

1 Outsourcing to an Unknown Workforce: Exploring 2 Opensourcing as Offshore Sourcing Practice

3 4 **Abstract**

5 *This paper presents a psychological contract perspective on the use of the open source*
6 *development model as an offshore outsourcing practice – opensourcing as we term it here –*
7 *whereby commercial companies and open source communities collaborate on development of*
8 *software of commercial interest to the company. Building on previous research on IS*
9 *outsourcing, a theoretical framework for exploring the opensourcing phenomenon is derived.*
10 *The first phase of the research concerned qualitative case studies involving three commercial*
11 *organizations (IONA Technologies, Philips Medical Systems and Telefonica) who had*
12 *‘liberated’ what had hitherto been proprietary software and sought to grow a global open source*
13 *community around their product. We followed this with a large-scale survey involving additional*
14 *exemplars of the phenomenon. The study identifies a number of symmetrical and complementary*
15 *customer and community obligations which are associated with opensourcing success. We also*
16 *identify a number of tension points on which customer and community perceptions tend to vary.*
17 *Overall the key watchwords for opensourcing are openness, trust, tact, professionalism,*
18 *transparency and complementariness: The customer and community need to establish a trusted*
19 *partnership of shared responsibility in building an overall opensourcing ecosystem. The study*
20 *reveals an ongoing shift from OSS as community of individual developers to OSS as community*
21 *of commercial organizations, primarily small to medium-sized enterprises. It also reveals that*
22 *opensourcing provides ample opportunity for companies to headhunt top developers – hence*
23 *moving from outsourcing to a largely unknown OSS workforce towards recruitment of*
24 *developers from a global open source community whose talents have become known as a result*
25 *of the opensourcing experience.*

26 **Keywords:** Open source, opensourcing, offshoring, outsourcing, global software development

Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy

INTRODUCTION

Companies have been offshoring manufacturing processes to lower-cost destinations for at least thirty years. However, it is really only since the mid 1990's that a significant portion of software development work is being performed offshore (Carmel, 1999). There are many potential advantages to be gained in offshoring software development, including reduced salary costs; reduced cycle time arising from 'follow-the-sun' software development; cross-site modularization of development work; access to a larger skilled developer pool; innovation and shared best practice; and closer proximity to customers (e.g. Carmel, 1999; 2006; Herbsleb and Grinter, 1999; Carmel and Agarwal, 2001; Ebert and De Neve, 2001; Carmel and Tjia, 2005). Of these, most emphasis has been placed on potential cost savings, which accrue largely due to the availability of a trained pool of development staff at a lower salary base. To take an example, the average annual base salary of a software development engineer in India in 2004 was less than one-seventh of the average annual salary of a US software engineer. Such potential savings have rapidly increased the amount of work being offshored from high-cost countries such as the US, UK and Scandinavian countries to lower cost economies such as India, China and Malaysia. The U.N. World Investment Report 2004 predicted that offshore outsourcing of IT-enabled business processes will increase 18-fold to almost \$25 billion by 2007 (United Nations, 2004).

Given that the primary force driving offshore outsourcing appears to be cost savings, it is perhaps natural that companies might eventually focus on the open source software (OSS) development model as a potentially even cheaper alternative, as there are significant cost savings

1 associated with the acquisition of OSS (Wheeler, 2004). Carmel and Tjia (2005) have
2 characterized offshore outsourcing as ‘outsourcing to a global workforce.’ In this study we
3 investigate the emerging trend towards opensourcing; that is, offshore outsourcing to a global but
4 largely *unknown* workforce of open source software developers.

5 Many different theoretical frameworks have been used to study outsourcing and offshoring.
6 Three of the more popular are agency theory (Cheon et al. 1995; Aubert et al. 2005), relational
7 exchange theory (Cheon et al. 1995; Goles and Chin, 2005) and transaction cost theory (Cheon et
8 al. 1995; Grover et al. 2002; Wang, 2002; Aubert et al. 2005). While these could potentially
9 provide candidate theoretical frameworks for our study, we saw shortcomings with each. For
10 example, transaction cost theory and agency theory are heavily oriented towards a traditional
11 economic perspective (Goles and Chin, 2005), whereas the majority of OSS development takes
12 place outside the realm of traditional economic theories. A further assumption of agency theory
13 is that the agent (vendor) and the principal (company) are able to share common goals and accept
14 the same degree of risk willingness and aversion (Gottshalk and Solli-Sæther, 2005), which is
15 probably hard to establish given the “unknown” factor inherent in opensourcing. While relational
16 exchange theory does recognize the importance of interactions, interdependencies, reciprocities,
17 and informally negotiated rules of exchange between parties, it also builds on the assumptions
18 that a formal contract is in place (Goles and Chin, 2005) and that the company can develop and
19 secure common norms with the vendor (Gottshalk and Solli-Sæther, 2005). Neither of these
20 assumptions is intuitively valid in an OSS context given the voluntary nature of typical OSS
21 communities and the tension between economic value that drives commercial organizations and
22 the community values that drive OSS (Fitzgerald, 2006). In order to overcome this bias towards
23 “traditional” long-term economic relationships, we chose to adopt the perspective suggested by

1 Koh et al. (2004), using psychological contract theory (PCT) as a basis for understanding the
2 mutual relationships between managers of offshore outsourcing customer organizations and
3 members of their global OSS community. Our reason for choosing PCT was thus because of its
4 emphasis on mutuality and reciprocity, its individual level of analysis and its acknowledgment of
5 the importance of unwritten commitments. As explained below, all of these three core
6 assumptions of PCT are core also to understanding OSS. Most importantly, though, PCT does
7 not impose *a priori* assumptions other than that people expect people to fulfil their obligations in
8 order for their relationship to be successful. Such theoretical openness lends itself well to the
9 exploration of an emerging phenomenon such as opensourcing.

10 Koh et al. (2004) note that most research on offshore outsourcing tends to focus on the customer
11 perspective, and argue for the importance of studying both the customer and supplier side of the
12 relationship. Interestingly, research to date on open source has focused inwards on investigating
13 the characteristics of the development process and projects, that is on the supplier side of the
14 relationship, and far less has been conducted on the customer side, in the sense of investigating
15 the consequences of the OSS phenomenon for organizations, for example. Building on Koh et
16 al.'s (2004) results, our research question is thus: *What are the critical customer and community*
17 *obligations in a successful opensourcing relationship?*

18 This study focuses on a phenomenon for which we have coined the term opensourcing, which is
19 the use of open source as an offshore sourcing strategy for the software development process in
20 organizations. One of the novel features of the study is that it considers both sides of the process
21 – that is, the organization which is ‘commissioning’ the opensourcing, and, in turn, the OSS
22 community who are doing the development work. The work presented here also builds on and
23 extends the work of Koh et al. (2004) in two significant ways. Firstly, it considers the unexplored

1 concept of opensourcing, as opposed to ‘traditional’ domestic outsourcing. Secondly, Koh et al.
2 (2004) focused on onshore outsourcing while this work considers *offshore* outsourcing, as
3 implicated by the opensourcing strategy, which is premised on the global nature of open source
4 development as confirmed in a wide range of studies on the issue and further discussed below
5 (e.g. Dempsey et al, 2002; Ghosh et al, 2002; Lakhani and Wolf, 2001; Robles-Martinez et al,
6 2001). To investigate this area, we adopted a two-phased research approach (see Appendix for a
7 schematic overview): In the first phase, we drew on the set of obligations identified and validated
8 by Koh et al. (2004) and conducted an exploratory qualitative multiple case study approach to
9 refine and elaborate those obligations that the customer must bear responsibility for and the
10 obligations that the OSS community must bear responsibility for in order for the opensourcing
11 arrangement to be successful. In the second phase, we conducted a quantitative survey study to
12 explore the validity of those obligations. Such a dual-phase approach, which has been used to
13 good effect in previous research (Kaplan and Duchon, 1988), allowed us to triangulate from
14 multiple data sources – an established strategy for improving reliability and validity (Yin, 1994).

15 The structure of the paper is as follows: In the next section, we define basic terminology to
16 distinguish the related concepts of outsourcing, offshoring and opensourcing. We then focus on
17 PCT, explaining why we have selected it for this study, and how it has been used. Following this,
18 our qualitative research approach is discussed, and the case study findings presented and
19 analysed. After that, the quantitative survey study is presented. Finally, the overall conclusions
20 and implications are presented. In the final section we also present some limitations of the study
21 and discuss possible future directions for research on opensourcing as global sourcing strategy.

BASIC TERMINOLOGY

1
2 While the terms offshoring and outsourcing are often used almost as synonyms¹, here we
3 distinguish between the two: Offshoring is about location – when an activity is offshored it is
4 performed in a different location to the main operation (which is then the onshore location).
5 Outsourcing, on the other hand, is about governance – when an activity is outsourced it is
6 performed by another organization, as opposed to ‘in-house’ by the organization itself.
7 Consequently, the two concepts are orthogonal – an activity can be performed either offshore or
8 onshore and can be performed in-house or be outsourced. Figure 1 shows the distinction and
9 relationship between the two concepts.

	In-house	Outsourced
Onshore	In-house (traditional model)	Subcontractor in the same locale
Offshore	Foreign branch of the same company	Subcontractor in a foreign locale

10 **Figure 1: Offshoring versus Outsourcing**

11 Open source software may be defined as software released under the terms of a license which
12 basically allows the licensee to use, modify and redistribute, either gratis or for a fee. Of
13 particular interest in this research, however, is the recent phenomenon of opensourcing as a

¹ We would like to acknowledge Erran Carmel who initially inspired our distinction between offshoring and outsourcing.

1 software development model. Similar to outsourcing, the OSS development model allows
2 companies to ‘subcontract’ development activities to another party. Since anyone (in principle)
3 can join any open source project, the development community can reasonably be assumed to be
4 global (Millar et al., 2005). The global nature of contributions to open source projects has been
5 confirmed in several studies. For example, Lakhani and Wolf (2001) conducted a survey on a
6 sample of almost 10,000 projects in SourceForge and found that 55% of respondents were from
7 outside North America. In another large-scale survey of almost 3,000 developers from across a
8 broad range of open source projects, Ghosh et al (2002) estimate that almost 86% of open source
9 developers are from outside North America. Another study which focused specifically on Linux
10 developers, for example, also reports that the majority of development is done by non North
11 American developers (Dempsey et al, 2002). Another relevant study is the one conducted by
12 Robles-Martinez et al (2001) of a number of popular open source projects which found that 62%
13 of developers were from outside North America. This latter study is particularly noteworthy in
14 that it the authors performed a series of exhaustive checks to identify the country of origin of
15 contributors. The global nature of the collaborative development found in all these studies
16 supports our classification of opensourcing as primarily in the bottom-right quadrant of Figure 1.

17 It is important also to emphasize that opensourcing as an offshore outsourcing practice is not
18 limited to releasing previously proprietary software under an open source licence and nurturing a
19 community around the product. This approach to opensourcing – as, for example, practiced by
20 IONA Technologies who released their flagship product Artix as open source under the name
21 Celtix – we refer to as the ‘*liberation*’ approach in this study. The reverse of this approach is
22 when a company evolves from an existing open source product, as exemplified by the MySQL

1 database management system and the JBoss J2EE application server (now owned by RedHat).
2 The latter can be referred to as the '*commercialization*' approach to opensourcing.

3 **A PCT PERSPECTIVE ON OPENSOURCING**

4 An early and influential contribution to psychological contract theory (PCT) was that of Argyris
5 (1960). PCT has since been widely used in studies on employment relationships (Anderson and
6 Chalk, 1998) and has featured in several IS research studies (e.g. Ang and Slaughter, 2001;
7 Piccoli and Ives, 2003; Koh et al., 2004; Raghu et al., 2004; Miranda and Kavan, 2005; Pavlou
8 and Gefen, 2005). A psychological contract represents 'the contractual parties' mental beliefs
9 and expectations about their mutual obligations in a contractual relationship, based on perceived
10 promises of a reciprocal exchange.' (Koh et al., 2004:357) Three aspects of the psychological
11 contract are particularly important in an outsourcing context: Firstly, the importance of mutuality
12 and reciprocity of obligations in a social context. This is distinct from the usual one-sided
13 perspective (customer *or* supplier) commonly adopted in outsourcing studies (Koh et al., 2004;
14 Goles and Chin, 2005). Secondly, psychological contracts are distinct from legal contracts.
15 Specifically they encompass people's beliefs about both written terms and unwritten implicit
16 terms. Thirdly, PCT promotes an individual level of analysis, focusing on the individuals' beliefs
17 and expectations in a social relationship. This is distinct from the more common inter-
18 organizational level of analysis.

19 These three issues are also central to the OSS concept. Firstly, mutuality and reciprocity are
20 critical to the success of the OSS development model. These values are effectively enshrined in
21 OSS development through the 'copyleft' terms found in OSS licenses, which decree that
22 software can be used, modified and redistributed provided subsequent modifications are made

1 freely available to others. Also, development is accomplished through the fulfilment of mutual
2 obligations with respect to the activities of coding, debugging, testing and documentation. One of
3 the most significant threats for the OSS movement has been suggested to be the ‘free rider’
4 phenomenon whereby someone profits from OSS without reciprocating, which thus contravenes
5 these values of reciprocity and fulfilment of mutual obligations (Von Hippel and Von Krogh,
6 2003).

7 Secondly, in relation to the PCT focus on psychological contracts (which differ from legal ones
8 as they encompass both written and unwritten terms), much OSS development is done in the
9 absence of any legal employment contract for developers. Also, the norms of how development
10 is conducted are both written and unwritten. Developers are expected to be familiar with written
11 rules and coding norms, for example, before they attempt to contribute (Feller and Fitzgerald,
12 2002). However, the unwritten rules must also be learned by developers over time. New recruits
13 to development ranks serve their apprenticeship in learning these unwritten rules and norms of
14 expected behaviour (Gorman, 2003; Raymond, 1999).

15 Finally, in relation to the PCT focus on the individual level of analysis, the signalling incentives
16 identified by Lerner and Tirole (2002) as the basic motivation for developers to contribute to
17 OSS projects (i.e. *career concerns* and *ego gratification*) apply primarily at the level of the
18 individual. Furthermore, since many OSS developers are not affiliated with any formal
19 organization, an individual level of analysis is to some extent required in any study of OSS
20 developers.

21 Similar to this study, Koh et al. (2004) used an initial qualitative case study approach to derive a
22 set of obligations that customers and suppliers need to fulfil in order to achieve a successful

1 outsourcing relationship. They followed this with a quantitative survey analysis to validate and
2 refine these factors. They recommend that their framework of obligations be applied in a context
3 in which clear and traditional authority structures do not exist. In this study, we drew on their
4 finalized set of validated obligations and sought to apply it in an open source context. In this
5 case, we use the ‘customer’ entity in the same sense as the Koh et al. study – i.e. as the
6 organization commissioning the opensourcing, seeking to grow a community around a product.
7 However, the ‘supplier’ entity becomes the open source community in our study. Below, we
8 consider the final refined set of obligations identified by Koh et al, and discuss their possible
9 relevance to an OSS context.

10 **Customer Obligations**

11 Koh et al. conclude that there are four specific obligations for which the customer must bear
12 responsibility and which are associated with outsourcing success. These are:

- 13 • Explicit and comprehensive requirements specifications for the services covered by the
14 outsourcing project
- 15 • Prompt payment to suppliers and no unreasonable withholding of payments
- 16 • Close project monitoring with active overseeing of project progress, attending project
17 meetings and regular discussions
- 18 • Project ownership to ensure that senior management provides strong leadership, support, and
19 commitment toward the project

20 **Explicit and Comprehensive Requirements Specifications**

21 At first glance, explicit and comprehensive requirements specifications might seem to be at odds
22 with the OSS development model which is predicated on the principle of a developer perceiving
23 ‘an itch worth scratching,’ to use Raymond’s (1999) memorable phrase, and thus not normally

1 associated with comprehensive requirements specifications. Also, OSS developers have typically
2 been users of the software being developed (Dinh-Trong and Bieman, 2004; Gacek and Arief,
3 2004; Mockus and Herbsleb, 2002), and the software was often targeted to a horizontal domain
4 (such as office desktop applications or software development tools). In such situations, clear
5 requirements specifications are not necessary as these are widely understood and internalized by
6 the individual developers. However, these aspects of the OSS development context are changing.
7 Increasingly OSS development is being purposively ‘steered’ as customers seek to stimulate OSS
8 development in specific vertical domains (such as automotive or telecoms) where a developer
9 may not perceive an itch worth scratching. In these development situations, the specifications are
10 not part of conventional software development knowledge and thus clear specifications are
11 becoming more important. This increasingly explicit formalization of specifications is already
12 evident in the commercially sponsored projects that are increasingly becoming a feature of the
13 OSS landscape.

14 **Prompt Payment to Suppliers and No Unreasonable Withholding of Payments**

15 Again, prompt payment to suppliers and no unreasonable withholding of payments might seem at
16 odds with the OSS model where actual monetary payment is often not a factor. However, recent
17 studies show that a significant part of OSS developers are indeed employed within professional
18 organizations (Lakhani and Wolf, 2001). Also, the Lerner and Tirole (2002) study illustrates that
19 ‘payment’ can come in forms other than mere monetary compensation. Many OSS developers
20 report the primary motivation as the rush they get from seeing their code in use and getting
21 prompt feedback from peers they really respect (Feller and Fitzgerald, 2002). This is in marked
22 contrast to the proprietary software development model, where developers may wait months or

1 even years to see their code in use. Thus, there is a sense in which prompt ‘payment’ can arise,
2 albeit in the form of surrogates such as peer feedback. When payments in the normal sense are
3 not part of the equation, then the issue of unreasonable withholding of (monetary) payments is
4 not likely to arise in the case of OSS.

5 **Close Project Monitoring with Active Overseeing of Project Progress**

6 Raymond’s (1999) characterization of the cathedral v. the bazaar to differentiate OSS
7 development from traditional development caused the perception that OSS development was
8 merely about developers following their own agenda developing in parallel in a spirit of
9 optimistic concurrency. However, Raymond’s characterization was based on a limited sample of
10 OSS projects which didn’t reflect the heterogeneity of the OSS development landscape even at
11 the time. In recent times, OSS development has become more formalized. This is evident in the
12 regular project meetings which are now a feature of a number of popular open source products,
13 such as the Apache conferences in the US and Europe, the Zope/Plone and PyPy development
14 project meetings, and the GNOME annual project conferences (German, 2004) which bring
15 together developers to coordinate and plan development.

16 **Project Ownership and Senior Management Leadership and Support**

17 The importance of strong management support has been verified in several studies of ICT
18 adoption (e.g. Agarwal, 2000; Chatterjee et al., 2002; Fichman, 2004; Gallivan, 2001). Project
19 ownership and senior management championship is undoubtedly critical for radical, high-risk
20 initiatives such as OSS deployment since it contravenes the traditional model where ongoing
21 support is legally guaranteed by a vendor. Indeed, top management championship is likely to

1 become even more important in the future as OSS adoption moves out of the domain of invisible
2 infrastructure systems to more visible, high-profile desktop systems and IS applications.

3 **Community Obligations**

4 Koh et al. (2004) identify five specific obligations for which the supplier must bear responsibility
5 and which are associated with outsourcing success. These are:

- 6 • Clear authority structures which delineate the decision-making rights and reporting structures
7 in the project, in terms of the roles and responsibilities of all parties involved
- 8 • Taking charge in terms of completing the job and solving problems independently, with
9 minimal customer involvement
- 10 • Effective human capital management in assigning high-quality staff to work on the project,
11 and seeking to minimize staff turnover during the project
- 12 • Building effective inter-organizational teams – investing time and effort to foster a good
13 working relationship among the team of customer and supplier staff working on the project
- 14 • Effective knowledge transfer in educating the customer in terms of the necessary skills,
15 knowledge, and expertise associated with using the outsourced system or service

16 **Clear Authority Structures**

17 In the absence of traditional organizational sanctions, some form of structure is necessary to
18 coordinate development. In many OSS projects, this takes the form of ‘benevolent dictatorship’
19 as initially suggested by Raymond (1999). Several studies of OSS development have detailed the
20 complex authority structure that evolves over time to ensure that all code contributions are vetted
21 and incorporated in a disciplined fashion (Mockus et al., 2002).

1 **Taking Charge and Solving Problems Independently**

2 OSS development has typically been characterized by the developers proactively taking charge,
3 solving problems independently with minimal customer involvement, and without the need for
4 traditional organizational sanctions to ensure development work is undertaken. Initially, OSS
5 developers did not engage in formal requirements analysis with customers (Scacchi, 2002), but
6 took direct responsibility for development decisions. Even though the OSS development process
7 is becoming more formalized (Fitzgerald, 2006), OSS developers are still more likely to retain a
8 strong sense of independence

9 **Effective Human Capital Management**

10 Research has also focused on the human capital management aspects of OSS (e.g. Hann et al.,
11 2002). This suggests that participation in OSS projects allows developers to gain highly
12 marketable technical skills which in turn can lead to higher earnings in the future, which is also
13 facilitated by the fact that the opportunity cost of participating in OSS development can be quite
14 low as developers can choose the amount of work they do and organize it to fit their own
15 personal timescale and agenda. Also, OSS developers have long been acknowledged as of high
16 quality (Raymond, 1999), and the loyalty of developers in the longevity of their commitment to
17 their development projects has been remarkable (Feller and Fitzgerald, 2002). Indeed, the
18 cardinal sin of OSS, that of project forking (whereby a project is divided in two or more streams,
19 each evolving the product in a different direction), is a strong community norm which acts
20 against developer turnover on projects.

1 **Building Effective Inter-Organizational Teams**

2 The particular characteristics of OSS position it as a good exemplar of the ‘whole-product’
3 concept of a market-driven business approach that seeks to deliver a complete solution to the
4 customer in terms of products and services (Moore, 1999). In this scenario, developers do the
5 coding while others complete the business model by adding sales and marketing services –
6 necessary activities but ones in which developers may not be interested. The OSS ‘whole-
7 product’ approach is also larger than a single company or software product or service. Indeed,
8 the network benefits of open source arise as a result of the size of the overall community and
9 ecosystem. Thus, an inter-organizational network of interested parties with complementary
10 capabilities can form an ecosystem to offer a professional product and service in an agile, bazaar-
11 friendly manner. Customer service requests can be routed to the most appropriate expert partner
12 in the network, perhaps even to the developer who wrote the actual code.

13 **Effective Knowledge Transfer in Educating the Customer**

14 Again, this is a topic that resonates well in the case of OSS on a number of aspects. In the cases
15 where the developer is also the user/customer, the issue of knowledge transfer does not arise.
16 However, the role of the user/customer is significantly elaborated in OSS as they can contribute
17 to debugging, testing, documentation, etc (Feller and Fitzgerald, 2002). Thus, a close working
18 relationship can emerge between developer and user.

1 **QUALITATIVE PHASE: IDENTIFYING PSYCHOLOGICAL CONTRACT**

2 **OBLIGATIONS IN OPENSOURCING**

3 **Research Method – Qualitative Phase**

4 Given the relative newness of the opensourcing concept, it is unsurprising that there is not a solid
5 research base to date on this phenomenon. Bearing this in mind, this study was concerned with
6 initially achieving an increased understanding of this phenomenon. Thus, analysis of a small
7 number of cases was deemed appropriate, as such ‘revelatory cases’ (Yin, 1994) may provide the
8 required rich insight. The primary case selected for the study was the Celtix project, an open
9 source Java Enterprise Service Bus (ESB) sponsored by IONA Technologies. This was
10 complemented with case study interviews of the DVTK (DICOM Validation Tool kit) project
11 initiated by Philips Medical Systems and the Morfeo project initiated by Telefonica.

12 **Background to Case Study Projects**

13 **IONA Technologies – The Celtix Project:** IONA Technologies, a NASDAQ-quoted company,
14 is headquartered in Dublin, Ireland, with U.S. headquarters in Waltham, Massachusetts and
15 offices worldwide. IONA was founded as a campus company at Trinity College Dublin in 1991,
16 and provides products and services to help organizations build B2B enterprise portals. IONA is
17 currently rated as the leading provider of standards-based platform middleware technology, with
18 more than 4,500 blue-chip enterprise customers worldwide, who use IONA products to address
19 large, complex application integration and achieve interoperability by means of a standards-
20 based, service-oriented architecture. In June 2005, Iona extended its business model to
21 incorporate open source by leading a community project to develop Celtix, an open source Java

1 ESB that will co-exist with Artix, the company's flagship integration product. The Celtix project
2 is hosted by an established open source community, ObjectWeb, who specialize in developing
3 open source middleware products. Most of ObjectWeb's members are small to medium-sized
4 enterprises based in continental Europe. The Celtix project has achieved an impressive
5 development productivity schedule, proceeding through four significant development milestones,
6 a beta release to a fully stable 1.0 release in just over 10 months.

7 **Philips Medical Systems – The DVTK Project:** Philips Medical Systems specializes in medical
8 devices and is part of the global technology organization Philips, headquartered in Eindhoven,
9 The Netherlands. DICOM (Digital Imaging and Communication in Medicine), initiated in 1993,
10 is a global standard for storing and processing medical images and is used in virtually all
11 hospitals worldwide. The DVTK product provides extra functionality for problem solving and
12 improving the quality of the DICOM interface. It began as a collaboration between Philips and
13 AGFA, and was released as open source under the GNU LGPL license in June 2005. Since then
14 a community has grown up around the product with contributions from independent developers
15 worldwide as well as developers in AGFA and Philips.

16 **Telefonica – the Morfeo Project:** Telefonica I+D, the R&D division of Spanish telecom
17 operator Telefonica, initiated the Morfeo project which operates in the area of Service Oriented
18 Architectures and aims to speed up the development of software standards in this area.
19 Telefonica I+D is the customer “engine”, releasing proprietary software components and
20 injecting resources into the community. The project has set up its own Morfeo implementation of
21 a SourceForge-like portal, where the development base of the sub-projects integrated in Morfeo
22 are hosted, including source repositories, binaries, mailing lists, bug and changes trackers, and
23 documentation.

1 **Data Collection and Analysis**

2 Data was gathered over a 12-month period from July 2005 to June 2006, and a number of
3 sources were drawn on (see Table 1). These ranged from workshops, to a series of interviews,
4 both formal face-to-face and informal telephone interviews. In keeping with Patton (1990), an
5 interview protocol guide was developed (see Appendix) based on the customer and community
6 obligations identified above. Marshall and Rossman (1989) identify the importance of being able
7 to gain entry to a company and maintain continuity of presence for as long as necessary. We
8 sought to achieve this by conducting initial interviews with the Chief Scientist at IONA (the
9 ‘customer’ in our study) and the Chairman of ObjectWeb (the supplier ‘community’).
10 Subsequently, in the Philips/DVTK and Telefonica/Morfeo projects, we conducted initial
11 interviews with the lead customer and community figures. These interviews served to give a
12 good strategic overview of the project and the high level obligations that were in place. These
13 lead individuals identified key figures in the projects and facilitated access to these interviewees.
14 Following this, as other key informants emerged during the interview process, support from
15 leadership in both the customer and community entities greatly facilitated achieving access. Most
16 comprehensive studies of open source developers up to now have relied on anonymous surveys
17 (e.g. Ghosh et al, 2002; Lakhani and Wolf, 2001), the studies by Hann et al. (2002) and Mockus
18 et al. (2002) being notable exceptions. This is caused in part by the difficulty in getting personal
19 access to key developers, but this study was notable in achieving such access.

20 The interview protocol guide approach was chosen as it is more comprehensive and systematic
21 for data collection than the purely conversational interview, and more flexible than the
22 standardized open-ended interview or the closed, fixed response interview. The duration of
23 interviews ranged between 30 minutes and 90 minutes. Interviews were recorded so as to

1 minimize data loss due to note-taking, and these recordings were subsequently coded. The
 2 interview guide was emailed to interviewees in advance to allow them an insight into the overall
 3 issues we wished to focus on. Informal follow-up interviews took place to clarify and refine
 4 issues that emerged following the interview transcription process. The interview transcripts
 5 comprised almost 60,000 words. These interviews were complemented by comprehensive
 6 reviews of documents and communications on the mailing lists, project wiki and web sites. Also,
 7 the project findings were presented to the participants and other researchers over a series of
 8 workshops with in-depth discussions feeding back into the analysis process (see Table 1).

Table 1: Data Sources for Qualitative Phase		
<i>Workshops</i>	<i>Interviews</i>	<i>Supplementary Sources</i>
<p>Sep 2005: Presentation and discussion of Celtix business model and strategy.</p> <p>Apr 2006: Workshop presentation on opensourcing strategy and discussion of DVTK and Morfeo projects.</p> <p>July 2006: Debriefing presentation of findings.</p>	<p>July 05 – June 2006: Interviews with</p> <ul style="list-style-type: none"> ○ Chief Scientist, IONA ○ Chairman, ObjectWeb ○ Admin, IONA ○ Open Source Program Director, IONA ○ Two Project Managers, IONA ○ Two Developers, ObjectWeb ○ Two Managers, Philips Medical Systems ○ Developer, DVTK ○ Manager, Telefonica ○ Developer, Morfeo 	<p>IONA and ObjectWeb maintain detailed and comprehensive web portals for the Celtix project. We also had access to mailing lists and project development wiki pages.</p> <p>Also, detailed web portals and mailing lists for DVTK and Morfeo projects were available.</p>

9

10 Qualitative analysis was undertaken using coding techniques proposed by Strauss and Corbin
 11 (1998), and exemplified by the research of Orlikowski (1993) and Baskerville and Pries-Heje
 12 (1999). Interviews were transcribed and then coded based on the seed categories (Calloway and

1 Ariav, 1991; Miles and Huberman, 1994) represented by the customer and community
2 obligations derived earlier, and analytical memos were written as patterns and themes emerged
3 from these field notes (see Appendix). This was combined with an open-ended analysis in which
4 we sought to break free from the initial obligations to allow for new categories to emerge. As a
5 result, our revised set of obligations, presented below, deviate but yet incorporate constructs
6 from the initially proposed set of obligations (cf. Klein and Myers, 1999). The analytical memos
7 provide a comprehensive grounding of these revised obligations in both existing theory (OSS
8 and outsourcing) and in our gathered empirical data. The overall structure of this research
9 process along with coding examples can be found in the Appendix.

10 The initial round of coding was done by the two authors – independently at first, followed by a
11 joint effort to compare and contrast findings and interpretations. We did not attempt to capture
12 inter-rater reliability. Instead, the results of this initial coding were then presented to two other
13 members of the research team, and issues which were unclear or on which there was no
14 unanimity between the initial two coders were discussed. Outstanding issues were clarified in
15 follow-up telephone interviews. Also, the emerging results of the qualitative research were
16 presented and discussed at three workshops attended by the authors and many of the
17 interviewees.

18 A problem that has been identified in relation to qualitative research is what is termed multiple
19 realities. This refers to the unavoidable fact that the understanding of reality is based on an
20 individual interpretation of the data, and that different individuals may interpret the same data in
21 different ways (Kaplan and Duchon, 1988). This problem was addressed in a number of ways.
22 Firstly, our approach to data analysis explicitly recognizes this problem of subjective data
23 interpretation, and to address it, uses rigorous coding and memoing processes which provide a

1 traceable, documented justification of the process by which research conclusions were reached,
2 thereby providing an audit trail of the process (Guba, 1981). Secondly, the method of venting
3 was used. This is a process whereby results and interpretations are discussed with professional
4 colleagues (Goetz and LeCompte, 1984). The findings were formally presented and discussed
5 with colleagues in detail on several occasions at practitioner/researcher workshops and
6 conferences. Also, in this study, IONA, ObjectWeb, Philips, Telefonica and Morfeo
7 representatives were active participants in an EU-funded research project led by the authors.
8 Thus, as findings were presented and discussed at the project workshops, quite detailed member-
9 checking of our interpretation of the findings was possible.

10 **Findings and Discussion – Qualitative Phase**

11 Here we discuss the obligations that emerged from the interviews. In our approach, we asked
12 both customer and community interviewees to discuss their perceptions of their own obligations,
13 and also the obligations they would expect from each other. Although the interviews were based
14 on the distinct obligations identified above, it quickly emerged that many of these obligations
15 were symmetrical.

16 Here we present the refined set of obligations based on the interview evidence.

17 **Refined Customer Obligations**

18 **Achieving Consensus on Development Roadmap:** Initially we expected that the customer
19 would provide an explicit requirements specification of required functionality, which would be
20 in keeping with the evolution of open source towards more formalized commercial phenomenon

1 (Fitzgerald, 2006). However, this was not in fact the case. Interviewees stressed the manner in
2 which requirements specification here differed from traditional outsourcing. Companies may
3 have a clear idea of what functionality they would like to see the community adding to the
4 product. However, there has to be consensus as to what functionality will be added. If a company
5 pushes its own agenda too much in seeking to control development, there can be problems. The
6 Celtix project manager within IONA expressed it well:

7 “A company cannot just go onto the mailing list or the community, and say ‘Can you guys
8 build this?’ When kicking off the project in the open source community, it’s about stating
9 the overall goal and the top-level requirements you are trying to achieve. Then it’s driven
10 by consensus. If people perceive you as driving your own agenda, then you will get
11 pushback on having things accepted.”

12 This again emphasizes the delicate equilibrium that must be maintained between acceptable OSS
13 community values and customer desire for value creation (Fitzgerald, 2006). Interestingly, it was
14 stressed that within the ecosystem formed around this mode of development, it was quite
15 permissible for customers to engage in more traditional outsourcing relationship directly with
16 some developers in the community, outside the strict remit of the opensourcing project.

17 Also, it was often the case that the initial development community contributions were to provide
18 something that existed within their repertoire. Thus, their contributions were not based on an
19 assessment of the most pressing commercial priority for the company; rather they considered
20 what they could offer based on their existing repertoire and expertise.

21 **Project Ownership:** The customer interviewees identified with the project ownership
22 obligation. One customer interviewee stressed the radical change in mindset represented by
23 opensourcing, suggesting that it represented a strategic initiative which differed from the normal

1 business model where developers could see that their salary was pretty much directly derived
2 from the sales of the commercial product that they developed. In the opensourcing model, their
3 work could appear to be benefiting the open source community, and not leading to an obvious
4 direct remuneration. Thus, top management championship was necessary to convince developers
5 that the strategy made business sense. In fact, the Celtix project could also grow the market
6 for other proprietary IONA products, enabling additional support contract revenue.

7 In keeping with demonstrating strong ownership and commitment to the project, IONA
8 established a full time position – Open Source Program Director. This person is responsible for
9 engaging with the community, and helps ensure that issues relevant to the open source
10 community receive prompt attention which helps ensure the quality of the software. Also, senior
11 management commitment can help ensure that R&D resources are provided. Interestingly,
12 however, there is a delicate equilibrium to be maintained here also. The Celtix project manager
13 at IONA suggested that if the project is seen as too much an IONA project, the developer
14 community may have less interest in getting involved. A manager at Philips Medical Systems
15 summarised this dilemma:

16 “This application is deeply used within Philips organisation. In several processes we
17 depend on it. And because it’s part of our product release, we want to stay in control of it.
18 So on the one hand you want to control the project, and then on the other hand you want to
19 be an open project and provide freedom for independent developers to join. And there’s a
20 little bit of conflict between these positions.”

21 **Marketing Project to Increase Visibility:** Several interviewees emphasized the need for the
22 company to market the attractiveness of the project and improve its visibility. This has a two-fold
23 purpose. Firstly, the development community will perceive their reputation has been improved

1 through involvement in a high-profile project with a high profile company. As a DVTK
2 community interviewee expressed it: “working with them [the company] is almost a sort of
3 professional honour, as it were”.

4 This helps ensure the vibrancy of the project, and can attract further developers to the
5 community, which cannot be taken for granted in an open source project. OSS is an emergent
6 phenomenon and very few projects to date have been deliberately started and nurtured to be
7 successful; rather some extremely successful ones have emerged over time, whereas others have
8 died off. Again, the Celtix project manager offered an interesting insight:

9 “There are a lot of open source projects which don’t go anywhere, even though they have
10 built good code. It also needs to be pushed so that it gets noticed and used by other
11 projects, documented and marketed. This is a big overhead, and commercial companies
12 have structures in place to help achieve that.”

13 Interviewees also suggested that the customer company could provide their expertise in relation
14 to software commercialization and productization in creating a professional OSS product and
15 subsequently marketing that product. This would involve a holistic approach and proactive
16 marketing to ensure that all who could usefully consume the product were made aware of it, and
17 could contribute. It was felt that a commercial vendor could usefully complement the OSS
18 community by providing this expertise. These efforts grow public awareness of the product
19 which can then create business consulting opportunities for members of the development
20 community. A DVTK community interviewee captured the essence of this well:

21 “What I am looking for is a multiplier. I am providing a piece of my work. I know they [the
22 company] already have a useful chunk of work. But adding my bit to their larger existing

1 work in a cooperative way creates something of greater multiplied use to everybody,
2 including myself.”

3 **Transparency and Close Project Monitoring:** Given that there is a strong external element in
4 opensourcing, customer interviewees stressed the necessity for clear project milestones and more
5 visibility about product releases. This was contrasted with traditional proprietary development
6 where internal milestones and actual release times are perhaps deliberately kept vague, whereas
7 the need for the customer to be clear about future project plans was important to the community.
8 More frequent product releases was identified as a by-product of the open source approach.

9 Also, it was suggested that the customer could not insist on a particular project monitoring
10 regime. Rather, different open source communities may have different norms and approaches in
11 this area, and the customer has to be flexible and prepared to adapt to the particular regime in
12 vogue in the particular open source community.

13 Related to transparency there was also a community expectation that the company have an open
14 process for accepting community contributions. Interestingly, the company can have unrealistic
15 expectations as to the level and significance of contributions. Indeed, one company manager
16 suggested his experience was that of “one significant contribution per month from the external
17 community.”

18 Related to this is the issue of licensing and Intellectual Property (IP). A senior community
19 interviewee with a background in commercial development suggested that it was sensible and
20 pragmatic to have a clear IP policy. They had requested that IONA release the Celtix project
21 under the Lesser General Public License (LGPL) and IONA agreed. The Celtix project manager
22 suggested that IONA were keen to embrace open source and build trust within the community,

1 and the choice of license is a key determinant for developers in deciding whether to participate in
2 a particular OSS project, and also for companies to adopt. Thus, the development community
3 tend to have a preference for a restrictive (so-called viral or reciprocal) GPL-style license which
4 safeguards their contributions due to its reciprocal nature. However, companies may tend
5 towards more permissive licenses which afford them greater control over the future of the
6 software. IONA had perceived the need to be even more open to other companies and
7 communities, and hence dual licensed Celtix under the less restrictive Eclipse Public License
8 also.

9 Related to this issue was the idea raised within Philips Medical Systems of creating a single legal
10 entity or foundation to represent the project. This trend towards legal incorporation has been
11 established in several OSS projects (O'Mahony, 2005).

12 **Creating a Sustainable Ecosystem – Customer Responsibilities:** It was seen as important that
13 both company and the community members strive towards the creation of a sustainable
14 ecosystem around the product. A vital factor here is to create an atmosphere of trust. Creating an
15 ecosystem and engendering trust is also facilitated by the fact that the project interaction tended
16 to be “very much techie to techie” as one interviewee put it. The Celtix project management
17 committee is chaired by an IONA Distinguished Engineer who would garner respect from the
18 technical OSS development community.

19 There was broad agreement from the community interviewees on this issue also. However, one
20 community interviewee stressed the importance of meaningful content in the feedback. While
21 promptness was appreciated, this evaporated if the content of the feedback was “empty”.

1 However, the “techie to techie” nature of the relationship helped ensure that feedback was
2 meaningful.

3 Another issue with significant implications for project management practices within the
4 customer company emerged in relation to development staff rotation. Normally, within
5 proprietary software development companies, development staff are rotated after a few months
6 onto other projects. However, because of the strong techie-to-techie interaction between
7 developers in the company and community, there was a strong pressure from the community that
8 developers would remain on the project for a lengthy period. This is in turn matched by the
9 company expectation that community developers show strong loyalty and commitment to the
10 project.

11 **Refined Community Obligations**

12 We now present the refined open source community obligations which emerged from the
13 qualitative phase.

14 **Clear and Democratic Authority Structure and Process Transparency:** Community
15 representatives identified clear and transparent authority structures as important – indeed, one
16 stressed that these are “not only important, but mandatory”. It was argued that since professional
17 involvement in OSS is increasing, the OSS development community is expected to show the
18 same level of quality and transparency as could be expected from any professional organization –
19 people need to understand how decisions are made. Indeed, professionalism is permeating
20 opensourcing projects since “everybody knows there are business reasons why people are there”,
21 as pointed out by an IONA project manager, and also emphasized by a DVTK community

1 member confessing that “Getting good karma is great, but it’s not my primary objective.” When
2 the opensourcing community involves a large number of ‘traditional professionals’, this is
3 particularly emphasized.

4 The customer interviewees also agreed that clear authority structures are important. However, in
5 opensourcing, authority structures are framed by a strong belief in democratic principles:

6 “It would be good to ensure that that the [democratic] process is working, but I’m not sure
7 that it is possible to see any authority structures other than that. It will always be shared
8 responsibility.” – Distinguished Engineer, IONA.

9 It was furthermore pointed out that such structures are important in two different respects.

10 Firstly, they provide consistency between projects, which means that developers can easily
11 contribute to more than one project. Contributing to several projects is not uncommon in OSS,
12 and with the increasing interest in the so-called ‘whole product approach’, which was discussed
13 above, this is expected to be increasingly important – a point stressed by the Open Source
14 Program Director at IONA. Secondly, they provide for consistent terminology within and across
15 projects which ensures people are “on the same page, and really focus on innovation”.

16 As a complement to clear authority structures, the lack of a traditional written outsourcing
17 contract means that an open source community must be clear also about their actual work
18 processes. Interestingly, this mirrors the “transparency – close project monitoring” obligation
19 that is expected on the part of the customer.

20 **Responsible and Innovative Attitude:** Although community and customer interviewees alike
21 believed it to be essential that the community takes responsibility and delivers on what has been
22 committed to, this outsourcing obligation becomes somewhat blurred in the opensourcing

1 context. In the Celtix project, for example, a large part of the overall development is done by
2 paid IONA employees. Hence, the customer has the power to manage part of the development
3 effort more directly than would be possible in a traditional offshore outsourcing context.
4 Although this may change as the Celtix project matures and the number of external contributions
5 increases, an IONA project manager explained that there currently seems to be a feeling that
6 “there will always be customer involvement”. However, he also suggested that a lot more
7 community independence was expected in the future and that responsibility follows naturally
8 with independence. A Manager at Philips Medical Systems captured this crisply, referring to a
9 potential problem with lack of written contracts:

10 “I think one of the biggest risks you have in open source development is when someone
11 says – yes I’m going to do that, and he doesn’t put any effort into it, nothing is done.”

12 This obligation also supports the customer obligation of ‘achieving consensus on the
13 development roadmap’, which assumes that the OSS community is innovative and contributes
14 constructively to this roadmap. As this will have a positive impact on the project, the community
15 is expected to help achieving a positive impact among (the customer’s) customers.

16 **Creating a Sustainable Ecosystem – Community Responsibilities:** As mentioned above, both
17 company and community members are expected to aim for a sustainable ecosystem.

18 Consequently, community members are expected to be loyal and committed to continued
19 involvement in the project, which, as mentioned above, is a marked feature of OSS projects. This
20 was confirmed by a Telefonica manager who explained that “a benefit that open source brings is
21 that people are more prepared to commit”, which also transfers across to customer developers
22 who “even though they’re working on the same technology now commit more personally”.

1 From a community perspective, it was suggested that the high-quality software associated with
2 the OSS model is an indication that the ‘human capital management’ is working. It was also
3 perceived that the quality of the code is a way to attract more business, which is essential for the
4 ecosystem to develop. As OSS is moving away from networks of individuals to networks of
5 companies, if a contributor earns a reputation for producing high-quality code, customers will
6 keep coming back for more. It is also the case that customers sometimes use the OSS model to
7 identify the best suppliers, who are then approached directly and contracted in a ‘traditional’
8 outsourcing model.

9 Opensourcing customers expect the OSS model to attract “high calibre people” who understand
10 the project domain very well without requiring additional training. The Open Source Program
11 Director of IONA even argued that it attracts a certain personality, with traits not necessarily
12 those traditionally associated with a “top-notch programmer”. In her view, people are attracted
13 by the OSS model because they want to “build something better”, they want to “get involved”,
14 and they want to “be part of a community” – in summary, “these are the kind of people that I
15 would want on my team, whether I was doing open source or not.”

16 Interestingly, OSS community members do not necessarily see themselves as suppliers
17 commissioned by a customer in the traditional sense. In fact, the open source community as
18 represented by ObjectWeb sees its members not as OSS developers but as “ecosystem
19 developers”. There seems to be a definite trend towards more organized open source
20 communities, such as that of ObjectWeb. Hence, facilitating effective inter-organizational teams
21 is to a large extent an intrinsic property of the opensourcing model. As indicated above, this is
22 due to the merging of customer and community into one ecosystem: “I don’t consider IONA as a
23 customer. Iona is a member” was how the situation was described by the Chairman ObjectWeb.

1 Opensourcing is thus not just about building good working relationships between customer and
2 supplier. It is about “ecosystem development”. Hence, although there are business reasons for
3 people to participate, there is a lot more collaboration than in traditional outsourcing:

4 “In a traditional market you don’t call up your competitor and be like, oh, well tell me what
5 your stuff does. But in open source you do.” – Open Source Program Director, IONA.

6 In opensourcing, the software developed is typically not aimed for end-users but is more likely to
7 be tools and infrastructure components. Consequently, the customer and community participants
8 typically share the same level of technical expertise – “it is mostly developer-to-developer
9 communication.” Therefore, there is no need for formal training. Instead, knowledge transfer is
10 happening continuously “from one research lab to another” within the ecosystem. This was
11 emphasized by the ObjectWeb Chairman who asserted that “I don’t speak about education or
12 anything like that, I speak about exchange between researchers”. This view was acknowledged
13 by the Open Source Program Director at IONA who referred to it as “cross pollination”.
14 According to a Project Manager at IONA, knowledge transfer was also facilitated by an early
15 and proactive focus on documentation as part of the work towards achieving consensus on a
16 development roadmap discussed above (a customer obligation).

17 **Summary of Identified Obligations**

18 Table 2 summarizes the refined list of obligations in the context of opensourcing.

Table 2: Summary of Refined Customer and OSS Community Obligations in Opensourcing
Customer obligations (i.e. obligations for which the customer must bear responsibility): (1) Achieving consensus on development roadmap: <ul style="list-style-type: none">▪ Not too forceful and dominant in pushing own agenda▪ Accept a general roadmap (vision) of future functionality rather than seeking a precise requirements specification

- (2) Project ownership:
 - Provide senior management commitment to the project
 - Provide R&D resources to further develop the project
 - Help improve the quality of the software
- (3) Marketing project to increase visibility
 - Provide professional expertise in relation to marketing and productizing the software
 - Seek to improve the reputation of the community of contributors
 - Provide a business opportunity for the community to use the product
- (4) Transparency and close project monitoring:
 - Transparent in plans for the future of the project
 - Open to outside contributions
 - Use an appropriate license to safeguard community contributions
- (5) Creating a sustainable ecosystem:
 - Seek to create trust in the relationship with the community
 - Engage in community-sustaining activities
 - Behave as a responsible member of the opensourcing ecosystem
 - Preserve continuity by keeping developers on projects for a longer period than the norm in proprietary software development

Community obligations (i.e. obligations for which the community must bear responsibility):

- (1) Clear and democratic authority structure and process transparency:
 - Provide a transparent authority structure to allow customer see the decision making process within the community
 - Behave as a professional team
- (2) Responsible and innovative attitude:
 - Take responsibility and deliver on what is committed to
 - Be creative and innovative in suggesting new functionality and directions for the project
 - Help achieve a positive impact among customers
- (3) Creating a sustainable ecosystem:
 - Offer high quality people who understand the project domain very well without requiring additional training
 - Exhibit loyalty and continued involvement in the project

1

2 **QUANTITATIVE PHASE: EXPLORING EFFECTS OF FULFILLED**

3 **OBLIGATIONS ON OPENSOURCING SUCCESS**

4 In the second phase of the study, we drew on the refined obligations from the qualitative phase to

5 explore the opensourcing phenomenon further by means of a quantitative survey.

1 **Opensourcing Survey**

2 A survey questionnaire was constructed to operationalize the obligations summarized in Table 2,
3 and also comprising factors related to open outsourcing success. Respondents were asked to
4 consider a particular opensourcing project of which they had experience and, on a Likert scale 1-
5 5, assess to what extent these obligation factors were met and to what extent the project was
6 perceived as successful. In order to improve instrument validity, we drew on the advice of Straub
7 (1989) and incorporated relevant items that had previously been validated by Koh et al (2004) in
8 a similar survey². This questionnaire was then pre-tested with four practitioners with experience
9 of opensourcing – two from the customer side and two from the community side. Some minor
10 modifications were made based on this feedback, and then we considered the issue of
11 constructing a survey sample.

12 Given the newness of this area of research – to our knowledge this is the first comprehensive
13 study of opensourcing – it was not possible to identify an overall population from which a
14 random sample could be drawn, an issue which has been discussed in relation to open source
15 survey research in the past (Ghosh et al, 2002). Hence, our study is by necessity quite
16 exploratory, and a purposeful sampling strategy was employed which sought to identify
17 information-rich cases using a form of operational construct sampling (Webb et al., 1966) – that
18 is, we sought to identify real world operational examples of the phenomenon of interest. Prior to
19 the study we were aware of a number of possible exemplars of opensourcing in addition to the
20 IONA/Celtix, Philips/DVTK and Telefonica/Morfeo projects. Two other exemplars of the

² We would like to acknowledge gratefully the assistance of Christine Koh, Soon Ang and Detmar Straub in providing a copy of their survey instrument.

1 liberation approach to opensourcing were Eiffel Studio and OpenAdaptor. In addition, to also
2 capture the commercialization approach we included two well-known examples, MySQL and
3 Canonical/Ubuntu. Overall, these projects represented good operational candidates of the
4 opensourcing phenomenon.

5 Again, we chose to focus primarily of key figures in these companies and communities and sent
6 them a web-link to the online questionnaire, requesting that they complete and forward to
7 persons they considered relevant. Such sampling is known as snowball or chain sampling
8 (Patton, 1990), and is used as a tactic to try ensure information-rich cases. This approach has also
9 been used successfully in a previous survey of open source (Ghosh et al, 2002). Reminders were
10 sent weekly until the cut-off date, resulting in a total of 218 responses. A small number of
11 responses were incomplete, and these were eliminated. Also, a number of individuals who had
12 participated in the interview phase responded to the survey. These too were eliminated so as to
13 avoid any primacy bias (Koh et al., 2004). This left a total of 207 usable responses.

14 Given that the survey was online and that we asked the principal contact points to further
15 distribute it to relevant partners in their networks, we cannot indicate the total number of survey
16 questionnaires sent to the population, as can be usually done with a postal survey. Therefore, the
17 issue of non-response bias was quite important. We investigated this through the use of late
18 respondents as surrogates for non-respondents (Wallace and Mellor, 1988). We compared the
19 responses of a random sample of 20 early respondents with a random sample of 20 late
20 respondents. There were no statistically significant differences (0.01 level). On visual inspection,
21 the most notable difference was that early respondents tended slightly more towards the extremes
22 in reporting their levels of satisfaction and dissatisfaction with the opensourcing experience than
23 late respondents. This is somewhat intuitive, perhaps suggesting that early respondents tended

1 more towards the extremes of satisfaction and dissatisfaction with the opensourcing
2 phenomenon, and may have felt more compelled to respond quickly, but again it must be
3 stressed that this was not a statistically significant result.

4 **Analysis of Survey Responses**

5 **Demographics**

6 Respondents were quite evenly distributed with 56% representing a company and 44%
7 representing an open source community. Also, 53% of the respondents reported more than 5-
8 years experience of open source, with fewer than 10% reporting less than 1-years experience of
9 open source. An interesting demographic also arises in relation to gender. Previous studies
10 (Ghosh et al, 2002; Lakhani and Wolf, 2001) indicate that over 98% of OSS developers are male.
11 In this study, just over 4% of the respondents were female, breaking down into 3% of community
12 respondents and 5% of company respondents. While this is not a major breakthrough in
13 redressing gender imbalance, it does suggest that the relative success of women in the broader IT
14 industry in general may at last percolate into the open source arena.

15 Table 3 summarizes the primary work location of respondents. The fact that 37 countries across
16 all continents are represented confirms the global offshoring nature of the opensourcing
17 phenomenon.. The high number of respondents from Ireland and Spain and to a lesser extent,
18 France and the Netherlands is indicative of our continued focus on the IONA/Celtix.
19 Telefonica/Morfeo and Philips/DVTK projects which were the basis of the qualitative research
20 phase, as the companies driving these projects are based in these countries.

Table 3: Respondents by Country	
<i>Percentage of Respondents</i>	<i>Countries</i>
<1%	Argentina, Australia, Austria, Bangladesh, Chile, Costa Rica, Denmark, Hong Kong, Hungary, India, Iran, Lithuania, Mexico, New Zealand, Norway, Peru, Philippines, Poland, Portugal, South Africa, Venezuela
1-5%	Belgium, Brazil, Canada, Finland, Germany, Sweden, Switzerland,
5-10%	France, Italy, Netherlands, Pakistan, United Kingdom, United States
>10%	Ireland, Spain

1

2 **Construct Validity and Reliability**

3 To explore the validity of our proposed company and company obligations, we conducted
4 principal component factor analysis (PCA). One of the assumptions for factor analysis is that
5 data be measured on a continuous scale, but for exploratory factor analysis, ordered categorical
6 data based on Likert scales, as in this study, is justifiable (Hutcheson and Sofroniou, 1999). Also,
7 factor analysis is based on correlation coefficients and thus is more reliable with larger sample
8 sizes, estimated at 150 cases (Tabachnik and Fidell, 1996). Our sample size of 207 cases exceeds
9 this threshold considerably. Finally, as a precaution in order to reinforce our sampling approach
10 we investigated sampling adequacy, which in the context of factor analysis, can be measured
11 using the KMO statistic (Kaiser, 1974). Kaiser suggests that KMO statistic values in excess of
12 0.9 are ‘marvellous’ while in excess of 0.8 is ‘meritorious’. The KMO statistic value for the
13 company and community obligation factors was 0.86 which clearly passes the sampling
14 adequacy test.

1 For the company obligations, PCA (with Varimax rotation, eigenvalues > 1.0 and 25
 2 iterations) of the company obligations produced a set of three higher-level component factors for
 3 the customer obligations, which explained 49% of the variance in the company sample. Three of
 4 the hypothesized customer obligation items with inadequate loadings (below 0.5) were dropped
 5 and excluded from further analysis. Factor loadings for the remaining items were all quite high
 6 (0.57 to 0.75). PCA (again with Varimax rotation, eigenvalues > 1.0 and 25 iterations) of the
 7 community obligations confirmed our original hypothesized set of seven factors for the
 8 community obligations, explaining which explained 46% of the variance in the community
 9 sample. Again, the factor loadings were quite high (0.58 to 0.76).

10 We also investigated the reliability of factors as outlined in Table 4 which shows
 11 descriptive statistics, correlations and Cronbach's α . The Cronbach α values (in parentheses on
 12 the diagonal) were all above the threshold level of .6 for construct reliability recommended for
 13 an exploratory study, apart from one of the customer obligations – Customer did not seek to
 14 dominate and control the process (.38). However, this factor was measured by just two items,
 15 and the Cronbach α calculation is sensitive to the number of items that comprise the factor, so
 16 this was retained. In terms of discriminant validity, none of the off-diagonal correlations were
 17 above .80 which can indicate extreme cases of multicollinearity (Ghiselli et al, 1981).

18 **Table 4 Means, Standard Deviations, Scale Reliabilities and Intercorrelations**

	Variable^a	Mean	SD	1	2	3	4	5
1	Create open company-community ecosystem	3.67	0.76	(.79)				
2	Provide professional business expertise	3.61	0.61	.42**	(.63)			
3	Did not seek to dominate	3.37	0.80	.33**	.12	(.38)		

4	and control process Community professional obligations	3.74	0.61	.42**	.27**	.28**	(.79)	
5	Opensourcing success	3.95	0.81	.54**	.37**	.28**	.57**	(.84)

1

2 ^a Items measured on a five-point scale (1 = strongly disagree, 5 = strongly agree)

3 Reliability coefficients in parentheses on diagonal.

4 ** Correlation is significant at the 0.01 level 2-tailed.

5 **Company versus Community Differences**

6 In our questionnaire, we asked both the company and community respondents to rate the
7 importance of fulfilment of each other’s obligations. We conducted Mann-Whitney tests to
8 investigate this. The analysis reveals a number of obligations on which company and
9 communities differed to a statistically significant extent ($p < .05$). The community were less
10 positive than company respondents in relation to the customer obligation factor *Was open to*
11 *outside contributions*. This suggests that the community were not fully persuaded that companies
12 were fulfilling their obligations in relation to accepting contributions. Typically, in open source
13 projects, a meritocracy emerges over time as developers ‘prove’ themselves and are allowed
14 commit contributions. The discrepancy between the company and community here suggests that
15 companies are cautious about accepting outside contributions into the code base, but presumably
16 a meritocratic structure could emerge in time which would see companies overcoming this
17 reluctance.

18 However, the company respondents also differed from community respondents ($p < 0.05$) in that
19 they were less positive about two community obligation factors: *Provided a transparent*
20 *authority structure to allow company see the decision making process within the community* and
21 *Helped improve public perception of the project*. This suggests that companies did not always
22 perceive communities as being open in relation to development decisions, and also that the

1 community might be remiss in relation to helping grow the public profile of the project. Given
2 that companies see open outsourcing as a means of growing product awareness, this is an
3 obvious point of potential conflict, as companies would expect the community to help in this
4 regard.

5 We also measured respondent opinion as to the success of opensourcing. However, there were no
6 statistically significant differences between the company and community respondents.

7 **Regression Analysis of Obligation Fulfilment and Opensourcing Success**

8 Finally, we conducted stepwise regression analysis to investigate the relationship between
9 fulfilment of company and community obligations and the perceived success of opensourcing.
10 One assumption behind the use of regression analysis is that the variables are measured on a
11 continuous scale and are normally distributed. The PCA factor analysis allowed us to generate
12 four composite factors for the company and community obligations which satisfied these
13 assumptions. These were used in stepwise regression. Our model suggested that all four
14 components were significantly associated with success, explaining over 47% of the variance ($F =$
15 $38.56, p < 0.001$). Table 5 provides the details of the regression analysis showing the standardized
16 beta coefficients and significance levels.

17 **Table 5 Regression Analysis**

Dependent Variable	Independent Variables	Model^a
---------------------------	------------------------------	--------------------------

Opensourcing success	Company Obligations	
	Create open company-community ecosystem	.35***
	Provide professional business expertise	.21***
	Did not seek to dominate and control process	.12*
	Community professional obligations	.38***
	<i>F</i>	38.56***
	<i>R</i> ²	.47
	Adjusted <i>R</i> ²	.46

1
2
3
4

^a Model statistics are standardised betas
* p< 0.05, ** p< 0.01, *** p<0.001

5

OVERALL CONCLUSIONS AND IMPLICATIONS

6 The aim of this study was to explore critical customer and community obligations that contribute
7 to success in an opensourcing relationship of offshore sourcing. To achieve this we adopted a
8 two-phased research approach consisting of an initial qualitative multiple case study followed by
9 a more quantitatively-oriented large-scale survey study. Adopting a psychological contract
10 perspective, the first phase used the outsourcing obligations identified by Koh et al. (2004),
11 adapted to the opensourcing context, as a basis for identifying customer and community
12 obligations that were associated with opensourcing success. These were then further explored by
13 means of a survey among customer and community representatives with experience of
14 opensourcing. Our results show that the fulfilment of certain customer and community
15 obligations is significantly associated with opensourcing success. Interestingly, these customer
16 and community obligations are partly symmetrical and complementary and thus represent a
17 tension between customer and community that needs to be managed for the opensourcing

1 arrangement to be successful. Our final set of obligations is shown in Table 6 and discussed
2 below where we consider the practical contributions of the study in terms of implications for
3 practice and also theoretical contributions with implications for research.

Table 6: Finalized set of customer and community high-level obligations with detailed responsibilities

<i>Customer Obligations</i>	<i>Community Obligations</i>
<p>Do not seek to dominate and control process</p> <ul style="list-style-type: none"> ▪ Not too forceful and dominant in pushing own agenda ▪ Accept a general roadmap (vision) of future functionality rather than seeking a precise requirements specification 	<p>Clear and democratic authority structure and process transparency</p> <ul style="list-style-type: none"> ▪ Provide a transparent authority structure to allow customer see the decision making process within the community ▪ Behave as a professional team
<p>Provide professional management and business expertise</p> <ul style="list-style-type: none"> ▪ Preserve continuity by keeping developers on projects for a longer period than the norm in proprietary software development ▪ Provide a business opportunity for the community to use the product ▪ Provide professional expertise in relation to marketing and productizing the software ▪ Provide R&D resources to further develop the project ▪ Provide senior management commitment to the project 	<p>Responsible and innovative attitude</p> <ul style="list-style-type: none"> ▪ Take responsibility and deliver on what is committed to ▪ Be creative and innovative in suggesting new functionality and directions for the project ▪ Help achieve a positive impact among customers
<p>Help establish an open and trusted ecosystem</p> <ul style="list-style-type: none"> ▪ Behave as a responsible member of the opensourcing ecosystem ▪ Open to outside contributions ▪ Transparent in plans for the future of the project ▪ Seek to create trust in the relationship with the community ▪ Engage in community-sustaining activities 	<p>Help establish a professional and sustainable ecosystem</p> <ul style="list-style-type: none"> ▪ Offer high quality people who understand the project domain very well without requiring additional training ▪ Exhibit loyalty and continued involvement in the project

Implications for Practice

1
2 Given that the opensourcing phenomenon has only recently emerged, there is no well-
3 documented ‘royal road’ which can guarantee success. Our study which sought to investigate the
4 extent to which the fulfilment of customer and community obligations are linked to success is
5 thus useful in that it identifies those obligations which appear most critical, and which can act as
6 a checklist of salient issues for customers and communities engaging in opensourcing. Also, the
7 symmetrical and complementary nature of the obligations illustrates how these obligations need
8 to be operationalized differently by the customer and community, but managed jointly, to ensure
9 achievement of the mutual goal. Here we identify the potentially problematic issues that
10 companies need to be aware of, and we also identify some new practices, or significant changes
11 to traditional practices which are required.

Complementariness of Obligations – Product Lifecycle

13 While, the customer must be prepared to compromise at all stages and not seek to dominate and
14 control the opensourcing process, the community must provide a transparent and democratic
15 authority structure with shared responsibility. Hence, while the customer must be tactful and
16 seek to embrace the OSS values of openness and democracy, the community must be able to
17 show the same level of management capability as would be expected from a traditional
18 outsourcing partner. It is important that the customer avoids pushing too hard its own agenda and
19 accepts that requirements evolve throughout the project with active input from the community.
20 To reciprocate, the community must make sure that its development process is communicated
21 and accepted by the customer. Furthermore, the opensourcing customer has to provide

1 complementary expertise in relation to product commercialization and marketing. In turn, the
2 community is expected to help improve public perception and awareness of the product.

3 When viewed holistically, a product lifecycle is evident. Firstly, the community is expected to
4 take responsibility and deliver on what is committed to. In turn, the customer can provide R&D
5 resources to complement and further develop innovative ideas, and also provide its professional
6 expertise in relation to marketing and productization of software, and in improving the visibility
7 of the product. Furthermore, the credibility provided by the company in marketing a product can
8 create a business opportunity for the product which community developers can then take
9 leverage.

10 **Differing Perceptions of Obligation Fulfilment**

11 Our study suggests that customer and community can have different perceptions of the extent to
12 which each has fulfilled its obligations. In particular, the community were significantly less
13 likely to concur with the customer on the extent to which the customer is open to outside
14 contributions. Further evidence of tension in this regard arose in the customer opinion expressed
15 above that there may be only one significant community contribution per month.

16 On the customer side, there was also significantly less satisfaction that the community were
17 providing a transparent authority structure in relation to the decision making process, and also
18 there was less satisfaction with the community efforts to help improve the public perception and
19 awareness of the product. Given that for many companies, one of the goals behind opensourcing
20 is to grow the potential market for the product, this is an important issue. Both customer and

1 community participants need to be aware of these sources of tension so they can be avoided if
2 possible.

3 **Achieving Balance between Value Creation and Community Values**

4 Furthermore, the customer must seek to reach consensus on the project monitoring system that
5 will be instituted, on trying to show leadership and ownership of the project but not so strongly
6 as to deter the development community. Overall, the customer must achieve that delicate
7 equilibrium between value creation in terms of a successful business model for itself while not
8 transgressing the OSS community values. This includes embracing OSS values of openness to be
9 trustworthy in the eyes of the community. On the other hand, the community must adopt a
10 professional attitude and approach in order to be taken seriously in the corporate world.

11 **Changes to Standard Development Management Practices**

12 We already referred to the need for management support on the company side to ensure buy-in to
13 a risky initiative where it appears as if the ‘crown jewels’ – the company’s software is being
14 given away for free. Standard operating practices that tend to apply in the customer company
15 may need to change. For example, more clarity is required in relation to software release
16 milestones so the community can plan their own business opportunities. Also, more frequent
17 product releases are likely rather than artificially separating product release functionality on the
18 basis of the revenue that can be generated which is often the case with proprietary software
19 product releases. Furthermore, the policy in companies of rotating developers onto different
20 projects after a period of several months may not be sustainable as company developers become
21 associated with the project in the OSS community. The community develop a trusted relationship

1 with a company developer and do not wish to see that relationship disturbed through the
2 company moving developers on to other projects.

3 **Global Recruitment – From Unknown to Known**

4 The study also reveals that outsourcing to the OSS community provides a significant opportunity
5 for companies to headhunt top developers, whereby community members become also
6 employees of the customer – hence moving from outsourcing to a largely unknown OSS
7 workforce towards recruitment of talented developers from the open source community. In this
8 study we see a move from ‘unknown unknowns’ as neither the customer nor the community are
9 known to each other, to a scenario of ‘known knowns’ as each gets to understand each other’s
10 position and build complementary skills. Essentially, the opensourcing model consists of a
11 company (as opensourcing customer) and a community of individual developers and other
12 companies with whom the customer interacts. This community provides potential for
13 development cost savings, recruitment opportunities, and the capability of increasing innovation.
14 Interestingly, while the community is expected to behave professionally and provide transparent
15 authority structures (thus reducing the ‘unknown factor’), innovation potential is primarily to be
16 expected from the ‘unknown’ part of the workforce. Hence, there are forces in the ecosystem
17 pulling in opposite directions: while cost-savings and innovation are facilitated by a large
18 ‘unknown’ workforce, recruitment of community developers by the customer, and trust-building
19 will tend to erode the unknown aspect.

Contribution to Research

We believe this is the first comprehensive study to focus on the opensourcing phenomenon. It thus represents an important step towards both (a) elaborating the IS offshoring research agenda to incorporate also this novel and unconventional approach to global sourcing and co-opetition, and (b) bringing the under-explored area of company-led OSS projects onto the OSS research agenda, in particular the *liberation* of hitherto proprietary software.

In terms of research method, the combination of a qualitative phase of in-depth case study interviews, complemented by a more quantitatively-oriented large-scale survey of the phenomenon has shown to be an effective way of triangulating from multiple sources to allow a comprehensive exploration of a novel phenomenon such as opensourcing. The rich understanding gained from in-depth qualitative data and analysis allowed us to propose a sophisticated initial model, and also to reinterpret and elaborate the results of the quantitative survey as in Table 6.

Our study is also notable in that most previous research on outsourcing has adopted a single perspective: the customer or the supplier (but most often focusing on the customer), the Koh et al. (2004) study being a notable exception, while this study considers both the customer and the community obligations. This is important since although the obligations are symmetrical to a large extent, they are also complementary, and there are differing emphases from each perspective, both sides of which must be fulfilled to achieve a successful opensourcing arrangement.

While we contribute to a cumulative research tradition by building on the research of Koh et al., we also identify the significant ways in which opensourcing differs from conventional outsourcing – the lack of a formal contract and requirements specification driven by the customer

1 as well as the absence of payment in the conventional sense, for example. Thus we position
2 opensourcing as an exemplar of global sourcing (Carmel and Tjia, 2005) in an offshoring
3 context, i.e. as a setting where sourcing occurs from external collaborators as well as from
4 offshore and onshore locations within the organization.

5 **Open Source – From Replication to Innovation**

6 Interestingly, open source is often assumed to be about replication rather than innovation.
7 However, creativity and innovation is stimulated by multi-disciplinary teams operating outside
8 conventional organization structures (Nonaka, 1991; Garvin 1993; Leonard-Barton 1995; Inkpen
9 1996; Goldman and Gabriel, 2005). Certain characteristics have been identified, including
10 *autonomy* (which forms the basis for self-organising and widens the possibility that individuals
11 will motivate themselves to form new knowledge); *creative chaos* (whereby individuals do not
12 have to follow organizational ‘rules’, but are challenged to investigate alternatives and rethink
13 assumptions); *information redundancy* (where individuals have information that goes beyond
14 their immediate needs for a particular task); and *requisite variety* (whereby the individuals
15 involved have the skill diversity to match the complexity and variety of the environment they
16 face). Interestingly, these characteristics appear to be present in OSS communities and can thus
17 be expected to play an important part in the success of opensourcing as offshore sourcing
18 practice. Previous studies of OSS have shown that about 40%³ of OSS developers are employed

³ Lakhani and Wolf (2001) estimate 40% and Jorgensen (2001) estimates 43% of OSS
developers are paid for their work.

1 within professional organizations, suggesting an ‘open’ community of about 60%. This latter
2 cohort may provide the creative and innovative spark as OSS loses its image of being merely
3 about imitation of proprietary products, and innovation becomes the defining feature. In this
4 study, the characteristics of the community facilitate innovation as community developers
5 operating outside traditional organisational constraints can identify new functionality and
6 develop the software in creative ways – innovation does indeed seem to happen elsewhere
7 (Goldman and Gabriel, 2005).

8 **Open Source – From Individual to Company**

9 The study reveals an ongoing shift from OSS as community of individual developers to OSS as
10 community of commercial organizations, primarily small and medium-sized enterprises,
11 operating as a symbiotic ecosystem in a spirit of co-opetition. Overall the goal seems to be to
12 create such a sustainable open and trusted ecosystem where customers and community
13 participants operate as equals with neither party dominating. Thus, in contrast to traditional
14 outsourcing, opensourcing is not primarily about commissioning software development to a
15 third-party, but rather about engaging in long-term collaborative activities leading towards a
16 sustainable ecosystem. Since many of the collaborators in this ecosystem are likely to be the
17 customer’s competitors, the collaboration is necessarily done in a spirit of co-opetition
18 (Brandenburger and Nalebuff, 1996). Both customer and community members have a shared
19 responsibility to actively contribute to the development and sustainability of the ecosystem.
20 Thus, research on ecosystems (e.g. Schnase et al., 2003) might be usefully applied to elaborate
21 this issue.

1 As mentioned above, much OSS research has focused inwards on the OSS phenomenon itself
2 and far less has been done on the organizational implications of OSS, and particularly on
3 company-led OSS projects in an opensourcing strategy as here. It is our hope that this study will
4 inspire researchers to investigate further this important area, both in terms of ‘liberation and
5 commercialization’ as well as other possible approaches and contexts.

6 REFERENCES

- 7 Agarwal, R. (2000) “Individual Acceptance of Information Technologies,” *Framing The*
8 *Domains of IT Management: Projecting the Future Through the Past*, R. W. Zmud (Ed.),
9 Cincinnati, OH: Pinnaflex Press, pp. 85-104.
- 10 Anderson, N., and Chalk, R. (1998) “The Psychological Contract in Retrospect and Prospect,”
11 *Journal of Organizational Behavior*, 19(S1), pp. 637-647.
- 12 Ang, S., and Slaughter, S. A. (2001) “Work Outcomes and Job Design for Contract versus
13 Permanent Information Systems Professionals on Software Development Teams,” *MIS*
14 *Quarterly*, 25(3), pp. 321-350.
- 15 Argyris, C. (1960) *Understanding Organizational Behaviour*, London: Tavistock Publications.
- 16 Aubert, B. A., Patry, M., and Rivard, S. (2005) “A Framework for Information Technology
17 Outsourcing Risk Management,” *The DATA BASE for Advances in Information Systems*,
18 36(4), pp. 9-28.
- 19 Baskerville, R., and Pries-Heje, J. (1999) “Grounded Action Research: A Method for
20 Understanding IT in Practice,” *Accounting, Management and Information Technologies*, 9
21 (1), pp. 1-23.
- 22 Brandenburger, A. M., and Nalebuff, B. J. (1996) “*Co-Opetition: A Revolution Mindset That*
23 *Combines Competition and Cooperation*,” Doubleday, New York, NY.
- 24 Calloway, L. and Ariav, G. (1991) Developing and using a qualitative methodology to study relationships
25 among designers and tools, in Nissen, H., Klein, H. and Hirschheim, R. (eds) (1991) *Information*
26 *Systems Research: Contemporary Approaches and Emergent Traditions*, Elsevier Publishers, North
27 Holland, 175-194.
- 28 Carmel, E., and Agarwal, R. (2001) “Tactical Approaches for Alleviating Distance in Global
29 Software Development,” *IEEE Software*, 18(2), pp. 22-29.
- 30 Carmel, E., and Tjia, P. (2005) *Offshoring Information Technology: Sourcing and Outsourcing*
31 *to a Global Workforce*, Cambridge, NY: Cambridge University Press.
- 32 Carmel, E. (1999) *Global Teams: Collaborating Across Borders and Time Zones*. Prentice-Hall,
33 Upper Saddle River: NJ.

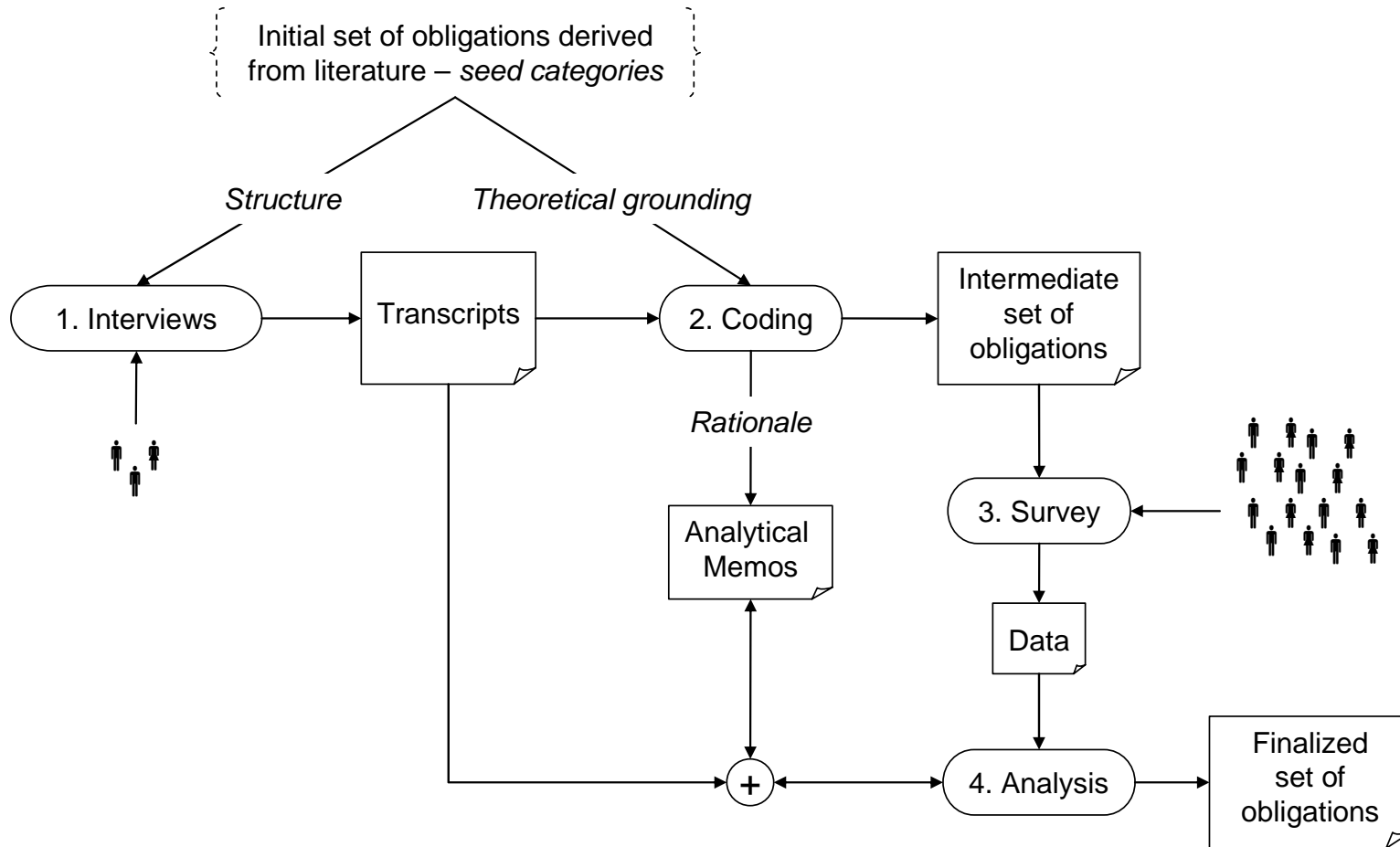
- 1 Carmel, E. (2006) "Building Your Information Systems from the Other Side of the World: How
2 Infosys Manages Time Zone Differences," *MISQ Executive*, 5(1), pp. 43-53.
- 3 Chatterjee, D., Grewal, R., and Sambamurthy, V. (2002) "Shaping Up for E-Commerce:
4 Institutional Enablers of the Organizational Assimilation of Web Technologies," *MIS*
5 *Quarterly*, 26(2): 65-89.
- 6 Cheon, M. J., Grover, V., and Teng, J. T. C. (1995) "Theoretical Perspectives on the
7 Outsourcing of Information Systems," *Journal of Information Technology*, 10(4), pp. 209-
8 219.
- 9 Dempsey, B., Weiss, D., Jones, P., and Greenberg, J. (2002) Who is an Open Source Software
10 Developer?," *Communications of the ACM*, vol. 45, pp. 67 - 72.
- 11 Dinh-Trong, T., and Bieman, J. M. (2004) "Open Source Software Development: A Case Study
12 of FreeBSD," in *Proceedings of the 10th International Symposium on Software Metrics*,
13 IEEE Computer Society.
- 14 Ebert, C., and De Neve, P. (2001) "Surviving Global Software Development," *IEEE Software*,
15 18(2), pp. 62 - 69.
- 16 Feller, J., and Fitzgerald, B. (2002) *Understanding Open Source Software Development*, London,
17 UK: Addison-Wesley.
- 18 Fichman, R. G. (2004) "Going Beyond the Dominant Paradigm for IT Innovation Research:
19 Emerging Concepts and Methods," *Journal of the Association for Information Systems*, 5(8).
- 20 Fitzgerald, B. (2006) "The Transformation of Open Source Software," *MIS Quarterly*, 30(3), pp.
21 587-598.
- 22 Fitzgerald, B., and Kenny, T. (2003) "Open Source Software in the Trenches: Lessons from a
23 Large Scale Implementation", in *Proceedings of 24th International Conference on*
24 *Information Systems*, Seattle, December 2003, pp. 316-326
- 25 Gacek, C. and Arief, B. (2004) "The Many Meanings of Open Source," *IEEE Software*, 21 (1),
26 34-40.
- 27 Gallivan, M. (2001) "Organizational Adoption and Assimilation of Complex Technological
28 Innovations: Development and Application of a New Framework," *Data Base*, 32(3), pp. 51-
29 85.
- 30 Garvin, D. (1993). "Building a Learning Organisation." *Harvard Business Review* 71(4): 78-91.
- 31 German, D. M. (2004) "The GNOME project: A Case Study of Open Source, Global Software
32 Development," *Software Process: Improvement and Practice*, 8(4), pp. 201-215.
- 33 Ghiselli, E, Campbell, J, Zedeck, S. (1981) *Measurement Theory for the Behavioral Sciences*,
34 W.H. Freeman, San Fransisco.
- 35 Ghosh, R, Glott, R, Kreiger, B and Robles-Martinez, G (2002) The free/libre/open source
36 software developers survey, available from <http://www.infomonics.nl/floss/report/>
- 37 Goetz, J., and LeCompte, D. (1984) *Ethnography and Qualitative Design in Educational*
38 *Research*, Academic Press, Orlando.

- 1 Goles, T. and Chin, W. W. (2005) "Information Systems Outsourcing Relationship Factors:
2 Detailed Conceptualization and Initial Evidence," *The DATA BASE for Advances in*
3 *Information Systems*, 36(4), pp. 47-67.
- 4 Goldman, R., and Gabriel, R. P. (2005) "*Innovation Happens Elsewhere: Open Source as*
5 *Business Strategy*," Morgan Kauffman Publishers, San Fransisco, CA.
- 6 Gottschalk, P., and Solli-Sæther, H. (2005) "Critical Success Factors from IT Outsourcing
7 Theories: An Empirical Study," *Industrial Management & Data Systems*, 105(6), pp. 685-
8 702.
- 9 Gorman, M. (2003) *A Design, Implementation and Algorithm Analysis of a Virtual Memory*
10 *System for Linux*, (personal communication).
- 11 Grover, V., Teng, J. T. C., and Fiedler, K. D. (2002) "Investigating the Role of Information
12 Technology in Building Buyer-Supplier Relationships," *Journal of the Association for*
13 *Information Systems*, 3, pp. 217-245
- 14 Guba, E. (1981) "Criteria for assessing the trustworthiness of naturalistic inquiries," *Educational*
15 *Communication and Technology*, 29, pp. 75-92.
- 16 Hann, I., Roberts, J., Slaughter, S., and Fielding, R. (2002) *Why Do Developers Contribute to*
17 *Open Source Projects? First Evidence of Economic Incentives*, Paper presented at the 2nd
18 Workshop on Open Source Software Engineering, Orlando, FL.
- 19 Herbsleb, J. D., and Grinter, R. E. (1999) "Splitting the Organization and Integrating the Code:
20 Conway's Law Revisited," *Proc. 21st International Conference on Software Engineering*,
21 Los Angeles, California, United States, 1999.
- 22 Hutcheson, G and Sofroniou, N. 1999. *The Multivariate Social Scientist*, Sage Publications, London
- 23 Inkpen, A. C. (1996). "Creating knowledge through collaboration." *California Management*
24 *Review* 39(1): 123-140.
- 25 Jorgensen, N (2001) Putting it All in the Trunk: Incremental Software Development in the
26 FreeBSD Open Source Project, *Information Systems Journal*, Vol 11, No 4, pp 142-157.
- 27 Kaiser, H.F. (1974) An index of factorial simplicity. *Psychometrika*, 39, 31-36.
- 28 Kaplan, B., and Duchon, D. (1988) "Combining Qualitative and Quantitative Methods in IS
29 Research: A Case Study," *MIS Quarterly*, 12(4), pp. 571-587.
- 30 Klein, H. K. and Myers, M. D. (1999). "A Set of Principles for Conducting and Evaluating Interpretive
31 Field Studies in Information Systems," *MIS Quarterly*, 23 (1), pp. 67-94.
- 32 Koh, C., Ang, S., and Straub, D. W. (2004) "IT Outsourcing Success: A Psychological Contract
33 Perspective," *Information Systems Research*, 15(4), pp. 356-373.
- 34 Lakhani, K and Wolf, R (2001) Does free software mean free labor? Characteristics of
35 participants in in FOSS communities, available at [www.osdn.com/bcg/bcg-](http://www.osdn.com/bcg/bcg-0.73/BCGHackerSurveyv0-73.html)
36 [0.73/BCGHackerSurveyv0-73.html](http://www.osdn.com/bcg/bcg-0.73/BCGHackerSurveyv0-73.html)
- 37 Leonard-Barton, D. (1995). *Wellsprings of Knowledge: Building and Sustaining the Sources of*
38 *Innovation*. Boston, Harvard Business School Press.

- 1 Lerner, J., and Tirole, J. (2002) "Some Simple Economics of Open Source," *The Journal of*
2 *Industrial Economics*, 50(2), pp. 197-234.
- 3 Marshall, C., and Rossman, G. (1989) *Designing Qualitative Research*, Sage Publications,
4 California.
- 5 Miles, M. and Huberman, A. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd ed., Sage,
6 Thousand Oaks, CA.
- 7 Millar, C., Choi, C. J., Russell, E. T., and Kim, J. B. (2005) "Open Source Communities: An
8 Integrally Informed Approach," *Journal of Organizational Change Management*, 18(3), pp.
9 259-268.
- 10 Miranda, S. M., and Kavan, C. B. (2005) "Moments of Governance in IS Outsourcing:
11 Conceptualizing Effects of Contracts on Value Capture and Creation," *Journal of*
12 *Information Technology*, 20(3), pp. 152-169.
- 13 Mockus, A., and Herbsleb, J. D. (2002) "Why Not Improve Coordination in Distributed Software
14 Development by Stealing Good Ideas from Open Source?" in *Meeting Challenges and*
15 *Surviving Success: The 2nd Workshop on Open Source Software Engineering*, pp. 19-25,
16 <http://opensource.ucc.ie/icse2002/MockusGerbsleb.pdf>
- 17 Mockus, A., Fielding, R. T., and Herbsleb, J. D. (2002) "Two Case Studies of Open Source
18 Software Development: Apache and Mozilla," *Transactions on Software Engineering*
19 *Methodology*, 11 (3), pp. 309-346.
- 20 Moore, G (1999) *Crossing the Chasm*, Harper, NY
- 21 Nonaka, I. (1991). "The Knowledge-Creating Company," *Harvard Business Review* 69(6): 96-
22 104.
- 23 O'Mahony, S (2005) Non-Profit Foundations and their Role in Community-Firm Software
24 Collaboration, in Feller, J, Fitzgerald, B, Hissam, S, and Lakhani, K. (2005) (Eds)
25 *Perspectives on Free and Open Source Software*, MIT Press, Cambridge, pp. 393-414.
- 26 Orlikowski, W. J. (1993). "CASE Tools as Organizational Change: Investigating Incremental
27 and Radical Changes in Systems," *MIS Quarterly* (17):3, pp 309-340.
- 28 Patton, M. Q. (1990) *Qualitative Evaluation and Research Methods*, 2nd ed., Newbury Park,
29 CA: SAGE.
- 30 Pavlou, P. A., and Gefen, D. (2005) "Psychological Contract Violation in Online Marketplaces:
31 Antecedents, Consequences, and Moderating Role," *Information Systems Research*, 16(4),
32 pp. 372-399.
- 33 Piccoli, G., and Ives, B. (2003) "Trust and the Unintended Effects of Behavior Control in Virtual
34 Teams," *MIS Quarterly*, 27(3), pp. 365-395.
- 35 Raghu, T. S., Jayaraman, B., and Rao, H. R. (2004) "Toward an Integration of Agent- and
36 Activity-Centric Approaches in Organizational Process Modeling: Incorporating Incentive
37 Mechanisms," *Information Systems Research*, 15(4), pp. 316-335.
- 38 Raymond, E. S. (1999) *The Cathedral and the Bazaar: Musings on Linux and Open Source by*
39 *an Accidental Revolutionary*, Sebastopol, CA: O'Reilly.

- 1 Robles-Martinez, G, Scheider, G, Tretkowski, I and Weber, N. (2001) Who is doing it?
2 <http://widi.berlios.de/paper/study.html>
- 3 Scacchi, W. (2002) "Understanding the Requirements for Developing Open Source Software
4 Systems," *IEE Proceedings–Software*, 149(1), pp. 24-39.
- 5 Schnase, J. L., Cushing, J., Frame, M., Frondorf, A., Landis, E., Maier, D., and Silberschatz, A.
6 (2003) "Information Technology Challenges of Biodiversity and Ecosystems Informatics,"
7 *Information Systems*, 28(4), pp. 339-345.
- 8 Straub, D (1989) "Validating Instruments in MIS Research," *MIS Quarterly* 13(2)147-169.
- 9 Strauss, A., and Corbin, J. (1998) *Basics of Qualitative Research: Grounded Theory Procedures
10 and Techniques*, 2nd ed., Sage, California.
- 11 Tabachnik, B and Fidell, L (1996) *Using Multivariate Statistics* (3rd Edition), Harper Collins, New York.
- 12 United Nations (2004) World Investment Report 2004 – The Shift Towards Services, New York
13 and Geneva.
- 14 Von Hippel, E., and Von Krogh, G. (2003) "Open Source Software and the "Private-Collective"
15 Innovation Model: Issues for Organization Science," *Organization Science*, 14(2), pp. 209-
16 223.
- 17 Wallace, R and Mellor, C. 1988. Non-response bias in mail accounting surveys: A pedagogical
18 note. *British Accounting Review* 20(2) 131-139.
- 19 Wang, E. T. G. (2002) "Transaction Attributes and Software Outsourcing Success: An Empirical
20 Investigation of Transaction Cost Theory," *Information Systems Journal*, 12(2), pp. 153-181.
- 21 Webb, E, Campbell, D, Schartz, R and Sechrest, L (1996) *Unobtrusive Measures: Non-reactive Research
22 in the Social Sciences*, Rand McNally, Chicago.
- 23 Wheeler, D. (2004) *Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)?
24 Look at the Numbers!* Online: http://www.dwheeler.com/oss_fs_why.html
- 25 Yin, R. K. (1994) *Case Study Research: Design and Methods*, 2nd ed., Thousand Oaks, CA:
26 SAGE.
- 27

1 APPENDIX



2

3

Figure A-1: Schematic overview of research process

4

Table A-1: Interview Guide used in Qualitative Phase

<i>Customer obligations</i>	<i>Community obligations</i>
<p>(1) <i>Explicit and comprehensive requirements specifications for the services covered by the outsourcing project</i> – Although initially, requirements specifications were not part of the OSS landscape, this appears to be increasingly the case</p> <p>(2) <i>Prompt feedback to supplier community with no unreasonable delays</i> – Although payment in the monetary sense is usually (but not always) not a factor in OSS development, prompt feedback by peer developers and users is critical</p> <p>(3) <i>Close project monitoring with active overseeing of project progress, attending project meetings and regular discussions</i> – Again, project monitoring is increasingly a part of the more commercially focused OSS development process</p> <p>(4) <i>Project ownership to ensure that senior management provides strong leadership, support, and commitment toward the project</i> – Given the high risk, radical initiative that OSS deployment represents, strong project ownership and management championship may be necessary</p> <p>(...) <i>Other possible obligations</i></p>	<p>(1) <i>Clear authority structures which delineate the decision-making rights and reporting structures in the project</i> – Given the absence of normal organizational authority, the ‘benevolent dictatorship’ and meritocracy in OSS projects is necessary</p> <p>(2) <i>Taking charge in terms of completing the job and solving problems independently, with minimal customer involvement</i> – OSS development has traditionally been characterized by developer independence and prompt problem solving, although customer involvement in terms of user feedback has been a marked feature</p> <p>(3) <i>Effective human capital management in assigning high-quality staff to work on the project, and seeking to minimize staff turnover during the project</i> – OSS developers are acknowledged to be high quality, and exhibit strong loyalty to projects due in part to avoidance of project forking and the freedom to choose what development tasks to work on</p> <p>(4) <i>Building effective inter-organizational teams – investing time and effort to foster a good working relationship in the customer and community project team</i> – Community networks of OSS companies are becoming a common mode of delivering ‘whole product’ OSS offerings to customers</p> <p>(5) <i>Effective knowledge transfer in educating the customer in the skills, knowledge, and expertise associated with using the outsourced system or service</i> – The user/developer relationship is very close in OSS thus facilitating knowledge transfer</p> <p>(...) <i>Other possible obligations</i></p>

Table A-2: Coding Examples			
<i>Initial Set of Obligations</i>	<i>Transcripts (excerpts)</i>	<i>Analytical Memos</i>	<i>Emerging Set of Obligations</i>
...
Explicit and comprehensive requirements specifications	When kicking off the project in the open source community, it's about stating the overall goal and <i>the top-level requirements</i> you are trying to achieve. Then it's <i>driven by consensus</i> .	Although initially, requirements specs were not part of OSS, this appears to be increasingly the case.	Achieving consensus on development roadmap
		High-level requirements evolve.	
...
N/A ⁴	There are a lot of open source projects which don't go anywhere, even though they have built good code. It also <i>needs to be pushed</i> so that it gets noticed and used by other projects, documented and marketed. This is a big overhead, and <i>commercial companies have structures in place</i> to help achieve that.	Companies have marketing experience and resources that OSS communities typically lack	Marketing project to increase visibility
	What I am looking for is a multiplier. I am providing a piece of my work. I know they already have a useful chunk of work. But adding my bit to their larger existing work in a <i>cooperative way creates something of greater multiplied use</i> to everybody, including myself.		
...

⁴ Note that there is not a one-to-one mapping between the two sets of obligations, which is a natural consequence of the open-ended analysis in which new obligations were also allowed to emerge.