
Collectivism, creativity, competition, and control in open source software development: reflections on the emergent governance of the SPDX[®] working group

Matt Germonprez*

College of Information Science and Technology,
University of Nebraska at Omaha,
Omaha, NE 68182, USA
E-mail: germonprez@gmail.com
*Corresponding author

Julie E. Kendall and Kenneth E. Kendall

Rutgers School of Business Camden,
Rutgers University,
Camden, NJ 08102, USA
E-mail: julie@thekendalls.org
E-mail: ken@thekendalls.org

Brett Young

Georgia Gwinnett College,
Management Information Systems,
1000 University Center Lane,
Lawrenceville, GA 30043, USA
E-mail: brett@brettwyong.org

Abstract: For over a decade, we assumed that software developers had a choice between the cathedral and the bazaar: developers could choose to develop propriety software in a structural hierarchy of planned development, testing, and documentation or in a tumultuous and exciting, quick release open source marketplace. We are finding that these assumptions are no longer valid. Corporate members' participation in open source software development and greater attention to compliance with open source licenses has signalled the need for new metaphors for open source software design and development. In this study, we use participant observation and functional discourse analysis to determine which metaphors are useful and seek evidence for these metaphors in the emergent governance structure of the Software Package Data Exchange (SPDX[®]) working group, which is one of the important players in the future of open source development. We observed three governing structures for the SPDX working group: meritocracy, adhocracy, and family/republic and reflected back to the SPDX working group the need to pay greater attention to their governance structures to help in future strategic decisions. We conclude that open source developers now have a choice among collectivism, creativity, competition, and control.

Keywords: open source software; software development; metaphors; governance; cathedral and bazaar; collectivism; creativity; competition; control; bureaucracy; adhocracy; republic; meritocracy; Linux; SPDX.

Reference to this paper should be made as follows: Germonprez, M., Kendall, J.E., Kendall, K.E. and Young, B. (2014) 'Collectivism, creativity, competition, and control in open source software development: reflections on the emergent governance of the SPDX[®] working group', *Int. J. Information Systems and Management*, Vol. 1, Nos. 1/2, pp.125–145.

Biographical notes: Matt Germonprez is currently a faculty member at the University of Nebraska at Omaha. Prior to joining UNO, he was a faculty member at UW-Eau Claire and Case Western Reserve University. His research focuses on theory and method development and investigation with particular focus on emerging and tailorable technologies. His work has been funded by the NSF and accepted in *MIS Quarterly*, *The Journal of the Association for Information Systems*, *ISJ*, *Information & Organisation*, and *CAIS*.

Julie E. Kendall is a Professor of Management in the School of Business-Camden, Rutgers University. She is a Fellow of the Decision Sciences Institute and Past Chair of IFIP Working Group 8.2. She is the Vice President for the Association for Information Systems. She is an Associate Editor for CAIS. She is a co-author of the text *Systems Analysis and Design*, now in its 9th edition.

Kenneth E. Kendall is a Distinguished Professor of Management in the School of Business-Camden, Rutgers University. He is one of the founders of the International Conference on Information Systems (ICIS) and Fellow of the Decision Sciences Institute. He is the past President of DSI and was awarded the Silver Core from IFIP. He was named Educator of the Year in IS for 2010 by the Association for Information Technology Professionals.

Brett Young is an Assistant Professor of MIS at Georgia Gwinnett College. He is also a researcher on the NSF-funded 'Organizational Participation in Open Communities Project'. His research reflects his interests in stakeholder perceptions of technology and user contributions within online communities. He is a co-Chair for the 2013 International Conference on Engaged Management Scholarship and received his PhD in 2011 in Computer Information Systems from Georgia State University.

1 Introduction

According to researchers of software development (Kelty, 2008; Lerner and Tirole, 2005), it has become exceedingly apparent that the world of open source software development has changed dramatically over the last couple of decades. In simpler times, the term open source software was synonymous with free software, but as time passed, this ideal has changed significantly. While some open source software has a cost of zero, much of open source software (for example, MySQL Enterprise from Oracle) has become part of for-profit, commercial products.

Therefore, it is possible, and even likely, that developers employed by for-profit corporations can develop a product, enhance the product with open source code, and

deliver a commercial software product that is part proprietary and part open source (Open Source Initiative, n.d.). According to extensive interviews with 80 different managers, developers, and executives participating in the Linux open source community from 40 different organisations, this is happening now in a significant way (Germonprez et al., 2013). In a recent article, Jonathan Corbet reported that 75% of the contributions to the Linux kernel are being done by paid developers for major corporations (Kidman, 2010).

Open source is not the 'Wild West' of software development anymore. We are no longer living in a world where coders write and publish free or shareware software without rules or worries (Germonprez et al., 2014). The world has now become a place where corporations who regularly produce propriety software now participate in the 'shared design' of software (Germonprez et al., 2013). This new approach to the development of open source software means that there will be new metaphors for software development.

It is our goal in this paper to explore the changes that have occurred in ways that open source software is developed cooperatively between paid corporate developers and open source communities. We describe those changes by identifying the currently used metaphors for cooperating in open source projects, and then link those to earlier work (Kendall and Kendall, 2010) revealing the influence of metaphors on governmental structures on the state level for deterring or sustaining information systems.

We use the illustration of the ongoing governing processes being designed by the newly formed, Linux Foundation sponsored Software Package Data Exchange (SPDX[®]) specification working group. We identify the metaphors the working group has embedded in recent governance documents (Linux Foundation, 2012), discuss what they mean in light of earlier research findings, and discuss what governance structures and metaphors will serve to stabilise the working group, to attract other members, and to fulfil their other goals (Linux Foundation, 2014).

2 Relevant open source software research

There are numerous studies of open source software development, including those on governance (Markus, 2007; O'Mahony and Ferraro, 2007), studies of corporate open source projects (Daniel et al., 2011; Gurbani et al., 2010), studies on a variety of specific open source projects including Linux (Raymond, 2000, 2001), and Apache (Dueñas et al., 2007; Fielding, 1999; Mockus et al., 2000; Roberts et al., 2006) and Apache compared against Mozilla (Mockus et al., 2002), and even studies of what clout a country might exert on a firm's decision to adopt open source software (Qu et al., 2011). Adding to the knowledgebase on open source are also online publications, downloadable software, lists of open source projects to join, and webinars conducted by the non-profit foundations such as the Linux Foundation (<http://www.linuxfoundation.org>) and the Apache Software Foundation (<http://www.apache.org>).

Recently, two in-depth and helpful literature reviews of open source research were published. The first to appear was one by Aksulu and Wade (2010), in which they reviewed 618 peer-reviewed research articles in order to gain insight into where open source research has been. Their resulting holistic framework of open source research includes not only current research, but also points out which research areas are lacking, where research areas overlap, and proposes ideas for potentially fruitful areas of open

source research. In terms of software licensing and intellectual property rights, Aksulu and Wade (2010, p.23) note:

The present body of OSR [open source research] mostly covers two endpoints of the intellectual property rights spectrum: licensing of end products (technology outputs), and the effect of licensing and the strength of intellectual property regimes on community participation (people inputs).

They note that many areas of open source research fall in between these two positions, and that they may merit further investigation. It is our intent to position our research in this paper partially within that spectrum.

Aksulu and Wade (2010) also examine research on governance, which they define as studies that “look at how communities producing collective goods govern themselves”, and go on to say the research literature includes a variety of definitions of governance as well as “community hierarchies, conflict management, and decision making mechanisms” (p.635). They categorise work from 36 different authors or author-teams in the governance grouping alone.

In another recent, comprehensive review and appraisal of open source research, Crowston et al. (2012) identify, categorise, classify, and analyse research in two ‘waves’ examining studies published by early 2006 and then again in 2009. They note the ever-increasing body of open source research, and also note that it is widely dispersed in the literature of many fields, making it difficult to stay abreast with, or even to be comprehensive in any solitary review of open source research literature. The authors use their review to suggest potential future research directions. Of particular interest for us in our research is their encouragement of further study in actual firms.

Crowston et al. (2012, pp.28–29) state:

Despite the increasing commercialization of FLOSS, there are not many studies of the details of firm participation in projects. This lacuna may be due to the relative difficulty of obtaining data from firms. But since one of the often cited reasons for studying FLOSS is the potential for adapting FLOSS practices to proprietary production environments, additional research needs to be conducted to investigate how firm-involved FLOSS projects differentiate from non-firm-involved FLOSS projects. One particular interesting topic might be how firm involvement in a FLOSS project changes project development over time.

In this paper, we are narrowing our field of literature and focusing our attention on studies that concentrate on forms of government and governance as it exists or as it is proposed for open source organisations. With this focus, we build a springboard for our research on forms of governance and open source communities, and aim to also reveal where the literature is sparse, thereby demonstrating where our work can further bolster this area both theoretically and in practice.

In the domain of open community governance, the work by Nyman and Lindman (2013) provides an interesting exploration of governance. They contend that forking is a central right of open source communities that serves as a mechanism to sustain those communities. Nyman and Lindman (2013, p.7) assert:

The right to fork open source code is at the core of open source licensing. All open source licenses grant the right to fork their code, that is to start a new development effort using an existing code as its base. Thus, code forking represents the single greatest tool available for guaranteeing sustainability in open source software.

In addition to bolstering program sustainability, code forking directly affects the governance of open source initiatives. Forking, and even the mere possibility of forking code, affects the governance and sustainability of open source initiatives on three distinct levels: software, community, and ecosystem.

It is interesting to note Nyman and Lindman (2013) are persuasive in suggesting that the mere possibility of exercising a right to forking works in a very powerful way to govern and sustain open source initiatives.

In the upcoming section, we discuss forms of government and their relationship to metaphors. We then describe the SPDX working group (part of the Open Compliance Program) sponsored by the Linux Foundation. Following that, we compare and contrast the metaphors currently used in the governance documents of the SPDX, assess the working group's current state within the framing afforded by the forms of government metaphors, examine SPDX in light of earlier open source community governance studies, and make recommendations for possible paths to governance structures that the SPDX working group might reasonably take.

3 Methodology

We used three approaches to fully understanding the governing structures emerging for the SPDX working group. This multi-method approach permitted us to supplement and complement what we were learning from each of the methods, resulting in a richer picture of the SPDX working group overall. The three approaches used are:

- 1 The review of relevant OSS literature regarding licensing and forms of governance.
- 2 Use of the observer-participant approach to understanding the SPDX working group by attending and participating in working group workshop sessions at LinuxCon in order to observe interactions as well as contribute to the SPDX knowledge.
- 3 Application of functional discourse analysis to examine SPDX working group governance documents.

We used a review of relevant open source software development literature to focus our attention on trends in open source software development, approaches to licensing, and forms of governance for open source software projects. Since the number of articles published on open source software is growing, and is dispersed into publications in many referent fields, a targeted literature review helped us recognise how our work fits in with what has been accomplished, showed us where there was a lack of research in working with actual for-profit corporations that were collaborating with the open source community, and made clearer where our contributions reside.

The second approach we took was to become participant-observers in the SPDX working group hosted by the Linux Foundation. Participant observation (Kluckhohn, 1940; Nandhakumar and Jones, 2002; Prasad, 1993; Schein, 1988) means that the authors systematically engaged with the SPDX working group in order to understand and interpret as well as influence what was occurring in the working group.

Thirdly, we used functional discourse analysis (de Graff, 2001; Heracleous and Barrett, 2001; Kendall et al., 2003) to analyse the SPDX working group governance documents. We chose functional discourse analysis because it served as a tool for us to examine documents as ways to communicate, but also as a way socially to construct reality and eventually to shape that reality.

4 Two early software development metaphors

Two early metaphors were used to demonstrate why open source software development was unique. The building of the cathedral was used to characterise the dominant, careful, risk mitigating commercial development of software. The transactions in the bazaar were meant to characterise the process of developing open source in the Linux community. They became so widely referenced in the open source literature, as to become shorthand synonymous for crystallising two highly different programming styles. They originated in Raymond's (2000, 2001) visionary and prescient presentation and his subsequent book updating and extending the same metaphors.

Raymond retraces his journey with developing open source software by discussing his early preconceptions about the open source development process, using the metaphor of the building of the great cathedrals of Europe as a metaphor. Anonymous, individual craftsman were largely responsible for building cathedrals, often involving generations of craftsmen working over centuries. Using that slow, meticulous process cathedrals rose as monuments to God, rather than as paeans to individual architects or designers. Individual contributions went unrecognised, artisans and craftsman were totally unknown, but the important aftermath was the totality of the cathedral as it stood and its centrality to worship, as well as to the town where it was created.

Raymond recounts that, to his surprise, the way programmers were developing the Linux operating system was far different than the cathedral metaphor he had assumed, and he goes on to frame developers' interaction by invoking a bazaar metaphor instead. Interestingly, the bazaar or marketplace metaphor is highly engaging. A bazaar, or as we might be more familiar with today, an open market: is a noisy, lively place with large numbers of diverse people, exotic and familiar goods, and haggling transactions existing side by side in what seems like endless diversity in a raucous, fluctuating, and wildly uninhibited setting. Raymond (2001, p.21) credits the architect of Linux, Linus Torvalds, with a style of development that he believes is synonymous with interactions in the bazaar:

No quiet, reverent cathedral-building here – rather the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who'd take submissions from *anyone*) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.

The cathedral and the bazaar provided an immensely rich approach to compare and contrast two disparate ways to understand what has happened in the open source community, particularly in the Linux community. The cathedral and the bazaar metaphors also formed a workable frame of understanding from which to comprehend what has been happening in both open community and commercial software development since the start of this millennium.

5 Beyond the bazaar

However, as the efforts of open source software changed to include corporate development of open source, the metaphor of the cathedral and the bazaar were no longer as compelling. The volunteers, the “happy networked hordes of programmer/anarchists outcompeting and overwhelming the hierarchical world of conventional closed software” [Raymond, (2001), p.54] were being paid to develop code. Corporations began to use, maintain, distribute, and develop open source software as part of their commercial packages via paid developers interacting with the open community.

Wayner (2011), for example, observed that the cultures were different even within the free software distribution world. Wayner (2001, p.135) notes, “GPL projects tend to be more cultish and driven by a weird mixture of personality and ain’t-it-cool hysteria, the people on the side of BSD-style license, on the other hand, seem pragmatic, organized, and focused”.

Another worthwhile attempt to explain the cathedral and bazaar metaphors was made by Fitzgerald (2006), who recognised the trend toward corporate participation in open source development and repurposed the bazaar metaphor, suggesting it was ‘better suited’ to what he termed OSS 2.0, the metamorphosis of open software development into mainstream corporate software development. He suggested that the bazaar metaphor could still be useful as a way to describe new OSS 2.0 product delivery and the product support process surrounding that. While Fitzgerald makes a contribution, the metaphor ‘OSS 2.0’ has not been widely accepted by practitioners.

Kuhn (2011), Executive Director of the Software Freedom Conservancy, and free software activist, when commenting on Nokia’s reported failure with Symbian (an open source mobile operating system), observed that too much structure (as in the cathedral-building approach) was not conducive to the development of open source software:

Upon considering Nokia’s bad trajectory, it led me to think about how Open Source companies tend to succeed. I’ve noticed something interesting, which I’ve confirmed by talking to a lot of employees of successful Open Source companies. The successful ones – those that get something useful done for software freedom while also making some cash (i.e., the true promise of Open Source) – let the developers run the software projects themselves. Such companies *don’t* relegate the developers into a small non-profit that has to lobby dozens of proprietary software companies to actually make an impact. They *don’t* throw code over the wall – rather, they fund developers who make their own decisions about what to do in the software. Ultimately, smart Open Source companies treat software freedom development like R&D should be treated: fund it and see what comes out and try to build a business model after something’s already working. Companies like Nokia, by contrast, constantly

put their carts in front of all the horses and wonder why those horses whinny loudly at them but don't write any code. (Kuhn, 2011)

Along with the involvement of for-profit corporations in open source come risks that may have been ignored in the early excitement of the bazaar style of software development. Corporations must dutifully honour software licenses, so dual licenses (used in MySQL), embedded open source licenses (used in IBM's WebSphere) and Open Core licenses (used in SugarCRM) have been introduced (Aslett, 2008). In time, risk mitigation became a concern of companies who began sharing open source software applications (Germonprez et al., 2012).

Risk mitigation in software development, which has been researched in both commercial and open source software projects for the last decade or so (Iversen et al., 2004; Keil et al., 2008; Kogut and Metiu, 2001; McGhee, 2007) can be properly addressed in the field with the help of special tools and procedures. One tool that has recently been developed is SPDX.

SPDX stands for the Software Package Data Exchange specification (Linux Foundation, 2014). It is essentially a standardised human and machine-readable 'bill of materials' format for listing and communicating licensing information and copyrights that belong to a software package. It originated with, and is still being developed by, the SPDX workgroup hosted by the Linux Foundation. The SPDX working group began as a grass roots, non-hierarchical effort by a community consisting of people from more than 20 different organisations who shared a common goal – to develop a standard format for SPDX[®]. In doing so, the workgroup seeks to help corporations share license information that enables them to achieve license and other policy compliance.

The workgroup itself is relatively new (forming in 2010) and is still in the process of organising. It is this emerging stage that is the subject of this article, because the success or failure of the group may depend not only on its culture, but also on its eventual adoption and evolution of governing structures. Our question is therefore: What governing structures are evident for the SPDX working group?

Therefore, we look for other types of governing structures to see whether there are models beyond those of the cathedral and the bazaar that foster the development of open source software collaboratively created between corporations and the open source community.

6 Governance structures

Kendall and Kendall (2010) examined societal metaphors to evaluate whether certain forms of state government encouraged or deterred the development of and sustainability of information systems. They found that different forms of government had unique characteristics and that these, in turn, influenced design. They concluded that, while there are no guarantees of outcomes, the form of government does affect the attitude, openness, and sustainability of systems.

The use of metaphors in information systems appear in early studies (Kendall and Kendall, 1993, 1994) that assesses development methods and metaphors and the success or failure of systems in which predominant organisational metaphors were observed. In the recent article, Kendall and Kendall (2010) studied the nine different organisational

metaphors identified in their earlier studies and matched them to forms of government ranging from autocracy to anarchy. In this paper, we extend the discussion to include governance on the organisational level.

Each of the forms of government unveiled different attributes that distinguish forms of government from each other. Although they explored these forms of government on a national level, we believe that some of the principles regarding forms of government can apply to organisations and communities. It is with this assumption that we examine the phenomena of corporations working in open source communities.

The first three forms of government listed by Kendall and Kendall are autocracy, stratocracy, and enlightened absolutism. These are associated with metaphors such as machine, war, and journey and are led by individuals or groups that clearly would not be appropriate for open source communities.

On the other extreme we have anarchy and panarchism, associated with metaphors like zoos and jungles. These forms of government or lack of governing structure may work for the radical free software movement (Kelty, 2008; Germonprez et al., 2014), but once again are not a good fit for the current development of open source software. In parallel the hacker culture grew. Imhorst (2005) notes that “The hacker ethic was not published as a manifesto, it was passed down orally. The ethic was never discussed.”

These structures represent chaos (anarchy) and the ability to reside in a community with the right to join or leave as one pleases, without caring about others in the community (panarchism). In open source development, these forms could have the disastrous results through the disregard of communal obligations inherent in open source cultural and belief systems.

As a middle ground, we come to what we suggest four relevant systems of governing for open source interactions among members of open communities and for-profit corporations: bureaucracy, adhococracy, meritocracy, and republic. We will describe how each of these systems of government may apply to the future development of open source software by paid developers in corporations in collaboration with the open source community.

6.1 Bureaucracy

A bureaucracy is centred on rules, regulations, policies and protocols. Bureaucracies are run by hierarchies of officials who are trying to ensure that others adhere to the rules. Individuals with creativity have a difficult time getting leverage in a bureaucracy. They need to be in a position to influence the right person or persons in the hierarchy. The focus of bureaucrats is often inward, so those individuals who want to contribute to the common good are not comfortable in the bureaucracy. Bureaucracies are about rules and structures that persist long after individuals leave.

The bureaucracy is the opposite of anarchy. Imhorst (2005) points out how the bureaucracy and anarchists are at odds:

Mainly the bureaucracy of the university made it difficult for the hackers to get valuable computing time. Open systems without bureaucracy and authorities allowed them to be more productive at the computers. Sitting behind the console of an IBM-machine they had the power over the computer. So it is maybe natural that hackers mistrusted any other power who kept them away from hacking and who wanted to make them powerless.

Is it conceivable that the metaphor of the cathedral embodies the bureaucracy? It is a society where every stage needs to be planned in advance and everything must conform to predetermined specifications. Filling out forms may seem inefficient and inflexible to some, but others find this governance structure to be effective and organised. It is unlikely that open source developers would choose to revoke the bazaar and choose the cathedral instead.

Brockmeier (2008) feels that there is no place for the bureaucracy in open source development, even if it involves large corporations. He notes in a blog piece that, “If possible, larger corporations that have open source components should do what they can to *leave them alone* and impose the very minimum amount of bureaucratic overhead on those teams. Results matter far more than process.”

6.2 *Meritocracy*

In a governing structure built on meritocracy, rewards are given to those who deliver. It is unlike other forms of governing structures where leaders are voted in or appointed. The leaders of a meritocracy have rights and responsibilities because of their intellect and their accomplishments.

While it is considered by some in a variety of professions including software development to be an ideal environment for innovation, meritocracies have been discounted as well. In *The Meritocracy Myth*, McNamee and Miller (2013, pp.215–216) explore the US dream and expose the ‘self-made man’ as an illusion and observing that other factors such as inheritance, social capital, and cultural capital are also critical in one’s success.

According to Gardler and Hanganu (2013):

The Apache Software Foundation (ASF) is perhaps the most famous example of a large-scale meritocratic community. The foundation operates with an almost completely ‘flat’ structure, which means that anyone willing to contribute can engage with their projects at any level. At the other end of the ‘control’ spectrum is the benevolent dictator governance model, which is led by one individual.

Recently, ecosystem has replaced organism as the metaphor for a meritocracy. When an ecosystem is in balance, some organisms help other organisms to survive and grow, but at other times it is survival of the fittest. Competition becomes a key dynamic in a meritocracy, and while the meritocracy is seen to be a common form in open source engagements, it may also encourage negative aspects of competition within the community.

6.3 *Adhocracy*

An adhocracy is a more flexible governing structure than the bureaucracy, and is, in effect, the opposite of the bureaucracy. Anyone within an adhocracy has the right and responsibility to make decisions and take subsequent action on their decisions. An adhocracy therefore is a social group in which people work in teams, often organised according to specialties. They share common values, demonstrate respect for one another, and help one another function within the system. According to Kendall and Kendall (2010, p.147), “the overarching principle is collectivism, where the good of the social group is more important than the individual”.

Kendall and Kendall (2010) note that the organisational metaphor that most closely resembles the adhocracy is the game. In a game, there is competition with other teams, but within the immediate group or team, individuals work together. They may bring in specialised skills and may contribute more or less than other team members from day to day. The success of the outcome depends on how the community functions as they work together towards a common goal.

Open source adhocracy was championed by Tim O'Reilly. When he introduced this notion of adhocracy, he cited Raymond's book *The Cathedral and the Bazaar* (O'Reilly, 2003). Burton (2003) notes that O'Reilly "uses the term 'adhocracy' to describe the kind of collaboration he prefers: like-minded software developers, finding each other and working in ever-shifting groups." He goes on to add, "Adhocracy is not just about free-flowing collaboration, though. According to O'Reilly, it actually shifts the balance of power from the company to the individual" (Burton, 2003).

Kendall and Kendall (2010) see the adhocracy as a competitive game in which otherwise unrelated teams of players get together to achieve some goal. The adhocracy needs a goal as well as players who are willing to work together for that goal.

6.4 Family/republic

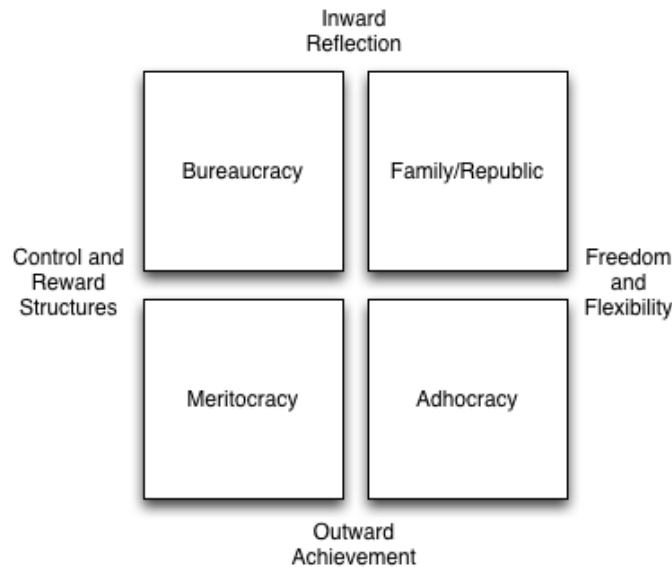
Kendall and Kendall (2010) discuss the republic as one of the governing structures, but state that it related to the family metaphor. Therefore, it is appropriate to use both terms when we discuss this governing structure. In a republic governance structure, the power resides with the people. (The word republic is derived from *res publica*, Latin for 'people thing'.) If leaders or spokespeople are needed, the people choose their representative. This governance structure also can be like a family where the father and mother are assumed to have specific roles, but are always working for the benefit of all family members.

The suggestion that a family metaphor might be an appropriate one for open source software was proposed by Zeitlyn (2003). He remarks:

I suggest that, rather than cathedrals or bazaars, we need to consider family relationships and particularly the idea of 'kinship amity' to understand the open source movement. Within the family there are no calculated economic relationships: parents do not bill their children for their upkeep and so on. It is a type of gift relationship but one with a different type of symbolic capital accruing to the givers, depending on the variety of kinship system a family belongs to Parents give their children life and for that can never be repaid. Children are eternally indebted to their parents hence the enduring symbolic power that parents have over their children. Parents benefit from the work of their children but also work to support them when small. No accounts are kept. This is as close to a human universal as it is possible to get. (p.1290)

The family/republic structure encourages communication, work groups with common goals, and values that strengthen the community.

Figure 1 puts the four governance forms into a relational form. Bureaucracies and meritocracies are about controls and rewards. Measurement is important to both. On the other hand, the family/republic and adhocracy stand for freedom and flexibility. Looking at the forms of government in another way, the bureaucracy and the family/republic tend to spend time on inward reflection, while the meritocracy and adhocracy pride themselves on outward achievement.

Figure 1 Four governance structures and their manifestations

7 Analysis – the governance of the SPDX working group

Literature has been presented that indicates that open source communities are moving in the direction of a more structured type of governance. We have observed that this structure can take a number of different forms. Communities may organise as a bureaucracy, an adhocracy, a meritocracy, or as a family/republic.

Examining the published governance structure of SPDX working group, we find no evidence or hint of a bureaucracy. Simply put, the SPDX working group is not engaged in cathedral building. However, there is evidence that SPDX chooses to reflect upon itself as a meritocracy, an adhocracy, and a republic. We will examine each below in more detail.

The SPDX see itself as a meritocracy, as its governance manual makes this absolutely clear:

SPDX operates much like a meritocratic, consensus-based community project. Anyone with an interest in the project can join the community, contribute to the specification, and participate in the decision making process. This document describes how that participation takes place and how to set about earning merit within the project community. (Section 0 of SPDX[®] Governance Document, Linux Foundation, 2012)

The SPDX community sees itself as an adhocracy, as demonstrated with common member goals.

The SPDX community realises that its success depends on a common goal and using words such as ‘evangelising’ and ‘providing moral support’. The community accepts values associated with collectivism and realises that the good of the community is more important than the good of the individual as stated:

The project asks its users to participate in the project and community as much as possible. User contributions enable the project team to ensure that they are satisfying the needs of those users. Common user contributions include:

- using the SPDX specification to describe the content and licensing of their software
- submit bug reports/code contributions to the tools following the development process set in place
- evangelising about SPDX (e.g., giving a talk at a conference, a link on a website and word-of-mouth awareness raising)
- informing contributors of strengths and weaknesses from a new user perspective
- join SPDX track at Linux Foundation conferences
- providing moral support (a ‘thank you’ goes a long way)
- potentially providing financial support (currently SPDX operates with no budget, but it may eventually need one) (Section 1.2 of SPDX[®] Governance Document, Linux Foundation, 2012).

Additionally, an adhocracy is apparent in the voting process. Although voting can occur, most decisions appear to be made through a specific form of consensus described below:

Any team member can make a proposal for consideration by the community. In order to initiate a discussion about a new idea, they should send an email to a team mailing list or enter the idea to the SPDX issue tracker. This will prompt a review and, if necessary, a discussion of the idea. The goal of this review and discussion is to gain approval.

In general, as long as nobody explicitly opposes a proposal, it is recognized as having the support of the community. This is called lazy consensus – that is, those who have not stated their opinion explicitly have implicitly agreed to the implementation of the proposal.

Lazy consensus is a very important concept within the project. It is this process that allows a large group of people to efficiently reach consensus, as someone in support of a proposal need not spend time stating their position, and others need not spend time reading such mails. (Section 4.1 of SPDX[®] Governance Document, Linux Foundation, 2012)

The SPDX community sees itself as a family/republic governing structure, in how Team Lead and others they are appointed within the working group:

These individuals can be nominated by Team Members or members of the Core Team. After discussion with the nominees, Team Leads are appointed by the Core Team taking into account such things as the nominees’ willingness to take on the role, skills, and level of participation as well as the need to maintain a balanced perspective on the Core Team (e.g., there should not be more than one Team Lead from the same organization or company). Once someone has been appointed Team Lead, they remain in that role until they choose to retire or the Core Team casts a two-thirds majority vote to remove them. Team Leads have no additional authority over other members of the team. (Section 1.3 of SPDX[®] Governance Document, Linux Foundation, 2012)

Furthermore, the chairman has limited authority as is typical for a republic.

The Chairman leads Core Team meetings and also coordinates and leads General Meetings. Once someone has been appointed Chairman, they remain in that role until they choose to retire, or the Core Team casts a two-thirds majority vote to remove them.

The Chairman has no additional authority over other members of the Core Team: the role is one of coordinator and facilitator. The Chairman is expected to ensure that all governance processes are adhered to and has the casting vote when the project fails to reach consensus. (Section 1.5 of SPDX[®] Governance Document, Linux Foundation, 2012)

It interesting to note that the leadership roles defined in the family/republic metaphor of the SPDX working group governance documents include both those of coordinator and facilitator. Other types of more authoritarian leadership roles are markedly absent in this governance structure.

8 Discussion

As the governance structure of the SPDX working group changes, we observe that the structure, communication, and feelings that transpire are welcomed into the group as if they were source code itself. We observed an atmosphere of transparency and cooperation.

The governance structure shows aspects of three of the four different forms of government: that of meritocracy, a republic, and an adhocracy as shown in Table 1. Bureaucracy was not found in the working groups writings. We found that although a detailed governing structure was described on the SPDX website, it was an idealised one, and one that intermingled the three of the four structures mentioned here.

Table 1 Four governance structures

<i>Structure</i>	<i>Emphasises</i>	<i>Central theme</i>	<i>Successful if it produces</i>
Bureaucracy	Policies and procedures	Formality	Top-down designed, fully documented software
Meritocracy	Results	Reputation	Enhancements to existing software packages
Adhocracy	Creativity and risk taking	Freedom	Innovative open source software
Family/republic	Collectivism and relationships	Loyalty	Reciprocal respect and admiration

Note: Evidence of meritocracy, adhocracy, and family/republic were found in the governance documentation on the SDDX Working Group.

In a bureaucracy emphasis is on following formal policies and procedures. Success depends on top-down design using the systems development life cycle approach to produce reliable software packages and documentation. It is unlikely that open source software will be developed within the bureaucracy. No evidence of a bureaucracy was found in the governance structure of the SPDX working group.

The meritocracy is emblematic of a no-non-sense, results-oriented group. Those who thrive in this group are competitive and goal-oriented. The central theme is producing.

Success, therefore, is in developing software packages that enhance software products. Since reputation is a big factor, concern about license compliance becomes an issue.

Hann et al. (2004) explore meritocracies in open source software communities of apache developers and empirically demonstrate that “employers pay significantly higher wages to participants who attain a higher rank in the OSS community.” While this is an important finding for those individuals in the community, we need to understand more about the meaning of merit itself.

Therefore, we fully recognise how broadly construed a meritocracy can be. It needs to include not just intellect and effort, but also values and sustainability. O’Mahony and Ferraro (2007) note:

Contradicting a simplistic meritocratic explanation, developers who engaged in organization-building behaviors were more likely to become members of the leadership team. Thus, Debian may be a meritocracy, but merit is not measured solely by technical contribution. A prevalent assumption in research on open source software communities is that the contributions community members value the most are purely technical. Our study shows otherwise. In this community, the informal work of coordinating individual efforts and linking them to community goals became vital to leadership, particularly as the project matured. (p. 22)

The word ‘meritocratic’ appears in the SPDX governance document (Linux Foundation, 2012), but this is the only place the concept of meritocracy occurs. The main problem of the SPDX working group adopting the meritocracy governing structure and accompanying metaphor, is the nature of the group itself. As Germonprez et al. (2012) note, “The goal of SPDX is to mitigate risk inherent between corporations in a supply chain of open source code.” In order for the meritocracy to work, more positive outcomes need to be stressed.

Next is the consideration of adopting adhocracy, which is both flexible and is working towards a common good. The adhocracy structure encourages creativity as well as risk taking. The phrase *governance structure* is used loosely as members cooperate and share because they like to innovate and even experiment. Freedom is a central theme, not an afterthought.

There was much evidence demonstrating that the SPDX working group embraced an adhocracy. It was apparent in their call to participate in the workgroup, including phrases like “informing contributors of strengths and weaknesses from a new user perspective” in their governance document, and their lazy consensus form of voting. These are all expressed in a very positive way in their governance document.

Finally, the emphasis in the family/republic form of governance is on relationships and this encourages a friendly atmosphere. Loyalty to the group is one of the central themes. Because of the family nature of this form, commitment should be very high. The leaders are considered to be nurturing figures or mentors. The group will be successful if they continue to show love and respect for everyone else in the group. Consensus is crucial in decision-making.

Then we can look at family/republic as described by Zeitlyn (2003) when discussing the kinship amity metaphor in the development of open source software:

Relations of amity are real and have been achieved with those you have long collaborated with. There are others that you may know about vaguely who are in similar relationships but not with you. Between you, *ab initio*, there is no relationship – you have not worked together. You are not kin. But kinship

amity can be created through interaction. And the crucial type of interaction is gift exchange – the gift of code! (p.1290)

The family/republic also appears in the SPDX working group governance document. The authority figures are nominated, then appointed and “they remain in that role until they choose to retire” much like a father or mother of a family. The statement goes on to say “or the Core Team casts a two-thirds majority vote to remove them,” but it is clear that there is some intended permanency to the leadership roles. Additionally, the chairman who leads the core team has ‘no additional authority’ affirming a republic governing structure. Families have various goals but a common bond. These statements demonstrate positive ideals that the SPSX working group has included in its governance structure.

Having observed mixed metaphors in the governance structures of the SPDX working group we ponder whether or not they will be successful. Will the confusion surrounding governance endanger and eventually hamper productivity towards their goals? Should there be concerns about the SPDX working group’s difficulty in ‘evangelising’ new members if types of governance are mixed or too complicated to convey? How can they describe the culture and governance structure of the SPDX working group to potential contributors or users? Should the working group embrace a single metaphor to replace the bazaar? These are valid questions that need to be monitored.

There are approaches that we as researchers and participant observers can recommend. One is to reflect back to the working group that at present the governance structure suggested in the governance documents is mixed. While the mixed metaphors and forms may make it likely that more people will be initially attracted to the working group, it makes future strategic moves ambiguous and unclear. Just as the bazaar represented the world of developers interacting voluntarily to programme free software, the SPDX working group may help themselves by identifying a new, encompassing metaphor that effectively captures the goals and values of the members.

9 Conclusions

Our initial contribution to the literature involved creating increased awareness that open source development is changing. We can no longer accept the dichotomy of the cathedral and the bazaar. While the cathedral may be a suitable metaphor for the traditional systems development life cycle of proprietary software, the bazaar now only describes the free software movement as a historical consideration. We note that we need new metaphors for open source software development as it exists today.

Our second contribution to the literature was introducing new metaphors for thinking about governance for collaborative open source projects undertaken jointly by for profit corporations and the open source community. Open source software development takes on new metaphors of the society, family, competitive game, and ecosystem. As we capture the essence of these metaphors and how they relate to the development of open source software today, we identify four representative expressions: control, collectivism, creativity, and competition.

In a third contribution to the literature, we supported and expanded the connections from earlier works concerning how metaphors influence forms of governance and how forms of governance influence the types of information systems being created. We also extended the discussion of governance on a state level to the organisational level. These

metaphors introduce us to these possible forms of open source governance: the bureaucracy, meritocracy, adhocracy, and family/republic. In this paper, we explore each of these in more detail, elaborating on the nuances of each governance structure in light of the SPDX[®] working group. We have observed that three of the governing structures identified by Kendall and Kendall (2010) appear in the governing structure for the SPDX working group as documented on their website (Linux Foundation, 2012). They are the meritocracy, the adhocracy, and the family/republic. Bureaucracy, the cathedral building described by Raymond (2000, 2001) was non-existent within the SPDX working group.

A fourth contribution to literature resulting from our work is that we provided insights and reflections on a group in their formative stages. We, as researchers and community members, participated in the emergence of a joint industry-open source group, and thus we were able to shape its future in some way. The significance of this opportunity stands in contrast to more typical studies represented in the IS literature where IS researchers study the recognised impact of groups *after* they have existed for a lengthy period of time (an historical perspective) or *after* they have ceased to exist (a *post mortem* or archeological approach), examining the causes of the organisation's demise.

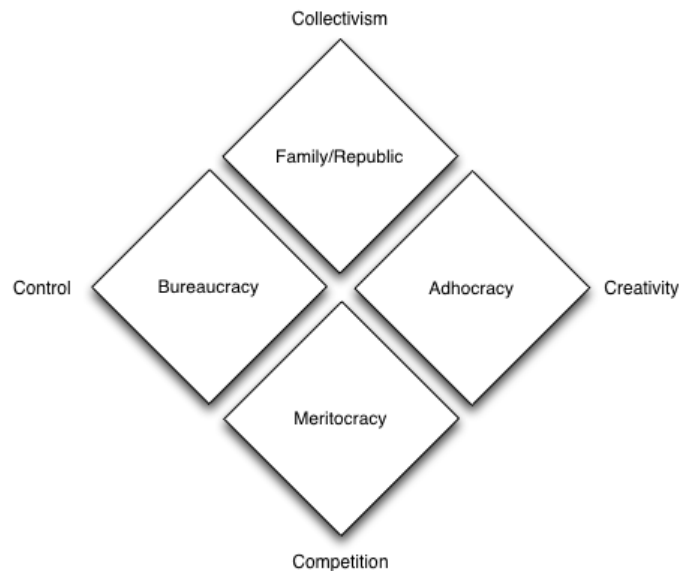
Our first contribution to practice through our research included heightening awareness of availability of the SPDX specification, which: "Enables suppliers and consumers of software that contains open source code to provide a 'bill of materials' that describes the open source licenses and components that are included. The specification defines a common file format to communicate this information," (Linux Foundation, 2012). Any corporation, developer, or manager involved in the open source software supply chain will find that this specification is relevant.

A second contribution to practice we made was to reflect back to the SPDX working group the need to pay greater attention to their governance because the evidence shows that they currently combine meritocracy, adhocracy, and family/republic in their governance documents. Our intent is to help the working group make strategic decisions about their governance in the future as the importance of open source licensing and compliance continues to grow for developers and corporations alike.

In Figure 2, we call attention to the cultural compass points (the 4Cs) of control, collaboration, collectivism, and creativity and their relationship to each of the governance structures. While the bureaucracy is all about control, the family/republic embraces the notion of collectivism, a space where collaboration is highly valued. The adhocracy encourages creativity. Innovation has a better chance of occurring when the group adopts an adhocracy. Finally, the meritocracy embraces competition by rewarding those who show results. The SPDX working group will eventually need to focus on one of these 4Cs.

It is an exciting and challenging time for open source generally and the SPDX working group more specifically. The need for a group dedicated to open compliance was expressed in 2007, and the SPDX working group was officially formed in 2010 by both developers and users in the open source community and for-profit corporations. Its success, which will in large part depend on how the working group is governed, is important to the Linux open source community as well corporations. The Linux Foundation (2012) has announced SPDX as "one of the pillars of its new Open Compliance Program". Their mission, of creating an open source specification to enable all types of users and developers to effectively 'see' what open source licenses are used, with minimal cost, across the supply chain, is an urgent one.

Figure 2 Governance structures and the cultural compass points of collectivism, creativity, competition, and control



Time will tell whether the SPDX working group will emerge more distinctly as a meritocracy, an adhocracy, or a family/republic. Its success, however, will not depend simply on results, creativity, or relationships. It will also depend upon positive strengths and values.

Acknowledgements

This project was funded through the National Science Foundation VOSS-IOS Grant: 1122642, Organizational Participation in Open Communities.

References

- Aksulu, A. and Wade, M. (2010) 'A comprehensive review and synthesis of open source research', *Journal of the Association for Information Systems*, Vol. 11, No. 11, pp.576–656.
- Aslett, M. (2008) *Andrew Lampitt Defines Open-Core Licensing* [online] <http://blogs.the451group.com/opensource/2008/09/01/andrew-lampitt-defines-open-core-licensing/> (accessed 10 January 2014).
- Brockmeier, J. (2008) *Bureaucracy and Open Source: Do they Mix?* [online] <http://www.zdnet.com/blog/community/bureaucracy-and-open-source-do-they-mix/116> (accessed 10 January 2014).
- Burton, J. (2003) *Open Source Adhocracy* [online] <http://www.spiked-online.com/newsite/article/5075#.UsYcJfZiqX0> (accessed 10 January 2014).
- Crowston, K., Wei, K., Howison, J. and Wiggins, A. (2012) 'Free/Libre open-source software development: what we know and what we do not know', *ACM Computing Surveys*, Vol. 44, No. 2, Paper 7.

- Daniel, S., Maruping, L., Cataldo, M. and Herbsleb, J. (2011) 'When cultures clash: participation in open source communities and its implications for organizational commitment', *International Conference on Information Systems*, Shanghai, China.
- de Graff, G. (2001) 'Discourse theory and business ethics: the case of bankers' conceptualizations of customers', *Journal of Business Ethics*, Vol. 31, No. 4, pp.299–319.
- Dueñas, J.C., Parada, H., Cuadrado, F., Santillán, M. and Ruiz, J. (2007) 'Apache and Eclipse: comparing open source project incubators', *IEEE Software Magazine*, Vol. 24, No. 6, pp.90–98.
- Fielding, R. (1999) 'Shared leadership in the Apache Project', *Communications of the ACM*, Vol. 42, No. 4, pp.42–43.
- Fitzgerald, B. (2006) 'The transformation of open source software', *MIS Quarterly*, Vol. 30, No. 3, pp.587–598.
- Gardler, R. and Hanganu, G. (2013) *Meritocratic Governance Model* [online] <http://oss-watch.ac.uk/resources/meritocraticgovernancemodel> (accessed 10 January 2014).
- Germonprez, M., Kendall, J.E., Kendall, K.E. and Young, B. (2014) 'Born free: how the origins and advancement of open source software through sharing will uphold the values of free software', *Decision Line*, Vol. 45, No. 1.
- Germonprez, M., Kendall, J.E., Kendall, K.E., Mathiassen, L., Warner, B. and Young, B. (2013) *The Shared Nature of Design: A Field Study of Corporate Participation with Open Source Communities*, Working Paper, University of Nebraska-Omaha.
- Germonprez, M., Young, B., Mathiassen, L., Kendall, J.E., Kendall, K.E., Warner, B. and Cao, L. (2012) 'Risk mitigation in corporate participation with open source communities: protection and compliance in an open source supply chain', *International Research Workshop on IT Project Management 2012*, Paper 3 [online] <http://aisel.aisnet.org/irwitpm2012/3> (accessed 10 January 2014).
- Gurbani, V., Garvert, A. and Herbsleb, J. (2010) 'Managing a corporate open source software asset', *Communications of the ACM*, Vol. 53, No. 2, pp.155–159.
- Hann, I., Roberts, J. and Slaughter, S. (2004) 'Why developers participate in open source software projects: an empirical investigation', *ICIS 2004 Proceedings*, Paper 66 [online] <http://aisel.aisnet.org/icis2004/66> (accessed November 2013).
- Heracleous, L. and Barrett, M. (2001) 'Organizational change as discourse: communicative actions and deep structures in the context of information technology implementation', *Academy of Management Journal*, Vol. 44, No. 4, pp.755–778.
- Imhorst, C. (2005) 'Anarchy and source code – what does the free software movement have to do with anarchism?' [online] <http://www.infoshop.org/pdfs/imhorst.pdf> (accessed 10 January 2014).
- Iversen, J.H., Mathiassen, L. and Nielsen, P.A. (2004) 'Managing risks in software process improvement: an action research approach', *MIS Quarterly*, Vol. 28, No. 3, pp.395–433.
- Keil, M., Li, L., Mathiassen, L. and Zheng, G. (2008) 'The influence of checklists and roles on software practitioner risk perception and decision-making', *The Journal of Systems and Software*, Vol. 81, No. 6, pp.908–919.
- Kelty, C. (2008) *Two Bits: The Cultural Significance of Free Software*, Duke University Press, Durham, NC.
- Kendall, J.E. and Kendall, K.E. (1993) 'Metaphors and methodologies: living beyond the systems machine', *MIS Quarterly*, Vol. 17, No. 2, pp.149–171.
- Kendall, J.E. and Kendall, K.E. (1994) 'Metaphors and their meaning for information systems development', *European Journal of Information Systems*, Vol. 3, No. 1, pp.37–47.
- Kendall, K., Kendall, J. and Kah, M. (2003) 'The emergence of information communication technologies for developing countries: using dramatisitic analysis to identify heroes and villains', in Montealegre, R., Korpela, M. and Poulymenakou, A. (Eds.): *IS Perspectives and Challenges in the Context of Globalization*, pp.403–418, Kluwer Academic Publishers, Boston.

- Kendall, K.E. and Kendall, J.E. (2010) 'Forms of government and IS sustainability: a positive design approach to the design of information systems', in Thatchenkery, T., Cooperrider, D. and Avital, M. (Eds.): *Positive Design and Appreciative Construction: From Sustainable Development to Sustainable Value, Advances in Appreciative Inquiry*, Vol. 3, pp.137–155, Emerald, Bingley, UK.
- Kidman, A. (2010) *75% of Linux Code Now Written by Paid Developers* [online] <http://apcmag.com/linux-now-75-corporate.htm> (accessed 10 January 2014).
- Kluckhohn, E. (1940) 'The participant observer technique in small communities', *American Journal of Sociology*, Vol. 46, No. 3, pp.331–343.
- Kogut, B. and Metiu, A. (2001) 'Open-source software development and distributed innovation', *Oxford Review of Economic Policy*, Vol. 17, No. 2, pp.248–264.
- Kuhn, B. (2011) *Will Nokia Ever Realize Open Source is not a Panacea?* [online] <http://ebb.org/bkuhn/blog/2011/08/18/open-source-not-panacea.html> (accessed 10 January 2014).
- Lerner, J. and Tirole, J. (2005) 'The economics of technology sharing: open source and beyond', *The Journal of Economic Perspectives*, Vol. 19, No. 2, pp.99–120.
- Linux Foundation (2012) *SPDX[®] Governance Document* [online] <http://spdx.org/about-spdx/governance> (accessed 10 January 2014).
- Linux Foundation (2014) *About SPDX[®]* [online] <http://spdx.org/about-spdx> (accessed 10 January 2014).
- Markus, M.L. (2007) 'The governance of free/open source software projects: monolithic, multidimensional, or configurational?', *Journal of Management & Governance*, Vol. 11, No. 2, pp.151–163.
- McGhee, D.D. (2007) 'Free and open source software licenses: benefits, risks, and steps toward ensuring compliance', *Intellectual Property & Technology Law Journal*, Vol. 19, No. 11, pp.5–9.
- McNamee, S.J. and Miller Jr., R.K. (2013) *The Meritocracy Myth*, 3rd ed., Rowman & Littlefield Publishers, Lanham, MD.
- Mockus, A., Fielding, R. and Herbsleb, J. (2000) 'A case study of open source software development: the Apache Server', in *Proceedings, International Conference on Software Engineering (ICSE)*, Limerick, Ireland, 5–7 June.
- Mockus, A., Fielding, R. and Herbsleb, J. (2002) 'Two case studies of open source software development: Apache and Mozilla', *ACM Transactions on Software Engineering and Methodology*, Vol. 11, No. 3, pp.309–346.
- Nandhakumar, J. and Jones, M. (2002) 'Development gain? Participant observation in interpretive management information systems research', *Qualitative Research*, Vol. 2, No. 3, pp.323–341.
- Nyman, L. and Lindman, J. (2013) 'Code forming, governance, and sustainability in open source software', *Technology Innovation Management Review*, January, pp.7–12 [online] <http://timreview.ca/article/644> (accessed 3 January 2014).
- O'Mahony, S. and Ferraro, F. (2007) 'The emergence of governance in an open source community', *Academy of Management Journal*, Vol. 50, No. 5, pp.1079–1107.
- O'Reilly, T. (2003) *The Open Source Paradigm Shift* [online] <http://oreilly.com/tim/archives/ParadigmShift.pdf> (accessed 10 January 2014).
- Open Source Initiative (n.d.) [online] <http://opensource.org/docs/osd> (accessed 10 January 2014).
- Prasad, P. (1993) 'Symbolic processes in the implementation of technological change: a symbolic interactionist study of work computerization', *Academy of Management Journal*, Vol. 36, No. 6, pp.1400–1429.
- Qu, W., Yang, Z. and Wang, Z. (2011) 'Multi-level framework of open source software adoption', *Journal of Business Research*, Vol. 64, No. 9, pp.997–1003.

- Raymond, E. (2000) 'The cathedral and the bazaar', Original presentation to Linux Kongress, 21 May 1997 and subsequent updates [online] <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s12.html> (accessed 10 January 2014).
- Raymond, E. (2001) *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Kindle Edition Published in 2008, O'Reilly Media, Sebastopol, CA.
- Roberts, J., Hann, I. and Slaughter, S. (2006) 'Understanding the motivations, participation, and performance of open source software developers: a longitudinal study of the Apache Projects', *Management Science*, Vol. 52, No. 7, pp.984–999.
- Schein, E. (1988) *Process Consultation: Its Role in Organization Development*, Vol. 1, Prentice Hall, Englewood Cliffs, NJ.
- Wayner, P. (2001) *Free for All. How Linux and the Free Software Movement Undercut the High-Tech Titans*, Harper Business, New York.
- Zeitlyn, D. (2003) 'Gift economies in the development of open source software: anthropological reflections', *Research Policy*, Vol. 32, No. 7, pp.1287–1291.