 Introduction

Outsourced software development has become commonplace in today's business environment [3, 5]. The outsourcing environments of today are complex, involving multiple organisations for different services: management, development, infrastructure and integration, to name but a few. Agile development, as it moves into the mainstream of software development, will come face-to-face with the realities of these complex outsourcing environments. Outsourcing arrangements typically result in the XP customer residing in a separate organisation to the development team. Our research, [6, 7] which focuses on the XP customer role, quickly brought to the fore that outsourcing is an issue facing the XP customer.

In this paper we present two case studies that highlight the practical realities faced in using XP with complex outsourcing arrangements. In the next section we outline some of the related work in this area, followed by our research method. The fourth and fifth sections outline the two cases, while the final section presents our conclusions.

2 Related Work

Members of the agile community have begun to raise some of the issues potentially associated with outsourcing arrangements and agile development.

Ambler [1] argues that outsourcing is riskier than it initially appears and organisations should seriously consider an alternative to outsourcing; that of running a small internal agile development team. Ambler's main argument is that the agile alternative may gain many of the intended cost savings without some of the inherent risks of outsourcing. So while Ambler does raise the question of should the organisation really be outsourcing, he does not provide detailed advice to the practitioner on how to effectively work in an outsourced environment.

Poppendieck & Poppendieck [9] take a different tack. Recognising that there is a perceived barrier to agile development in an outsourcing situation, they have explored the potential implications of different contracting models. The contracting models they have reviewed and analysed include fixed-price, time-and-materials, multi-stage, target-cost, target-schedule and shared-benefit contracts. They conclude that contracts that allow optional scope are more likely to create effective software development environments. Since the publication of their book [9], the Poppendiecks have run a series of workshops [10] at international conferences that are aimed at further evolving our understanding in this area.

3 Research Method

Information Systems Development (ISD) methodology researchers [2, 8] have expressed a growing concern that existing ISD methods do not meet the needs of today's business and software development environments. Studies [8, 7] 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. Two outsourced projects are explored, in the first we interviewed 5 project participants and in the second we interviewed 3 project participants. The interviewees, in both cases, have covered the spectrum of core XP roles including the customer, programmer, coach and tester. In the second case, the participants moved between roles over the life of the project. 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. 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.

4 Project Endeavour

This section tells the story of Project Endeavour, an outsourced XP project involving:

- KiwiCorp, a large New Zealand company, is the customer organisation
- DevCorp, a large consultancy, is the development vendor and
- 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 [6, 7].

4.1 Case Description

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

The relationships amongst these three companies, outside of the specific project, are complex and also worth noting. The items of particular interest are:

- KiwiCorp and BureauCorp have a business alliance, they work together to deliver outsourced services. Each company also delivers services outside of the alliance.
- BureauCorp is the outsourced vendor for all KiwiCorp’s internal infrastructure.
- BureauCorp and DevCorp both have software development service lines and compete against each other in this area.

Please see the technical report of this research [6] for further information concerning Project Endeavour.

4.2 Participant Reflection and Discussion

During the course of the interviews, the participants reflected on their experiences with XP and outsourcing. These reflections are outlined and discussed in this section.

Time and Materials Contract. The project manager from DevCorp reflected on some of the benefits they encountered working on a time and materials contract, rather than working on a fixed price arrangement:

“We were very fortunate that we were working on [a] time and materials contract with [KiwiCorp], if we were working on a fixed price per iteration ... I would have had to be a lot harder on the client... I would have to have been a lot more strict about things like getting change signed off and if circumstances came along [e.g. infrastructure difficulties caused by BureauCorp] and we lost a day ... I’d have to go back to the client and say well actually you’ve lost a day ... therefore you need to subtract that functionality [from this iteration] ... I was sort of able to let that ride with the client ... It didn’t become blindingly obvious to me until I was [discussing the issue with a manager responsible for a fixed price XP project] ... if at the end of the iteration if we had not finished all of the stories on the wall we would just put them into the next iteration, but [on the fixed price project] they’ve still got to finish them without getting any more money” – Project Manager, DevCorp

So a fixed price contract arrangement would result in a changed working relationship between the two companies. In a fixed price arrangement, the vendor needs to add an overhead to the process to ensure the sorts of issues noted above, would be at the client’s expense and not the vendors. However, the project manager also noted some of the limitations of the time and materials approach encountered on this project:

“We didn’t establish criteria for [the customer to complete acceptance testing] this is one of the differences between us [time and materials contract] and a fixed price contract ... if we were working on a fixed price basis we would have had to put some criteria around a sign-off.. .we’ve not had to do that but what it does mean is that you don’t get a very quick turnaround” – Project Manager, DevCorp

The obvious impact of not having agreed delivery dates for acceptance testing was that (a) the defect stream was unpredictable making it difficult to manage the development team’s workload and (b) the feedback from the customer was not immediate and hence errors in the development team’s understanding remained for a longer time. Perhaps the not so obvious issues are (a) the customer may not realise that their actions caused this chain of events that affects the cost and (b) the customer may not realise that DevCorp team members were liable to interpret the customer’s tardiness as a lack of commitment from KiwiCorp, potentially impacting their own commitment to the project. Although it would be possible to negotiate and agree delivery dates in this area the lack of a contractual agreement can make it difficult to enforce these dates. It is difficult for a services organisation such as DevCorp to ‘chastise’ KiwiCorp regarding this behaviour without endangering future work streams from KiwiCorp.

Multiple Organisations. This project not only had three organisations, but three organisations approaching the project with very different processes:

“We were happy to step out of XP and work within the processes that [BureauCorp] dictated for any kind of change or to migrate any software or to actually have any software installed so we’ve almost had to take the whole infrastructure thing and stick it out of XP” – Project Manager, DevCorp

All of the interviewees agreed that the effect of multiple organisations involved in the project, using a mixture of agile and non-agile processes, affected the timeline of the project, as summed up by this quote:

“I was fully [committed to] the project but I was ... all of these technical integration issues were just taking up about half the time [and later in the interview regarding the project delay] the big delay has been in the technical integration. You know, just getting the application to work on the software in the various environments” – Customer, KiwiCorp

The term “technical integration issue” was used to refer to all of the issues encountered between DevCorp and BureauCorp. The impact of the inter-organisational issue was that the customer had significantly less time to spend on the requirements and testing tasks. While the customer understood that this impacted the quality of the delivered software, the flow-on delay caused by the quality issues appeared to be eclipsed by the obvious technical integration issues encountered.

The project manager recognised that the relationship between the three organisations was not a simple one and noted:

“[KiwiCorp] at the end of the day is also a client of [BureauCorp] ... because of the nature of the relationship they have, [KiwiCorp] haven’t got any weight to push around, they signed up for these processes and they have to follow them so there is frustration on both sides of the camp” – Project Manager, DevCorp

He also reflected on how to improve the situation between the three organisations next time:

“We need to actually work together to build something that actually doesn’t hinder the project process ... this is my first time working with three parties as in another party owning a significant portion of the project and I think sitting down and sorting those rules of engagement out at the start would be something in hindsight I would have like to have done” – Project Manager, DevCorp

Summary. This story focussed on some of the differences between a fixed price and a time and materials outsourcing contract arrangement on agile development projects, concluding that there are benefits and limitations to both arrangements. It also highlighted the types of issues encountered when multiple organisations are involved in the project, which appears to be an increasingly common trend. We need to ensure that agile contracts and approaches are able to work in these complex outsourcing environments.

5 Project Pinta

This section tells the story of Project Pinta, an outsourced XP project involving:

- RCCorp, a large organisation based in Europe, is the customer organisation
- ManageCorp, an international management consulting, technology services and outsourcing company, is the outsourced vendor for all of RCCorp's IT functions and plays the role of proxy customer
- FalconCorp, a large international software product company based in the United States of America, is the development vendor.

The data that forms this paper was collected in a series of interviews with FalconCorp project participants.

5.1 Case Description

Project Pinta has not been seen as a success by FalconCorp, and part of the concerns with the project have been the use of an agile development method. The project has taken longer than the original schedule, currently more than three times the original estimate, and has yet to be accepted by the RCCorp. The project can be divided into two stages, Stage I covers the originally estimated period and Stage II covers the remaining time. Stage I started out as a typical “death march” project:

“Everyone that was on it said it was doomed for failure [but] we were going to make it work anyway” – Customer Proxy, FalconCorp.

The team had less than six months to cover almost 30 functional areas, so management quickly scaled the project up to more than 60 people across several XP labs.

Project Pinta is not a typical custom build project, as the intent of this project is to develop a product that will both meet the needs of RCCorp and become a product within FalconCorp's product suite. The labs were structured so that each lab had a customer representative, two of the labs had product managers from FalconCorp and two of the labs had business analysts from the European office of ManageCorp. The ManageCorp analysts had been involved in writing the initial high level requirements that had been accepted by RCCorp. No representatives from RCCorp were on the project.

The team quickly established a pace with weekly iterations. At the end of each week the team produced working software that implemented the prioritised stories. The problem, it quickly became obvious that all of the functionality would not be completed by the required deadline, despite everyone's best efforts. However, FalconCorp had committed to deliver the functionality by this deadline. The unintended outcomes of how this unwelcome knowledge was handled are outlined below.

In typical software development projects, *scope creep* must be managed by the outsourced vendor, particularly in an *aggressive* fixed price project such as this one. FalconCorp managed the scope by not exploring exceptions or evolving the requirements:

“We had the client on site but we didn't make use of them in [the] way that we should [as] we didn't want to raise any questions that would lead

to gaps in the requirements that we would ultimately be responsible for ... and so we built [to the specification] without asking questions ... [our aim was to build a product that] we could check off and say ... we made our deadline, we deserve our payment” – Customer Proxy, FalconCorp

In fact, the level of mistrust and organisational self-protection rose to such a level on this project, that the ManageCorp analysts were removed from the labs:

“There [were] roadblocks that were on both sides [that stopped] real communication. We had the client over here and I remember the first week that I was here they were talking about kicking one of the [ManageCorp] people out of the labs because they were a spy ... they were going back and reporting what was going on in the labs... You’d think that was part of what they were for but, you know, they thought they were giving a bad impression back to the client ... and so they actually kicked them out of the labs at one point.” – Customer Proxy, FalconCorp

The result of this environment occurred at the end of Stage I when members of the team presented the software to RCCorp in Europe:

“It was kind of a sales pitch because they [RCCorp] didn’t really test it. We [demonstrated how it worked with] some scenarios instead of letting them just hammer anything, because if you hammered on anything, you [would] have big huge gaping holes” – Customer Proxy, FalconCorp

The demonstration to RCCorp went well, so well that FalconCorp laid off over two thirds of the original staff, as they were now entering “*bug-fixing*” mode. During the first two months of Stage II, new project management was appointed in FalconCorp. These new managers quickly assessed that the project required significantly more work than “*bug-fixing*”. The final decision was to treat Stage I as a throw-away prototype and re-write the entire product. The contract was renegotiated and FalconCorp expect to complete the project in mid 2004.

5.2 Participant Reflection and Discussion

During the course of the interviews, the participants reflected on their experiences with XP and outsourcing. These reflections are outlined and discussed in this section.

Scope Definition. A non-negotiable area for FalconCorp as part of the contract renegotiation was an up-front requirements gathering and documentation stage. As an indication of the size of the gaps discussed, one of the documents expanded the initial requirements document from a 30 page document to a 250 page document.

Schwaber [11] writes that agile development, in this case Scrum, is not a silver bullet for fixed price contracts. In fixed-price contracts an upfront requirements gathering phase is required to accurately estimate scope. It is interesting that FalconCorp did not believe Stage I gave them sufficient knowledge and that an additional significant up-front requirements activity was commissioned. During the interviews potential reasons were uncovered, including (a) significant product knowledge was lost when the majority of the staff from Stage I were laid off and this was compounded with XP

documentation practices and (b) there was a perceived failure of XP by management and so waterfall processes were seen as 'safer' in a contracting arrangement.

Development Process. Stage II of this project has yet to complete but the interview participants' are clear that the project is now much closer to a typical waterfall process than an XP process. One participant reflected on this change:

" XP [if you go straight from the book] is useless from my perspective ... if you have a fixed scope, a fixed date, well ... how well is that spec going to play in that space – not very, because it relies on the ability to manage that and to change that and impact that over time as realities come out. Whereas with a contractual arrangement, you can't, you have very little wiggle room to be able to shift that and manage that and change that, and so XP finds itself significantly hampered and you have to start shifting it more towards Waterfall over time, because you just had to deal with the realities of the fact that you must get certain things out in certain timeframes." –Development Coach, FalconCorp

It appears XP will end up taking the blame for the problems encountered within this project in FalconCorp. And while it seems from both of these cases that there are issues with using XP on fixed price contracts without an upfront requirements gathering and documentation phase, perhaps there is more to this case. Kern, Willcocks and Heck [4] outline a typical outsourcing issue called the *Winner's Curse* that may play a significant role in this case.

The Winner's Curse occurs in many bidding situations, including IT Outsourcing. The example of an auction helps to explain the concept. Many of us can relate to the time when in the heat of the bidding we purchased an item for much more than it was worth, often making a loss or at least no profit on the transaction. The Winner's Curse is a prevalent issue in IT outsourcing. One of the common reasons it occurs is that the vendors do not understand the full size of the project, often due to inaccurate assumptions made during the bidding process. As a result, during the project the vendor concentrates on cost-cutting strategies, such as placing inexperienced staff on the project or focussing on the letter of the contract scope not the spirit, to reduce their loss. The end result for both parties is a loss situation, as the resulting software is unlikely to meet the client's needs. Project Pinta exhibits all of the signs of the Winner's Curse.

Summary. This story focussed on some of the typical issues faced by companies in an outsourcing environment. One of the key issues highlighted is the need for an up-front scope definition phase for fixed price contracts, irrespective of the development process. The second issue discussed was the Winner's Curse; that of being the successful bidder for a contract that will result in a loss, or very little profit, for the winning vendor. The Winner's Curse is a prevalent issue in IT outsourcing, irrespective of the contract type or development process, and significantly impacts the project environment. Developers who find themselves with the Winner's Curse should consider carefully what they want from an agile process.

6 Conclusions

We are studying the customer role in agile development, and in this paper we presented two interpretive case studies where XP was used in projects involving outsourcing. The studies are based on in-depth interviews with project participants. In both cases we saw a strong awareness of the interactions between outsourcing arrangements and the XP process.

In the first study, we learned that the agility fostered by continual contact between developers and customer worked well with the contractual arrangements based on time and materials charging. In the second study, we found a more fixed contractual basis for the project, and heard about experience that suggests this clashed with the XP process. In both cases, we saw that the involvement of multiple organisations required accommodation of the different cultures of the organisations, and presented challenges in interpreting the XP customer role.

All these findings show us that for XP to embrace change, an organisational and contractual context is needed that allows change to be embraced without penalty.

References

- 1 Ambler, S. Outsourcing Examined. *Software Development Journal*, April 2003.
- 2 Fitzgerald, B. Systems development methodologies: the problem of tenses. *Information technology and people*, 13 (3). pp. 174 - 185.
- 3 Kern, T. The Gestalt of an Information Technology Outsourcing Relationship: An Exploratory Analysis. *Proceedings of the Eighteenth International Conference on Information Systems*, Atlanta, Georgia, United States, pp: 37 – 58, 1997.
- 4 Kern, Thomas, Leslie P. Willcocks, Eric van Heck, The "Winner's Curse" in IT Outsourcing. *California Management Review* 44, no. 2, p. 47-69, Winter 2002.
- 5 Klepper, R & Jones, W. *Outsourcing Information Technology, Systems, & Services*. Prentice Hall PTR, 1997.
- 6 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.
- 7 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.
- 8 Nandhakumar, J. & Avison, D. The fiction of methodological development: a field study of information systems development. *Information Technology & People*, 12(2). pp. 176-191.
- 9 Poppendieck, M & Poppendieck, T. *Lean Software Development: An Agile Toolkit*. Addison-Wesley, 2003
- 10 Poppendieck, M. Poppendieck, T (Eds). *Proceedings of the Oopsla Workshop on Agile Contracts*, Anaheim, California, USA, 2003. <http://poppendieck.com/>.
- 11 Schwaber, K. Fixed Price, Fixed Date Contracts at Engage. In Poppendieck, M., Poppendieck, T. (Eds.), *Proceedings of the Oopsla Workshop on Agile Contracts*, Anaheim, California, USA, 2003. <http://poppendieck.com/>.