Outsourcing and Offshoring with Agility: A Case Study

Clifton Kussmaul[1], Roger Jack[1], and Barry Sponsler[2]

[1]Elegance Technologies, Inc.
1721 Green Valley Rd, Havertown, PA 19083, USA
{ckussmaul,rjack}@elegancetech.com
http://www.elegancetech.com
[2]EXTOL International, Inc.
474 North Centre St, Pottsville, PA 17901, USA
bsponsler@extol.com
http://www.extol.com

Abstract. We describe techniques and lessons learned from using agile methodologies with distributed teams, specifically outsourced and offshore development teams. Such teams often need to contend with multiple organizational boundaries, differences in time zone, language, and culture, and other communication challenges. First, we present concepts and issues in outsourcing and offshoring. Second, we describe a case study involving continually changing requirements, outsourcing, offshoring, and a method inspired by SCRUM and FDD. Third, we review key lessons learned, and conclude with a summary.

1 Introduction

Agile approaches to software development share a particular set of values [6], [11]:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Although agile methodologies usually assume that all team members are in one location, they have also been adapted to include distributed teams [17].

This paper describes techniques and lessons learned from using agile methodologies with outsourcing and offshoring. We begin with background on outsourcing and offshoring. We then describe an ongoing relationship in which Elegance Technologies, Inc. (ET) provides software product development services to EXTOL International, Inc. Next, we present lessons we have learned that can help other projects with distributed teams. These lessons are summarized in our conclusions.

1.1 Outsourcing and Offshoring

Outsourcing is the use of external companies to perform services, rather than using internal staff. According to a 2003 survey of CIOs, 70% of companies outsource some IT function or application [5]. *Offshoring* is the use of staff in other countries, and is often associated with India, China, and the former Soviet Union. Forrester Research estimates that 277,000 computer jobs and a similar number of management and operations jobs in the United States will move offshore by 2010 [10].

C. Zannier et al. (Eds.): XP/Agile Universe 2004, LNCS 3134, pp. 147–154, 2004.

Outsourcing is done for several reasons [5]:

- To reduce costs. This is the most commonly cited (59.8%) advantage of offshoring, although 73.5% of CIOs feel it is overrated as a cost-cutting strategy [5]. Hourly rates in countries like India and China can be less than 10% of those in the US [9], but IT organizations typically save 15%-25% during the first year and up to 40% by the third year [7].
- To access specialized skills or facilities. These resources may be expensive or unavailable locally, or may only be needed occasionally.
- To be able to increase or decrease the people or other resources on a project.
- To increase development speed and reduce time to market.

A useful framework [12] analyzes outsourcing relationships along two dimensions. *Dependence* is the degree to which ownership and control is transferred to the outsourcing partner. *Strategic impact* is the degree to which the outsourcing relationship affects competitive positioning and long-term strategy. As a relationship increases in either of these dimensions, collaboration becomes more critical.

2 Case Study

EXTOL International, Inc. is a privately funded software product company based in the mid-Atlantic region, with roughly $10M in annual revenue. Most employees work in a development and operations center, or a sales and marketing office, though there are several other offices across the US. Over the last several years, EXTOL has invested significant resources to develop EXTOL Business Integrator (EBI), a powerful and flexible data processing engine, with a strong development team, and an extensive feature roadmap. Thus, EBI is a strategic asset for EXTOL and EXTOL plans to keep most core EBI activities in-house.

To allow customers to use EBI's capabilities without first becoming expert users, EXTOL plans to develop a series of user-friendly, domain-specific front ends, beginning with one for UCCnet, a global, internet-based electronic commerce service [20]. However, these front ends present several challenges:

- They are often developed in response to external market conditions which are hard to predict and difficult to coordinate, and the number and intensity of these development efforts will vary over time.
- They may require intensive domain analysis phases, and are targeted at business users, not technical users. Diverting EXTOL's existing architects and designers to work on the front ends would adversely affect EBI.
- Cash flow must be monitored closely, since EXTOL is already supporting a significant development effort.
- EBI is under active development, and new capabilities are added regularly.
- The external standards are still evolving, in some cases. For example, several months into the project the UCCnet specification migrated from DTD to XSD.
- Time to market is critical, and system requirements may change frequently in response to customers and competitors.

Thus, EXTOL decided to outsource development of the UCCnet application to Elegance Technologies, Inc. (ET), which develops software products and provides software product development services, including managing offshore teams.

The outsourcing relationship began with one ET consultant working on an hourly basis, and spending several days a week onsite at EXTOL. The consultant studied the UCCnet application domain, analyzed requirements, developed a high-level design and a list of potential features, and supported development of an in-house proof-of-concept and demo. Based on the confidence and trust developed through these initial deliverables, EXTOL decided to outsource most of the UCCnet application development to ET.

2.1 Methodology

Our methodology is inspired by SCRUM and Feature-Driven Development (FDD). In SCRUM [16], a *product backlog* contains all potential product features, including customer functionality and technology features. A *sprint* is a 30 day development period. A *sprint backlog* is a subset of the product backlog to be implemented in a given sprint. A *sprint goal* defines the business purpose of a sprint, and is used to help decide whether specific features must be deferred or changed. During a sprint, daily 15 minute meetings are used to inform and coordinate the team. In FDD [13], a team begins by developing an *overall system model*. The team then generates a list of features, and plans the features. Finally, the team bundles sets of features into packages for design and construction. Thus, SCRUM emphasizes project management, and FDD emphasizes a design process.

Early in the project, we developed an initial feature list (or product backlog) and the overall product architecture (or system model). Ongoing development involves a team of 2-3 onshore staff and 5-10 offshore staff, working in time-boxed sprints of 4-6 weeks. Before each sprint, we develop and sign a formal proposal that identifies the major milestones, the sprint feature list, and a price range. During each sprint we implement sets of features. At the end of each sprint, ET determines the final price based on actual effort and the scope of work completed. This allows us to be more responsive to uncertainty and requests for change.

During a sprint, we use several coordination techniques:

- Both teams use a shared mailing list to archive all communication.
- At the end of their respective work days, each team sends a status email to the list, describing significant changes, current problems, and any questions.
- There are also daily meetings to review status and address major problems that are preventing the project from moving forward. They are usually late afternoon in India and early morning in the US, but sometimes early morning in India and late evening in the US. These meetings are usually via instant messaging and last around 15 minutes, though they occasionally run as long as 30 minutes and sometimes involve phone and/or web conferencing.
- A CVS repository stores all requirements, designs, source code, and related documents. The code is kept in a working state at all times, with any exceptions noted in the status email, so that we can see real progress. In effect, we have daily code deliveries between the offshore and onshore teams.

Onshore staff focus on analysis, high-level design, and coordination with EXTOL, but are also involved in the low level implementation and testing. For example, onshore staff may develop GUI mockups or framework code to be completed offshore.

A top priority for the onshore staff is to resolve any open issues from the daily meeting, so that the offshore team can continue work the next day. Offshore staff focus on low-level design, implementation, and testing, but are also involved in analysis and design, and review all design documents and proposals.

Effective collaboration is probably the most important element in the success of this project, which involves customers, EXTOL's sales and marketing group, EXTOL's development team, ET's onshore team, and the offshore team. Establishing trust over these multiple boundaries takes continual effort and attention.

3 Lessons Learned

We hope that other projects involving outsourcing, offshoring, or other distributed teams can benefit from lessons we have learned, including the following:

Avoid projects that are too small to amortize the overhead required for a effective distributed team. Very small projects are best done by local teams, unless an offshore team already has direct experience. On the other hand, it is often best to start with a small offshore team and add people over time as the project grows.

Keep research, architecture, and requirements analysis close to the customer. For example, during the first few sprints the onshore team developed a core architecture and set of practices to serve as guidelines, particularly as the offshore team evolves and becomes more familiar with the project and application domain. As the project and team grew, ET's onshore staff became less involved in coding, and more involved in planning, reviews, and other coordination to serve as a bridge between the offshore team and EXTOL.

Key documents help to bootstrap the project by establishing a common framework for stakeholders spread across multiple US and foreign time zones. We depend on:

- The master feature list – every completed, scheduled, or envisioned feature.
- A detailed data model and data dictionary are key interfaces between teams and components, including the interface to the EBI software engine.
- Requirements and designs for subsystems in the current or next sprint. These documents are usually discarded (i.e. not maintained) after the corresponding features are developed and tested.
- A list of tasks and related information for the current sprint.

This supports Brooks' *documentary hypothesis*: "a small number of documents become the critical pivots around which every project's management revolves" [4]. Thus, we try to minimize the use of unnecessary or throwaway documents. For example, our GUI reviews use real screens with incomplete functionality, rather than dummy screens that might change when implemented.

Minimizing requirement changes during a sprint is even more important with a distributed team, since it can be more difficult to agree on the scope, monitor the implementation, and ensure that nothing is forgotten. Often, it is feasible to defer changes until the next sprint, which is usually just a few weeks away. If changes are necessary, we may compensate by reducing other functionality or slightly extending the schedule.

As the project matures we have allowed the sprints to become longer, in part to reduce the testing and documentation overhead for each delivery. However, we spend more time debating the scope of each sprint, so we may need to revisit these issues.

Careful coordination enables us to respond quickly to necessary changes, which can provide a competitive advantage. For example, by focusing on activities such as analysis, customer interaction, and testing, at the end of the day the onshore team can send requirements to the offshore team, and have the resulting changes implemented the next morning. This round-the-clock cycle can be very effective. Because each team is eager to start work when the other finishes for the day, this approach also discourages excessive overtime, which can lead to burnout and other problems.

Early and frequent delivery of working software is especially important, for several reasons. It gives the client frequent opportunities to review the project status and make appropriate changes, which is especially important for user interfaces. It serves to build confidence and trust among people who have not worked together previously. Furthermore, it provides an effective communication mechanism between all stakeholders. These benefits are essential in the UCCnet project environment.

Planning is still important, although it takes less time than many outsourced projects. For example, we must monitor costs, and balance short-term priorities against the need for a sustainable level of staffing. Since each sprint is a separate proposal and contract, during each sprint we spend time planning the next one.

Ease participants into relationships with remote teams, and try to arrange regular and extended visits between locations. For example, several members of the offshore team have worked in the US. ET's onshore staff have all spent time in India, and typically spend at least one day a week onsite at EXTOL. We also arrange occasional teleconferences (usually for training) between EXTOL, the onshore team, and the offshore team.

Focus on the win-win aspects of the project to minimize the potential disruption caused by contractual relationships between the distinct organizations involved. For example, the proposal for each sprint specifies a price range to accommodate incomplete and changing requirements, and to avoid having to write complete requirements before the sprint can start. We are fortunate in that many key personnel at EXTOL and ET have been both producers and consumers of software services, and thus understand the issues from both perspectives.

Effective communication and interaction are particularly important in outsourcing and offshoring, where staff can be spread across multiple locations, time zones, and cultures. It can be difficult to determine and maintain the appropriate level of communication. Working to develop effective communication early in the project makes it easier for the team to grow as needed. Synchronous communication, such as face-to-face meetings, online chats, teleconferences, and web conferences, is ideal for quick status meetings, brainstorming sessions, and reviews. Asynchronous communication, such as email, discussion forums, and shared documents, provides a persistent record of discussions and decisions, and don't require participants to be available at the same time. We employ all of these techniques regularly.

Be sensitive to cultural differences, especially between organizations and between regions or countries, including differences in how people interact with each other and

resolve problems. This is especially true for outsourcing and offshoring. For example, some cultures place more value on centralized, top-down control, and may view direct questions as a challenge to authority.

Furthermore, many offshore development centers, particularly in India, have invested heavily in the Software Capability Maturity Model (SW-CMM®) [14], which defines five levels of increasing process maturity; offshore companies represent roughly 74% of CMM-4 organizations and roughly 84% of CMM-5 organizations [19]. When teams and managers are accustomed to disciplined processes and relatively static requirements, it can be quite difficult to convince them to explore other approaches. (Note that there is growing recognition that both agile and disciplined approaches have advantages, and that often a carefully designed combination of the two can be very effective [2],[3],[15].)

Concentrate on the organizational interfaces when defining processes, rather than trying to define all of the processes for everyone involved in the project. Defining processes for an outsourcing or offshoring project can be particularly challenging, since it may require coordination between different organizational cultures. It is quite feasible, and sometimes preferable, for teams to use different methodologies that reflect different cultures and requirements.

Use tools to work smarter, not harder. For example, early on we recognized that we could use a code generator for much of the access code for a database with several hundred tables. We also use JUnit and IBM Rational XDE Tester (formerly RobotJ) to support unit testing and regression testing of the user interface.

4 Conclusions

To quote DeMarco and Lister, "The major problems of our work are not so much *technological* as *sociological* in nature." [8] (original emphasis). In this paper, we have identified ways in which outsourcing and offshoring can utilize agile approaches to address these sociological problems. Key lessons learned include:

- Avoid projects that are too small to amortize overhead.
- Keep research, architecture, and requirements analysis close to the customer.
- Use a few key documents to provide a common framework.
- Minimize requirement changes during a sprint.
- Coordinate carefully to allow distributed teams to respond quickly to changes.
- Deliver working software early and often to build confidence and trust.
- Recognize that planning is still important.
- Ease participants into relationships with remote teams, and arrange face to face contact whenever feasible.
- Focus on win-win aspects to minimize potential disruptions.
- Provide appropriate tools and infrastructure for effective communication.
- Be aware of and sensitive to cultural differences.
- Focus on the interfaces between teams and organizations, and recognize the potential value of different processes in different locations.
- Work smarter, not harder, by using appropriate tools.

We expect outsourcing and offshoring to continue growing. We hope the lessons described above can help other organizations to work more effectively and efficiently.

Acknowledgements

We gratefully acknowledge the support, advice, and encouragement we have received from our customers and colleagues, both onshore and offshore.

Author Biographies

Clif Kussmaul is CTO of Elegance Technologies, Inc. He is also Assistant Professor of Computer Science at Muhlenberg College, where he delivers introductory courses through capstone projects to traditional and non-traditional students. Previously, he spent two years working with CMM5 development centers at NeST Technologies. He has a PhD in CS from the University of California, Davis, and is the author or co-author of over thirty publications and conference presentations.

Roger Jack is President of Elegance Technologies, Inc. He has experience in project management, and creating reliable and robust architectures. He is the former Vice President of U.S. Software Operations for NeST Technologies, where he managed many offshore projects. He has an MBA from Duke University's Fuqua School of Business, and an MS in Computer Science from Villanova University.

Barry Sponsler is Director of Development for EXTOL International, Inc., where he oversees development, documentation, quality assurance, and outsourcing relationships. In over 20 years in the IT industry, he has managed diverse development teams for legacy applications (e.g. AS/400), multiplatform applications using Java, a large data center with operations and programming staff, and ERP and Y2K projects for Fortune 500 companies.

References

1. Agile Alliance. Manifesto for Agile Software Development. `http://www.agilemanifesto.org` (2001)
2. Anderson, D. Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Prentice Hall PTR (2004)
3. Boehm, B., Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley (2003)
4. Brooks, F. The Mythical Man-Month. Addison Wesley (1995)
5. CIO Insight. Research: Outsourcing: How Well Are You Managing Your Partners? 1(33):75-85 (November, 2003)
6. Cockburn, A. Agile Software Development. Addison Wesley (2003)
7. Davison, D. Top 10 Risks of Offshore Outsourcing. META Group (Nov 2003)
8. DeMarco, T., Lister, T. Peopleware: Productive Projects and Teams. Dorset House (1999)
9. Dignan, L. Leaping, then Looking. Baseline 1(22):17-29 (September 2003)
10. Engardio, P., Bernstein, A., Kripalani, M. The New Global Job Shift. Business Week (February 3, 2003)
11. Highsmith, J. Agile Software Development Ecosystems. Addison Wesley (2002)

12. Kishore, R., Rao, H.R., Nam, K., Rajagopalan, S., Chaudhury, A. A Relationship Perspective on IT Outsourcing. Communication of the ACM 46(12):87-92 (2003)
13. Palmer, S. R., Felsing, J. M. A Practical Guide to Feature-Driven Development. Prentice Hall PTR (2002)
14. Paulk, M., Weber, C., Curtis, B., Chrissis, M.B., et al. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison Wesley (1994)
15. Paulk, M. Extreme Programming from a CMM Perspective. IEEE Software 18(6):19-26 (2001)
16. Schwaber, K., Beedle, M. Agile Software Development with SCRUM. Prentice Hall (2001)
17. Simons, M. Internationally Agile. InformIT (March 15, 2002)
18. Software Development. Offshore by the Numbers. 12(1):39-41 (Jan 2004)
19. Software Engineering Institute. Process Maturity Profile: Software CMM® - CBA IPI and SPA Appraisal Results. (2003)
20. UCCnet. `http://www.uccnet.org` (2004)