

Mutability and Becoming: Materializing of Public Sector Adoption of Open Source Software

Maha Shaikh

► **To cite this version:**

Maha Shaikh. Mutability and Becoming: Materializing of Public Sector Adoption of Open Source Software. Working COnference on Shaping the Future of ICT Research, Dec 2012, Tampa, FL, United States. pp.123-140, 10.1007/978-3-642-35142-6_9 . hal-01515850

HAL Id: hal-01515850

<https://hal.inria.fr/hal-01515850>

Submitted on 28 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Mutability and *Becoming*: Materializing of Public Sector Adoption of Open Source Software

Maha Shaikh

Warwick Business School, University of Warwick, Coventry CV4 7AL, UK
Maha.shaikh@wbs.ac.uk

Abstract. Juxtaposing two local council cases of open source software adoption in the UK we highlight their differences and similarities in open source adoption and implementation. Our narratives indicate that for both cases there was strong goodwill towards open source yet the trajectories of implementation differed widely. We draw on Deleuze and Guattari's ideas of becoming, tracing versus mapping and multiplicity to explain how *becoming* occurs at different speeds. Our data shows that the *becoming* of adoption can be both constrained and precipitated by various forms of materiality (of the assemblage of the open source ecosystem). The interesting point of departure of our study is how open source software – a much touted transparent and open phenomenon – is by its nuanced and layered mutability able to make the process and practices surrounding it *less visible*.

Keywords: open source, public sector procurement, becoming, mutability, materiality

1 Introduction

Why is it that when two different local councils adopt open source software that one proves to become more adept at it, while the other finds itself implicated in different machinations? This paper approaches this question with a focus on the *becoming* (performative understanding [1, 2]) of the primary adoption process. Our contribution lies in unpacking the adoption and procurement of open source software (OSS) by two different local councils in the UK sensitized by ideas of becoming, mutability and materiality. We recognize and show how the becoming (complicated, uncertain, never stable or complete) of OSS adoption indicates that the process of *becoming* occurs at different speeds [3]. The speed of becoming is managed and controlled and can be purposively directed. Our cases show how management in the local councils reined in (or otherwise) the process of becoming via material instantiations of OS. The nature of materiality was manipulated in both cases to different ends, and results.

We draw on Deleuze and Guattari's [4] ideas of becoming, tracing versus mapping and multiplicity alongside the shared ontology of Actor Network Theory (with [4] – ie

a relational ontology where information technology and users are not defined outside their relationship but *in* their relational networks [5]. This consideration moves the focus of the analysis from the actor, either human or non-human, towards a more complex and less defined phenomenon, which is the interaction [6-8]. It has a “relational materiality” [9]. This reflects an aversion to accept *a priori* the pre-existence of social structures and differences as somehow intrinsically given in the order of things, or what Barad terms “agential realism” [10, p810]. This ontological predisposition sensitizes us to the idea that more than one reality is possible. Indeed successful software adoption is never a certainty but drawing on ideas of becoming takes our analysis further by laying bare both successful and unsuccessful possibilities that are attempted, but perhaps never quite become. The relevance of such an approach lies in its ability to unpack various criteria, actors, relations and material considerations that a simple adoption study would do little justice to as ‘performativity leaves open the possibility of events that might refute, or even happen independently of, what humans believe or think’ [11, p323].

These ideas and our theoretical underpinnings are explained further in section 3. Section 2 provides background literature to this study and contextualizes our work, and section 4 and 5 explain the methodology and two cases respectively. Section 6 is the analysis and discussion and the paper concludes with section 7.

2 Open Source Adoption and Procurement in the Public Sector

Adoption of software and IT often does not follow a well-laid out plan and every context is different. Context, as argued by Robey and Sahay [12] plays a very important role as technologies are adopted and used in situ and need to be studied as such. A certain amount of drift is usual [13] and perhaps even necessary as this *performs* acceptance at the individual and collective level. Other IT adoption studies usually focus on workarounds as a manner of performative adoption [14-17]. However, such work has always kept users as central. This paper is not an adoption study in the conventional sense of user adoption. The aim of this paper is to understand higher level procurement decisions where decisions are made by strategists, top IT managers and policy writers so users, unlike for Boudreau and Robey [18], are not the real focus for us. Instead, we look to literature on procurement of open source and primary adoption of open source software (by IT staff and developers, and not secondary adoption by users [19]).

Open source software implies openness of the source code thus making it possible to change, and improve the code. In effect open source encompasses certain freedoms¹ that are embedded in the license of the code². Procurement [20] and acquisition decisions by many governments are currently under question, and greater scrutiny has led to governments in the European Union, UK, Australia [21] and the USA [22] to amend their habits. Research to date in the area of open source use and

¹ <http://www.opensource.org/docs/osd>

² <http://www.opensource.org/licenses/index.html>

adoption in the public sector, though growing, is still quite patchy. A UK based study [23] focused on eight different local councils and agencies. This work outlined a number of concerns and key areas that need improving in the public sector before successful adoption can emerge. This was however, a high level study where details of each case the various struggles were not the focus. Likewise in the US, studies have shown open source use adoption needs top level support and encouragement for success [24]. Brazil is a very interesting case where the success of open source adoption has been explained and emphasized as a product of insurgent experts [25].

The European Commission has an explicit directive to promote software alternatives [26], especially open source software. And very recently, the British Government's Cabinet Office [27-29]³ met with the large and influential system integrators to declare a greater need to have open source choice offered to the government. The argument put forward by the Cabinet Office was that the government was unable to choose open source as an alternative if this was not offered as an option by the integrators. Open source software, along with open data and open standards is fast becoming part of the language that governments all over the world are eager to adopt [30]. It is, for example, one of the basic building blocks of the US government in relation to its encouragement of its open government initiative [31].

Open source software is part of the easing of recession and costs of IT in the public sector in the UK. However, as the UK government is aware, open source software is still a rather unknown phenomenon. The true and complete costs involved with switching to another software, be it open source or not, are not easy to evaluate [32]. Open source software further complicates matters with close to zero license costs, but this does not necessarily translate to lower costs in other aspects [33].

Private companies [34] tend to adopt open source software for a mix of reasons which clearly include the promise of reduced costs of adoption, but there is often a strategic aspect, as well as a strong desire to innovate [35, 36]. The public sector would like to enjoy these benefits as well but till very recently the desire to innovate was not foremost for most governmental agencies. Public sector organizations are not profit orientated yet there is much to learn from private companies and their manner of dealing with open source. The larger idea here is the level of experience and comfort that private companies bring to open source adoption which is sorely lacking in the public sector. There are some exemplary cases of open source adoption by the public sector like the Extremadura case in Spain [37, 38] but there are far more 'success' stories of open source adoption by commercial companies [39-42]. What we found missing in the literature was attention to the role played by politics within and external to public sector organizations attempting to adopt OSS.

Our research was thus motivated by a desire to make sense of OS adoption while deliberating on politics and other heretofore ignored actors. More specifically we were driven by a need to understand how open source software adoption was being managed by public sector organizations and why, when the circumstances and reasons for adopting open source for both councils were so similar the results were so very

³<http://webarchive.nationalarchives.gov.uk/+http://www.cabinetoffice.gov.uk/resource-library/procurement-policy-note-ppn-use-open-standards-when-specifying-ict-requirements>

different. Thus, our main research question is: *How can, and is the becoming of open source software adoption managed and controlled?*

3 Ontological Positioning

Becoming is not a specific state but rather a focus on movement from the then to now, not a move from one state to another. It reflects a passing of time and a process, 'becoming thus sees the idea of an organization's existence not as an ontologically stable, but rather as something that exists only in its duration' [43]. It is in this becoming that organizing materializes with a focus 'on movement rather than that which is moved' [43, p159]. The previous tradition of studies of change have been criticized for focusing on stability in order to understand change [15, 16, 44]. This suggests the need to reverse 'ontological priorities' [2] and for keener perceptions of the ongoing nature of change or 'changing' [45]. This reversal is helpful not least if it allows a better understanding of the micro-processes of change, treats change as dynamic and unfolding rather than as a fait accompli, and makes it ontologically possible to 'see' change by directly looking for changing, rather than as a byproduct of some comparative stabilities [2]. Thus in developing the concept of becoming Clegg et al. [43] emphasize the focus on movement, not on what has moved or where it arrives (at best mere snap shots, moments in time); becoming is about travel and mutation rather than what has mutated. Stability is then at best fleeting but more likely to be illusionary; change is reality.

Drawing on various interpretations of organizational change [15, 16, 45] Tsoukas and Chia [2] argue that improvisation within a context is somewhat narrow in recognizing the prevalence of change, changing routines focus solely on human agents and agency, and any and all collectives and organizations never quite become, and indeed are in a constant form of changing (becoming). And becoming is performative [2, 15] where performative [46, 47] implies that something becomes into existence and has effect and materiality through action and performance – action through words, movement or some more abstract change is needed.

Becoming implies an ontological understanding where the world (reality) has a middle but there is no beginning or end. Reality is not seen as hierarchical but rather as a rhizome of multiplicities that can and do fracture, rupture, and entangle. Such an ontological position creates an imposition on the researcher to make a methodological cut into reality where Law [9] has argued that attention to differences and interactions can become that cut. Drawing on this idea we 'map' rather than trace the unfolding performance of OSS in two local councils. The idea of a map builds on the importance of change and seeing the world as constantly new in its emergence. Tracing implies almost a replication of the old that is stifling (and unrealistic?). A map is thus performative, a becoming – we do not know and will find it difficult to predict assemblages or rhizomes (various possible futures). A tracing, on the other hand, is a static understanding of the world.

The desire to understand the status quo does not mean we must have a static view of the world [48]. The trapeze artist walks across the tightrope appearing to be not

doing very much and keeping straight, however this keeping upright requires many muscles and nerves are in a constant flux else the artist would fall. In an organizational setting this form of changing occurs all the time but is difficult to perceive. Thus an ontological understanding that becoming brings with it nudges the researcher to focus on ruptures as an epistemological tool to cut into the data and allow the changing to emerge for us.

4 Methodology

We carried out in-depth, semi-structured interviews with core personnel in each organization. We chose two local councils within the UK, Camden and Bristol City. Our choice was dictated by their deep and long interest in open source software where they had experienced rather different results to date.

4.1 Data Collection

We conducted 32 in-depth interviews over the course of late 2010 to early 2012. The personnel interviewed ranged from the open source policy writer, IT and developer team, floor-walking members, users, and strategy level staff, but also those involved in making procurement decisions and strategy of open source use in the organization. Each of our interviews lasted for an hour or more. Our short interview guide covered questions relating to basic information about the length of open source adoption, and the role of the interviewee in the process to more detailed examination of obstacles, opportunities, and challenges involved. The main ideas the respondents focused on included the lack of maturity level of open source software, there is no policy in most councils for open source adoption, license confusions and lack of knowledge about the implications of various open source licenses (see Table 4).

4.2 Data Analysis

The material from the interviews was analyzed [49] systematically (using Atlas.ti software) for the main lessons, decisions, challenges, strengths, advice, best practices, consequences and other interesting elements that emerged from the interviews. Our code book consisted of phrases consistent with ideas of becoming, change, rhizomes and mutability, but we also allowed the data to give rise to new codes.

We focused on open and axial coding of the data. For nearly half of the interviews we open coded very finely by studying phrases. As we progressed through the coding process we noticed that few new codes or ideas were emerging so we began to code *paragraph by paragraph*. Our coding process thus remained rigorous and we stayed faithful to our data and the ideas emerging from it. Table 1 shows some sample data and the manner of coding we employed.

Table 1. Sample of Data and Coding.

Open codes	Sections of Data	Axial codes
Code TCO Requirements	There is a procurement template we have had to develop... it will have the magic box of hidden costs. If we invite a number of suppliers, they are all asked the same questions.....there are some essential criteria that must be met.	Embedding materiality
Ownership Responsibility Sustaining community	We need laboratories set up to force them to work through such ideas and we are struggling for techniques of how to do that..... need to make it part of people's job description and they are monitored on it ...otherwise I am left buffering between developers and businesses which is a compromise.	Transparency via materiality
Conversion issues License differences User resistance	But there were users that were doing somewhat complex things with spreadsheets ... their material would not convert so easily and such users were more difficult to bring around to open source use.	Recalcitrant materiality
Certification Documentation Reliability Persuasive standards Interoperability	An issue did arise. We are part of a government connect called Extranet. It is a part of a government secure intranet. There is a code of connection to connect up to this service which we are required to do. That code of connection is built upon the advice of the CESG. CESG certified Blackberry as the only IL3 secured email product. Blackberry only support Exchange, Notes...then how could we choose an open source email platform? It was not an issue of security, it was an issue of whether it (open source product) was certified.	Material control(ling)
Content mutability Resistance Open source breakdown	We had an issue with document fidelity α documents would not display as they were intended to in the open source application. We tried hard to convert with fidelity- but it wasn't easing their work, these were genuine problems.	Mutating materiality

5 Two Cases of Open Source Software Adoption

Juxtaposing two local council cases of open source software adoption in the UK we highlight their differences and similarities in open source application and implementation. One can now be distinguished by its 'success' (Camden Council) whereas the other (Bristol City Council) has undergone a very mixed engagement with open source. Our narratives indicate that for both cases there was strong goodwill towards open source right from the start and yet the trajectories of implementation are widely different.

Camden Council guided the open source process internally with a strong manager as leader. He built up a team of IT staff over the last ten years that progressed from a simple interest in open source to what is now considered to be an evangelist team of highly skilled developers. Camden co-created on an open source project and is now

able to offer its expertise to other local councils that share an interest in moving to open source software and platforms.

Bristol City Council was in the media spotlight from the moment it announced its open source intentions. There was a grand move to open source desktop software use. However, this euphoric open source sentiment did not last for more than a year, after which stories of open source failure began to leak. Open source software was then discarded and Bristol was forced to return to proprietary software. More recently Bristol has shown a renewed interest in open source but this time there is a more cautious approach to such change.

There are many similarities across both cases in the UK and we found it intriguing to make sense of where in this process of open source adoption did open source play a role, and what provoked the failure in one case and the relative success of the other.

Table 2. Chronological Tracing of Open Source Software for CC.

Year	Tracing IT/IS Events at Camden Council
2001	Took part in the Pathfinders project by submitting a proposal for an open source content management system which would be reused by other local councils.
2002	CC won some funding for the CMS. A partnership of five local councils began work on the CMS in collaboration with Philip Greenspan of the MIT.
2003-04	Won second round of funding to build an expanded version that had more functionality, be easier to install and would work on an open source database as opposed to Oracle. The CMS was taken up by 30 UK local authorities and then also Australia, Malaysia and China. CC faced growing complexity of working closely with an external open source community with different motivations, deadlines and agenda.
2004	Began to build an ecosystem around their open source projects by enrolling SME help and enthusiasm (and also the funding they had left).
2005	The ecosystem proved harder to maintain as there were not enough tenders to keep the companies afloat. The number of support vendors began to disappear from the market forcing CC to find other forms of support. CC found another academic institution that was willing to work on the CMS.
2006	CC created a validation process for incoming contributions from external sources. CC hired the services of Red Hat to objectively validate the contributions thus creating a meritocratic process of acceptance. CC toyed with the idea of setting up a subscription payment for the CMS but dismissed it considering that this would not be open source friendly.
2005-2011	CC now works in conjunction with the academic institution that took up the development of the CMS. The academics nurture the community and manage it. The development is done partly by university students, CC IT staff and some contributions are from external developers and opens source communities.
2011-12	CC has a strong team of IT developers and staff that are busy building up open source projects, and expertise. The growing concern is however, that this team and its energy are being nudged towards more maintenance work rather than challenging new open source projects.

5.1 Camden Council (CC)

Camden council and its move towards open source has been lead by a strong IT leader who has a reputation for doing things differently. The clear objective is to have a content management system which can manage the load of thousands of constituents and where many dissimilar functions are possible. The decision to go open source was not dictated by any form of ideology but rather it was a practical decision based on a need to work with other local councils and cut back on a waste of resources and expertise. Table 2 traces the chronological history of open source adoption by Camden Council.

Camden Council faced numerous challenges with its decision to adopt and co-create open source software in the form of community management, limited funding, a dying ecosystem of SME vendor support, and lack of uptake of the CMS by as many councils as expected yet it has persisted in its endeavor with some good degree of success.

5.2 Bristol City Council (BCC)

Approximately 15 years ago (1997-1998) the IT staff of BCC were using open source software to support the council's first experiments with websites (see Table 3). There was no philosophy or political backing for open source at this point. It was simply a practical need to create a website that led the BCC IT staff to use open source software options. The interviewees added that there were always 'overtones of being open' but no clear direction or strategy was apparent in the first five years.

Evaluating various open source options has not been cheap – indeed this part of the selection process was lengthy and consuming in terms of time and various resources. This expense would be more acceptable if it had led to a viable set of OSS options. The issue was that BCC was recommended a package where numerous products were 'sewn together' to provide a solution that did not meet all the functional requirements. While Microsoft offered BCC the usual standard government option – no extra expenses or strings attached – and still emerged cheaper than the OS option.

BCC has a roadmap for changes required over the next few years. This will no doubt entail exit costs but BCC believes that with the many companies offering support in migration from proprietary to open source software nowadays there is a clearer idea (and value attached) to exit costs, which makes migration less problematic and fuzzy.

Table 3. Chronological Tracing of Open Source Software for BCC.

Year	Tracing IT/IS Events at Bristol City Council
1997-98	Open source used to create and host the first BCC website
2001-2004	BCC evaluated and selected StarOffice rather than Microsoft products because the former was cheaper. Initial research with local government system vendors indicated that integration with StarOffice would be technically possible, and several key vendors expressed a willingness to do the work necessary.
2005-06	StarOffice rolled out across the council to over 5000 users, supported by 8 person team of floorwalkers and technical specialists, providing document conversion, training and coaching to staff. BCC shared experience and learning from evaluating and selecting StarOffice by publishing guidance documentation via ‘Open Source Academy’, a UK National e-Innovations Project.
2007-08	BCC continued to invest in StarOffice, rolling out version 8 and working with Sun and key third party integration system vendors (ISVs) in the local government market, seeking to establish integrations with a variety of business systems. Vendors began to step back from willingness to do integration work without Bristol contributing significant extra funds, citing lack of market take-up of StarOffice and open document format (ODF). BCC staff shared challenges and options for addressing them with Sun, IBM, and other EU governments interested in the issues in a series of conferences and seminars.
2008-09	Continued use of StarOffice became problematic, as fidelity of file conversion was still not as effective as necessary for problem free ‘round-tripping’ of documents, and lack of system integration meant that a high number of MS Office licenses had to be retained.
2009	BCC adopted a new Information Systems & Technology Strategy and restructured the ICT Service, introducing Enterprise Architecture and establishing a new approach to evaluating and selecting applications and technology to ensure fitness for purpose. Open Standards were made a formal and central part of the strategy at this point. EA team recommended that due to experiences between 2005 and 2009, it was no longer viable to continue using StarOffice, and that MS Office should be adopted. Project created to develop business case and plan.
2010	BCC Cabinet approve business case for adopting Windows 7 and MS Office 2010 on all council PCs, with other elements of ‘desktop and collaboration’ software stack to be selected through a rigorous ‘level playing field’ approach of comparing OSS and proprietary options to business requirements. OSS would be selected wherever it met requirements and provided best value for money. Computacenter and Sirius (OSS SME subcontractor) were taken on as System Integrators to design and deliver the project.
2011	Breakdown of relations between Computacenter (SI) and SiriusIT SME support led to delays in selection and design stages. LinuxIT eventually selected as replacement by Computacenter. Some OSS options were selected, e.g. Big Blue Button for web-conferencing and video conferencing, and Alfresco for team collaboration. MS software selected for other elements, e.g. email, IM and presence management, directory services. EA team led separate series of evaluations and selections for an integration platform, business process management system (BPMS), website, and electronic document and records management system (EDRMS). Open standards based products from Tibco chosen for integration and BPMS. OSS products Drupal and Alfresco chosen for website and EDRMS.

6 Analysis

Our study finds that becoming occurs at different speeds [3, 50]. Considering the uncertain nature of becoming this is not so surprising, but what is more interesting is the question of the nature of becoming – how and what can impede or accelerate the process? Our data shows that the becoming of adoption can be both constrained and precipitated by various forms of *materiality* (of the assemblage of the open source ecosystem) [51]. Open source and its transparent process, character, code and license do not necessarily lead to more transparency. The interview material lead to this interesting conclusion forcing us to reconsider and unpack open source software based on our data (and literature) – see Table 4.

6.1 Mutability of Open Source

Open source software and its development process have a number of key elements (see Table 4) such as license [52], community [53], the code [54], coordinating mechanisms [55], and documentation [56]. This is a fairly familiar characterization of OSS but we want to highlight how all these characteristics are not fixed – even within the same project. They are changing, indeed in a state of constant becoming. Scant attention, if any has been paid to the idea of how OSS mutates within a project or over time. The two cases, in their own manner, emphasize how malleable and yielding OSS was, and is. The license of an open source product can range from a variety of accepted (OSI approved) types, however, each license offers some form of viral control mechanism. Some licenses like the General Public License (GPL) are more viral than others. This in turn affects the ability of code to mutate and restricts the variation in becomings possible. The materializing of each element matures the becoming and expedites it in a manner that makes further (variations in) becomings less likely (see Table 5). Thus open source use and adoption can be controlled and managed. In the case of BCC their choice of enterprise open source software was based on an open core model rather than a more ‘pure’ open source license. Such a model implied that the enterprise edition of the software being procured by BCC was actually not strictly open source as the code was not necessary viewable. Open core models are a form of dual licensing where there is an open source version whose functionality is often limited by comparison to the enterprise open core version, thus giving rise to the term *crippleware* to describe the reduced functionality OS version. Such choices are becoming more common yet as were told by the developer team, such a model often undermines community contributions eventually killing the project itself.

Community in turn, is a multifaceted phenomenon where variations are visible in the level of skills and expertise of the members and contributors. The size of the community in both cases not only varied but there was a constant flux of developers experienced. Another form of mutability was introduced into the BCC case because it relied on commercial vendor support. Such projects can see diverse forms of sponsorship and resource injection which give rise to changing loyalties, and focus. On the other hand, when questioned about the community support side of their OS

project, CC replied that, “I suppose there was and there wasn’t. Because there is a community of local government. There are...potentially 399... members of that community. There wasn’t at that time any kind of community of webmasters. So there were these relatively new posts being created in local authorities but there wasn’t any communication or anything set up to communicate. So that’s one of the first things we did as part of the project, we set up five webmasters from the five partners. But then we got them to try and go out and invite people ...we did a couple of workshop meetings where we just invited people. So that was the basis of the community.

Table 4. Implications of Open Source Mutability.

Mutability of Open Source		
	Areas and Level of Mutability	Implication
License	<ul style="list-style-type: none"> • Choice of license • Version of license • Level of reciprocity involved • Level of transparency 	The varied viral nature of some licenses makes them more (or less) amenable to change. Dual licenses are yet another form of mutability.
Community	<ul style="list-style-type: none"> • Skill level of members • Core team size • Turnover rate • Number of company backed employees 	A community with a strong core team of developers backed by company resources and high skill level has greater potential to adapt and change.
Code	<ul style="list-style-type: none"> • Level of stack • Reusability • Language • Modularity 	Code, depending on which language is used, the level of application or product being built and its reusability can affect the sort of mutability possible.
Coordinating Mechanisms	<ul style="list-style-type: none"> • Public or private discussion groups • Face-to-face meet-ups • Levels and types of mailing lists • Access level of version control software 	Coordinating mechanisms in open source are key. Some mechanisms are open to the public, whereas others need to be for developer only access. Such variations in access can blur the level of transparency offered.
Documentation and Metadata	<ul style="list-style-type: none"> • The type of (detail) documentation provided • Level of updating documentation • Access to metadata • Search-ability of documentation and metadata 	Documentation in open source can be patchy and incomplete thus eroding transparency and changing the mutable nature of open source.

Source code, depending on which language is used, the level of application or product being built and its reusability can affect the kinds of mutability possible. As can the variety of coordinating mechanisms at the disposal of a developer community. Much of the discussion about development is carried out over public forums but this is not always true. There is also a growing trend for face-to-face meetings in open source development where traceability of ideas is less transparent and archivable. The various OS projects in both cases used tools such as version control to manage the code, contributions and metadata as explained by an IT manager at CC, “the software does have full version control and there is you know, a nightly build that kind of rolls

up all of the code contributions and produces the head build as opposed to like a version” but as he clarified that not all members of the community had equal access to all levels of the tool and code. This again built in varying degrees of transparency and mutability.

6.2 Materializing of Open Source – Speed and Time Control

In table 4 we illustrated how each element of open source software like the license, community, code and so on, encapsulates the potential for more or less mutability. Our two cases of open source adoption by the public sector in the UK narrate this story, and help us to explain the difference in adoption ability and ‘success’ of both in terms of mutability of open source software, and how this mutability was constrained or encouraged by the material inscriptions adopted for manoeuvre. As much of Science and Technology Studies literature explains materiality is more than tangible ‘things’, it includes ideas, feelings, and silent action.

It may well be considered that when anything becomes more materialized that it would be less vague and opaque, however, we found that this was not necessarily true. In fact, there was little ability to trace all the possible trajectories of becoming when the situation was as complex as a politically infused public sector organization. There were more than one possibilities for mapping but experienced bureaucrats in both councils were able to manage the possible rupturing of the adoption process. Instead of building transparency into the system with a greater reliance on materiality, the local council IT staff and policy writers were able to contain the opaqueness in a strategic manner to their advantage.

In both councils we saw that license of the software was a key concern. For BCC the open core model created complications and a strong possibility for lock-in. however, it was in its practicing (of the license) that the license materialized. Each license of open source may be slightly different but they are all alike in behaving as the Constitution of the project. As Constitution it dictates what can be done with the code, who owns it and how this ownership can change in processes of redistribution and even multiple licenses. Camden Council adopted a single (as opposed to dual license) and it was the BSD. With the BSD it is possible for anyone to take the source code and change the license of their particular strand. So though the practicing of the license makes open source more material it does little to solidify its mutability thus leaving the possibility of managing the speed of open source adoption. Another manner of understanding such becoming made visible was to make sense of in terms of time [57]. More than license and thus different code branches make visible multiple becomings or *multiple parallel times*.

The community leaves traces in its process of collaboration, turnover, expertise sharing and so on. Its materializing is manifested in this very change and flux where members join while others leave. Other forms of materializing involve sponsoring employees to become a part of the community, and the training (through the community or on the job) of members. This is part of the process of making open source software prepare for *future present time*. The possible future is being pulled back in the present becomings to force a certain tracing (not mapping) of the adoption

process. This is because the future is unknown so the way the councils made sense of future software and requirement needs was by using the present as indication. However, this becoming was constrained as and constraining as new emergent changes were inevitable – yet by attempting to trace the future both councils were restricting new possibilities.

Table 5. Becoming Manifested as Materializing.

<i>Becoming and Materializing</i>		
Areas and Level of Mutability	Materializing	Time
<ul style="list-style-type: none"> • Choice of license • Version of license • Level of reciprocity involved • Level of transparency 	License is the Constitution – written and in practice. <ul style="list-style-type: none"> • Changing license • Practicing the license • Implementing the license 	Multiple parallel times
<ul style="list-style-type: none"> • Skill level of members • Core team size • Turnover rate • Number of company backed employees 	Community traces, voices, decisions, sense of belonging (expressed through T-shirts, brand, etc). <ul style="list-style-type: none"> • Community in flux • Sponsoring employees • Training of developers 	Future present time
<ul style="list-style-type: none"> • Level of stack • Reusability • Language • Modularity 	Code, requirements, functionality and use <ul style="list-style-type: none"> • Changing requirements • Greater encapsulating • Reusing code 	Hiding time Revisiting time
<ul style="list-style-type: none"> • Public or private discussion groups • Face-to-face meet-ups • Levels and types of mailing lists • Access level of version control software 	Coordinating Mechanisms <ul style="list-style-type: none"> • Making discussions transparent • Increasing security level of access • Varying governance structure to cope (change) access levels 	Traceable time
<ul style="list-style-type: none"> • The type of (detail) documentation provided • Level of updating documentation • Access to metadata • Search-ability of documentation and metadata 	Documentation and Metadata <ul style="list-style-type: none"> • Making the search algorithms visible (or not) • Maintaining documentation • Detailed documenting and instructions 	‘Moment’ of time (capturing)

Speed of becoming is managed through the code by controlling the changing nature of requirements, varied forms of encapsulation and even encouraging reuse of code.

These are material forms of the code where traces are left and can be followed. Code and its materializing thus make it possible to *hide time*, yet at the same *revisit time*. Camden Council was hopeful of reusing its code and system across other councils and did manage this for a while, “and so we built our proposal to the funding around that basis that we were going to produce an open source content management system that would be reused by other local authorities... And so that release got taken up by more than 30 UK local authorities and then started being taken up in Australia and Malaysia and China and all kinds of places in the world. However, a shrinking ecosystem of vendor support over time made a change in code less possible. This in turn led to impaired materializing of the code, and the objectives of Camden Council. Traceability and materializing in relation to open source coordination are (theoretically) built into the open development process. Discussions between developers are made visible, and traceable, access to discussion forums and version control software are managed by security levels, and we also found in our data that the governance structure changes in relation to security access and expertise of the developer. The materiality of code makes time *traceable* and retraceable (though each retracing will no doubt be a variation of other becomings and not quite a tracing). Finally, time, or moments of it can be *captured*. This reinforces the idea of materializing of open source because good documentation of code makes algorithms visible, and future documentation easier and more possible. However, as the example of BCC shows poor documentation, and impaired interoperability can force a breakdown of software use. The fact that other councils that BCC needed to work with did not use open source made it difficult for BCC to share documents without trouble. Often the documents created by BCC using open source were not rendered in the expected manner by other councils or were completely illegible, “we had an issue with document fidelity – documents would not display as they were intended to in the open source application”.

7 Conclusion

Considering each element of open source individually is useful to understand mutability, materializing and transparency (or lack of) but as one interviewee explained open source is complex and has ‘vectors of lock-in’. It is an entanglement of all these elements in proportions that are beyond complete control that build in uncertainty make the becoming of open source software so challenging. The data revealed a richness in its material element (as the codes show). Literature on open source procurement and adoption in the public sector has not only ignored this idea but we find that it is in general (though not true for all) often atheoretical. We have attempted to redress this issue by sensitizing our data collection and analysis with ideas of becoming. Such an ontology allowed us to move beyond a focus on only the human [15], or practices, or likening change to merely improvising [16]. There has been more recent work in IS that shows concern for a relational ontology where the social and material are understood to be entangled [58] and imbricated [59] but there

has been little use of such ideas to understand OSS in the public sector, and how this implicates the process of becoming.

In this paper we understand better how the becoming of open source software adoption by two different local councils in the UK indicate that the process of becoming occurs at different speeds because of the nature of their materiality. Our data shows that the becoming of adoption can be both constrained and precipitated by various forms of materiality (of the assemblage of the open source ecosystem) [51]. The interesting point of departure of our study is how open source software – a much touted transparent and open phenomenon – is by its nuanced and layered mutability [60, 61] able to make the process and practices surrounding it *less visible*.

Acknowledgement

This research was funded by the UK Cabinet Office, OpenForum Europe and a few other organizations.

References

1. J. Law and V. Singleton, "Performing Technology's Stories: On Social Constructivism, Performance, and Performativity," *Technology and Culture*, vol. 41, pp. 765-775, 2000.
2. H. Tsoukas and R. Chia, "On Organizational Becoming: Rethinking Organizational Change," *Organization Science*, vol. 13, pp. 567-582, September-October 2002.
3. I. Colville, *et al.*, "Simplexity: Sensemaking, organizing and storytelling for our time " *Human Relations*, vol. 65, pp. 5-15, 2012.
4. G. Deleuze and F. Guattari, *A Thousand Plateaus. Capitalism and Schizophrenia*. Minneapolis: University of Minnesota Press, 1987.
5. B. Latour, "Give Me a Laboratory and I Will Raise the World," in *The Science Studies Reader*, M. Biagioli, Ed., ed New York/London: Routledge, 1999, pp. 258-275.
6. B. Latour and J. Johnson, "Mixing humans and nonhumans together: The sociology of door-closer," in *Ecologies of knowledge: work and politics in science and technology*, S. L. Star, Ed., ed: SUNY Press., 1995, pp. 257-277.
7. M. Callon, "Some Elements of a Sociology of Translation: Domestication of the Scallops and Fishermen of St. Brieuc Bay," in *Power, Action and Belief: A New Sociology of Knowledge?*, J. Law, Ed., 32 ed London: Routledge & Kegan Paul, 1986, pp. 196-233.
8. M. Callon, "The Sociology of an Actor-Network," in *Mapping the Dynamics of Science and Technology*, M. Callon, *et al.*, Eds., ed London: Macmillan, 1986.
9. J. Law, "After ANT: Topology, Naming and Complexity," in *Actor Network Theory and After*, J. Law and J. Hassard, Eds., ed Oxford and Keele: Blackwell and the Sociological Review, 1999.
10. K. Barad, "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter," *Signs: Journal of Women in Culture and Society*, vol. 28, pp. 801-831, 2003.
11. M. Callon, "What Does it Mean to Say that Economics is Performative?," in *Do Economists Make Markets?: On the Performativity of Economics*, D. MacKenzie, *et al.*, Eds., ed New Jersey: Princeton University Press, 2007, pp. 311-357.

12. D. Robey and S. Sahay, "Transforming Work Through Information Technology: A Comparative Case Study of Geographic Information Systems in County Government," *Information Systems Research*, vol. 7, pp. 93-110, 1996.
13. C. U. Ciborra, Ed., *From Control to Drift*. Oxford: Oxford University Press, 2000, p.^pp. Pages.
14. L. Gasser, "The Integration of Computing and Routine Work," *ACM Transactions on Office Information Systems*, vol. 4, pp. 205-225, 1986.
15. M. Feldman, "Organizational routines as a source of continuous change.," *Organization Science*, vol. 11, pp. 611-629, 2000.
16. W. J. Orlikowski, "Improvising organizational transformation over time: A situated change perspective," *Information Systems Research*, vol. 7, pp. 63-92, 1996.
17. E. Monteiro, *et al.*, "The family resemblance of technologically mediated work practices," *Information and Organization*, vol. 22, pp. 169-187, 2012.
18. M.-C. Boudreau and D. Robey, "Enacting Integrated Information Technology: A Human Agency Perspective," *Organization Science* vol. 16, pp. 3-18 January/February 2005.
19. B. Fitzgerald, *et al.*, *Adopting Open Source Software: A Practical Guide*: MIT Press, 2011.
20. S. Phipps. (2011, March 3rd, 2011) Open Source Procurement: Subscriptions. *ComputerWorldUK* [blog post]. Available: <http://blogs.computerworlduk.com/simon-says/2011/03/open-source-procurement-subscriptions/index.htm>
21. G. Archer, "Open Source Software Policy: Australian Government Information Management Office (AGIMO) Circular," ed. <http://fwd4.me/wY6>: Australian Government: Department of Finance and Deregulation, 2010.
22. V. Kundra, *et al.*, "Technology Neutrality: Memorandum for Chief Information Officers and Senior Procurement Executives," ed. <http://fwd4.me/wY5>: Washington DC Executive Office of the President: Office of Management and Budget, 2011.
23. T. Waring and P. Maddocks, "Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future," *International Journal of Information Management*, vol. 25, pp. 411-428, 2005.
24. A. Oram, "Promoting Open Source Software in Government: The Challenges of Motivation and Follow-Through," *Journal of Information Technology & Politics* vol. 8, pp. 240-252, 2011
25. A. Shaw, "Insurgent Expertise: The Politics of Free/Livre and Open Source Software in Brazil," *Journal of Information Technology & Politics* vol. 8, pp. 253-272, 2011.
26. R. A. Ghosh, *et al.*, "Guideline on public procurement of Open Source Software," ed. Brussels, Belgium: <http://fwd4.me/wXx>: UN University/MERIT and Unisys Belgium, 2010.
27. M. Ballard. (2011, Friday 25 February 2011) Government IT suppliers claim procurement system excludes open source. *ComputerWeekly.com*. Available: <http://www.computerweekly.com/Articles/2011/02/25/245598/Government-IT-suppliers-claim-procurement-system-excludes-open.htm>
28. C. Saran. (2010, Wednesday 22 September 2010) Government plans procurement overhaul to slash IT spend. *ComputerWeekly.com*. Available: <http://www.computerweekly.com/Articles/2010/09/24/242958/Government-plans-procurement-overhaul-to-slash-IT-spend.htm>
29. K. Hall. (2011, Wednesday 23 February 2011) Government tells major IT suppliers - we want more open source software. *ComputerWeekly.com*. Available: <http://www.computerweekly.com/Articles/2011/02/23/245555/Government-tells-major-IT-suppliers-we-want-more-open-source.htm>

30. R. Burkhardt, "Seven Predictions for Open Source in 2009," <http://drdobbs.com/open-source/212700284>, Ed., ed, 2008.
31. B. S. Noveck, "Defining Open Government," <http://cairns.typepad.com/blog/2011/04/whats-in-a-name-open-gov-we-gov-gov-20-collaborative-government.html>, Ed., ed, 2011.
32. B. Russo and G. Succi, "A Cost Model of Open Source Software Adoption," *IJOSSP*, pp. 60-82, 2009.
33. [33] R. Gallopino, "Open Source TCO: Total Cost of Ownership and the Fermat's Theorem," <http://robertogaloppini.net/2009/01/08/open-source-tco-total-cost-of-ownership-and-the-fermats-theorem/>, Ed., ed, 2009.
34. P. Agerfalk and B. Fitzgerald, "Outsourcing to an Unknown Workforce: Exploring Open sourcing as a Global Sourcing Strategy," *MIS Quarterly*, vol. 32, pp. 385-400, 2008.
35. R. Sutor, "Managing open source adoption in your IT organization," <http://www.sutor.com/newsite/blog-open/?p=3260>, Ed., ed, 2009.
36. M. Shaikh and T. Cornford, "Understanding Commercial Open Source as Product and Service Innovation," in *2011 Academy of Management Annual Meeting*, San Antonio, Texas, USA, 2011.
37. P. Zuliani and G. Succi, "Migrating public administrations to open source software," in *E-society 2004 IADIS International Conference*, Avila, Spain, 2004, pp. 829-832.
38. P. ZULIANI and G. SUCCI, "An Experience of Transition to Open Source Software in Local Authorities," in *E-challenges on Software Engineering*, Vienna, Austria, 2004
39. J. Dinkelacker, *et al.*, "Progressive Open Source," in *Proceedings of the 2002 ACM International Conference on Software Engineering (ICSE'02)*, 2002, pp. 177-184.
40. L. Dahlander, "Penguin in a newsuit: a tale of how de novo entrants emerged to harness free and open source software communities," *Industrial and Corporate Change* vol. 16, pp. 913-943, 2007.
41. B. Fitzgerald, "The Transformation of Open Source Software," *MIS Quarterly*, vol. 30, pp. 587-598, September, 2006 2006.
42. S. O'Mahony, *et al.*, "IBM and Eclipse (A)," *Harvard Business Review Case Study*, vol. December 16, 2005.
43. S. Clegg, *et al.*, "Learning/Becoming/Organizing," *Organization*, vol. 12, pp. 147-167, 2005.
44. K. E. Weick, "Improvisation as a mindset for organizational analysis.," *Organization Science*, vol. 9, pp. 543-555, 1998.
45. K. E. Weick and R. E. Quinn, "Organizational change and development," *Annual Review of Psychology*, vol. 50, pp. 361-386, 1999.
46. J. Austin, *How to do Things with Words*. Clarendon: Oxford, 1962.
47. J. Butler, *Excitable Speech: A Politics of the Performative*. London: Routledge, 1997.
48. G. Bateson, *Steps to an ecology of mind*. New York: Chandler, 1972.
49. A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*: Sage Publications, 1999.
50. K. E. Weick, "Organized sensemaking: A commentary on processes of interpretive work " *Human Relations*, vol. 65, pp. 141-153, 2012.
51. J. Rose and M. Jones, "The Double Dance of Agency: a socio-theoretic account of how machines and humans interact," in *ALOIS 2004 - Action in Language, Organisations and Information Systems*, Linköping - Sweden, 2004.
52. Y. Benkler, "Coase's Penguin, or, Linux and the Nature of the Firm," *Yale Law Journal*, vol. 112, pp. 369-446, 2002.

53. S. O'Mahony and F. Ferraro, "The emergence of governance in an open source community," *Academy of Management Journal*, vol. 50, pp. 1079-1106, 2007.
54. B. Fitzgerald and J. Feller, "A further investigation of open source software: community, co-ordination, code quality and security issues," *Information Systems Journal*, vol. 12, pp. 3-6, 2002.
55. K. Crowston and E. E. Kammerer, "Coordination and collective mind in software requirements development," *IBM Systems Journal*, vol. 37, pp. 227-245, 1998.
56. G. von Krogh, *et al.* (2003). *Collective Action and Communal Resources in Open Source Software Development: The Case of Freenet*. Available: <http://opensource.mit.edu/papers/vonkroghhaefligerspaeth.pdf>
57. D. Kavanagh and L. Araujo, "Chronigami: Folding And Unfolding Time.," *Accounting, Management and Information Technology*, vol. 5, pp. 103-121, 1995.
58. W. J. Orlikowski and S. V. Scott, "Sociomateriality: Challenging the Separation of Technology, Work and Organization," *The Academy of Management Annals*, vol. 2, pp. 433-474, 2008.
59. P. M. Leonardi and S. R. Barley, "What's Under Construction Here? Social Action, Materiality, and Power in Constructivist Studies of Technology and Organizing," *The Academy of Management Annals* vol. 4, pp. 1-51, 2010.
60. A. Mol and J. Law, "Regions, Networks and Fluids: Amaemia and Social Topology," *Social Studies of Science*, vol. 24, pp. 641-671, 1994.
61. I. Moser and J. Law, "Fluids or flows? Information and qualcalulation in medical practice," *Information Technology & People*, vol. 19, pp. 55-73, 2006.