Maha Shaikh and Tony Cornford

Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe

Report

The report has been jointly financed by the Cabinet Office and OpenForum Europe, together with some of its members and partners. In particular sponsorship has been received from Alfresco, Deloitte, IBM and Red Hat.

INFORMATION SYSTEMS
AND INNOVATION GROUP
Department of Management

Total Cost of Ownership of Open Source Software

A report for the UK Cabinet Office supported by OpenForum Europe.

Version 8.5 Final

November 2011

Maha Shaikh & Tony Cornford
Department of Management
London School of Economics
Houghton Street
London WC2A 2AE

maha.shaikh@wbs.ac.uk

This report has been prepared by LSE Enterprise for the UK Cabinet Office. We acknowledge the support given by OpenForum Academy, the think tank and research programme of OpenForum Europe. The report has been jointly financed by the Cabinet Office and OpenForum Europe, together with some of its members and partners. In particular sponsorship has been received from Alfresco, Deloitte, IBM and Red Hat.

We want to emphasise the independence of the report and that no company or organisation, including those named above, has made any direct input to the report conclusions.

This report is also available on the Cabinet Office website at www.cabinetoffice.gov.uk/resource-library/open-source-procurement-toolkit and LSE website at http://eprints.lse.ac.uk\39826

Any queries regarding this document/publication should be sent to Tony Cornford at T.Cornford@lse.ac.uk or Maha Shaikh at Maha.Shaikh@wbs.ac.uk

Table of Contents

Total Cost of Ownership of Open Source Software

Executive Summary

Governments across the world want to save money, indeed they need to save money. At the same time they seek to achieve urgent transformation and reform in their organisational structures - a process that often requires new information systems and data infrastructures.

In the area of information systems and data management the opportunity to acquire new systems within tight budgets or to make operational savings for established systems are limited. One major opportunity however seems to be on offer: to shift to the use of open source software and reap the direct benefits of lower software costs and further indirect benefits such as greater adherence to open standards, more choice of vendor and service supplier, and working to establish flexible incremental architectures. Initiatives in many countries reflect these policy ambitions, including across Europe, in the USA and in the BRIC countries.

One way to validate the wisdom or credibility of such a policy is by careful attention to the costs associated with using open source software as compared to more conventional or established alternatives. The usual conceptual model applied to this is that of 'total cost of ownership' or TCO.

TCO reflects a measure of all the costs of identifying and acquiring software, installing it and operating it, and finally the exit costs found in migrating away from the software. TCO reflects not just the balance of the direct qualities of competing software products (price, functionality, reliability etc.) but also the relationship of the software to the organization's broader set of technology platforms, installed systems, culture and skills base, and strategic goals, as well as the ability to access market and community based services and support.

Participants in this study, all with experience of acquiring and using open source software, were well aware of the concept of TCO and it framed well their understanding of the trade-offs that adopting open source imply. However, they (or their organisations) did not usually use TCO as a formal decision making tool, though some wished that it was. One consequence is that not many organisations maintain a data set that can estimate TCO with much reliability or offer robust comparative evidence.

Nonetheless, when people with experience apply their judgement to the question of TCO many are clear that cost advantages – cost saving and cost avoidance - are achievable, and case studies support this contention.

Based on interviews and case studies this report identifies 14 specific TCO drivers within 5 broad software life cycle stages – Selection, Acquisition, Integration, Use, Retirement. Unsurprisingly the *cost of software* itself is judged the most significant factor and the easiest to estimate, with cost *of maintenance and upgrades* as number two and *cost of contracted support* as number three. Factors that are identified as less easy to judge are in the domain of the consequential *organisational change* and *exit costs* as well as in the establishment of appropriate *in-house services*.

The softer benefits of open source adoption are also widely appreciated – those of flexibility, openness, ability to tweak and customize, and support for open standards and open data - altogether a more open and accessible software environment. This broader vision is where the enduring benefits and associated cost reductions are often seen to come from.

Open source adopters do note that they needed to hire experts and look for support to meet their organisation's ambitions including for control of code and configuration, and taking more direct control of their infrastructure to allow agile innovative responses to changing needs. As reported, the support desired is less the large scale outsourcing practiced in the current Systems Integrator marketplace. Rather these adopters desire a more targeted and technically focused support that can be delivered by smaller and perhaps more local suppliers.

Adoption and development of open source can support the sharing of both expertise and expense between government bodies, for example among local authorities facing very similar needs. Just as the business sector uses open source models of collaborative work as a flexible and efficient way to work with complementors or even rivals on mutually beneficial projects, so for public bodies it can serve as a flexible route to productive collaboration.

There is a perception among many that if open source is to become an accepted and substantial part of information systems activity within the public sector then it needs government-level policies to sustain the change including an overhaul of procurement processes and practices. Most see this as an essential levelling of the playing field that can provide open source service companies, particularly in the SME sector, with new opportunities for growth.

The growing body of experience among public sector bodies reveals important lessons for political leaders, public sector policy makers and IT managers.

- Open source is not just or only or always about 'cheap'. But it can bring a number of distinct and enduring benefits when contrasted to proprietary software, and many of these are directly financial.

- Pragmatism needs to guide open source adoption and not ideology. This pragmatism is reflected in the need for a strategy and a plan to work towards open source options. It is not a quick fix, low hanging fruit or an easy win-win scenario. Any organisation interested in adopting open source software must not foster false hopes, and then suffer premature feelings of failure.

- Migrating to open source is more likely to be successful if it is done when there is a real and present need for change or a new approach, rather than simply on the basis of finding open source attractive on infrastructure cost arguments.

- Adoption of open source can be part of building a more agile organisation able to innovate and respond to change. It can also be part of (re)building in-house expertise and regaining control.

- Success with open source software requires that senior managers within public organisations understand the potential and the obstacles, and provide sustained leadership. Their role is to support and promote an appropriate regime for software acquisition but also to endorse a shift in thinking or new vision for the wider systems strategy.

- As for any change programme moves to exploit open source software needs strong champions, wide commitment, planning and follow through. Innovation can be risky, (but ultimately rewarding), and risk requires institutional as much as business unit or individual commitment.

About the Authors

Maha SHAIKH is an Assistant Professor at Warwick University Business School. Prior to this she was a Research Associate at the London School of Economics and Political Science (LSE). Other affiliations include the University of Limerick where she worked on a number of projects including the OPAALS project with Professor Brian Fitzgerald. She has also worked with Professor Leslie Willcocks at the LSE studying the relationship of open source to outsourcing, open innovation and open business models. Dr Shaikh is a coauthor of Adopting *Open Source Software: A Practical Guide,* to be published in late 2011 by the MIT Press.

Tony CORNFORD is Senior Lecturer in the Department of Management at the London School of Economics and Political Science. His research and advisory work focuses on e-Government and IT and health, particularly systems supporting the use of drugs and medicines. He has published work on open source software processes and business models.

Introduction

1. Getting more for less, driving down costs and driving up efficiency are at the core of many governments' response to the financial crisis. This has led central and local government bodies across the European Union, in Australia[1] and the USA[2] to take renewed interest in the role of open source software as a part of cost saving policies. This ambition is also reflected in the recent European Commission commissioned report on *Guideline on public procurement of Open Source Software*[3]. Open source software, along with open data policies and the promotion of open standards has become a significant part of the language that governments speak when they consider how to develop and manage their future information systems and data infrastructures[4]. Open source is, for example, one of the basic building blocks of the US government open government initiative - http://www.whitehouse.gov/Open[5].

2. This report is intended to provide a balanced assessment of the potential of open source software within the public sector based on evidence collected from those who have taken this path and from members of the community of firms offering support services to such public bodies. The audience we are addressing includes politicians and senior decision makers across central and local government, senior IT managers and the supplier communities.

3. Open source alternatives for substantial operational software are, relatively speaking, new in much of the public sector. We thus lack a strong body of evidence of the costs and benefits of its adoption. It is not obvious or inevitable that making such choices will translate to lower overall costs for developing and running systems even if open source software can be an important part of the delivery of other policy goals[6]. The evidence on which such judgement can be made is scarce, in part because the extent of open source adoption by the public sector varies across Europe and within countries, hence obtaining good evidence of its costs and benefits, including financial savings is not easy.

4. Spain is often presented as a country where the use of open source software has grown and sustained itself within the public sector. Beyond Europe Brazil has achieved a high level of open source software use in the public sector. Indeed openness has for some time been emphasized explicitly by Brazilian political parties, and promoting the use of open source has for example featured as part of election campaign messages[7]. Meanwhile in the USA many state and City governments have undertaken various types of open source initiatives.

5. Thus around the world governments have started to take actions to foster open source alternatives. It is, for example, now widely understood that public bodies are unable to choose

★★★★★★★★★★

The ways in which acquisition decisions are made is in part a financial issue, and in part one of balancing many other legitimate concerns and the recognition of certain unbending constraints.

★★★★★★★★★★

open source software, if and when appropriate, if it is not offered as an option by major system integrators (SI). One response, the EU guidlines, is mentioned above while in the UK the Cabinet Office [8] has met with the large and influential system integrators, those who develop and run most central government information systems, to emphasise the need to have open source choices offered within procurement activities[9]. This understanding is also reflected in the recent House of Common Public Administration Select Committee report [10]. But more fundamentally, we need to ask, "if such a choice is offered, will it prove to be advantageous?"

6. The true and complete costs associated with adopting any new software, be it open source or not, are never easy to evaluate *ex post*[11]. Open source software of course brings close to zero license costs (the easy bit), however, open source applications and support services derive from different and rapidly evolving supply chains drawing on a number of new business models thus complicating the ability to compare the costs of open source software with propriatory alternatives.

7. Experience from the private sector certainly suggests that adopting open source software can bring benefits[12]. But it also suggests that such moves are undertaken for a mix of reasons which can include the promise of reduced net costs, but also reflect more strategic goals and constraints that are bound up with a desire to retain freedom of movement and to support innovation[13]. The public sector would like to enjoy all these benefits too. Thus early adopters of open source applications in the public sector quote benefits such as reduced vendor lock-in as one of their key arguments alongside lower costs[14].

8. Early adopter local authorities in the UK have also found that the costs of adopting open source software are not directly translatable into the existing frameworks used for evaluating proprietary software and making acquisition decisions. They also report that cost profiles tend to differ if the software is supported by a conventional vendor, an open source value added consultancy, or when working alongside an open source community (see appendix E).

9. Reviewing a number of industry studies[15], we see that indeed costs are often reported as lower overall for open source but there is some difference as to where the costs emerge and at what stage of adoption.

10. For any organisation setting out to acquire a substantial resource or infrastructure including software choices need to be made between alternative offerings. These choices can be in part made using narrow financial or economic criteria once basic functional thresholds have been passed, but they also demand wider and often qualitative considerations. Even in their own terms financial criteria may not tell the whole story – for example making a choice based on comparison of net present values over 10 years may not take into account limits on year one capital expenditure.

11. So too with the acquisition of information systems and the software that supports them. The ways in which acquisition decisions are made is in part a financial issue, and in part one of balancing many other legitimate concerns and the recognition of certain unbending constraints. The concept of Total Cost of Ownership (TCO) is one way to help to bring together financial concerns with the wider organizational context and the temporalities of software costs, providing the multi-faceted account to inform decision making[16].

12. The TCO approach considers software, and the information systems it supports, placed within its organisational context and related to the business ecosystem from which software, support and services are drawn. The 'total' in TCO is expressed by use of an extended life cycle model which recognises the various stages through which software goes, from selection through acquisition, implementation, use and finally decommissioning (retirement). Figure 1 shows such a simple 5 stage lifecycle and the costs of two software alternatives plotted per stage and cumulatively over the life cycle. In both cases the cumulative cost (before any Net Present Value (NPV) calculations) is the same, but these costs occur at different stages of the life cycle and as the software interacts with other aspects of the proximal and extended context of deployment.

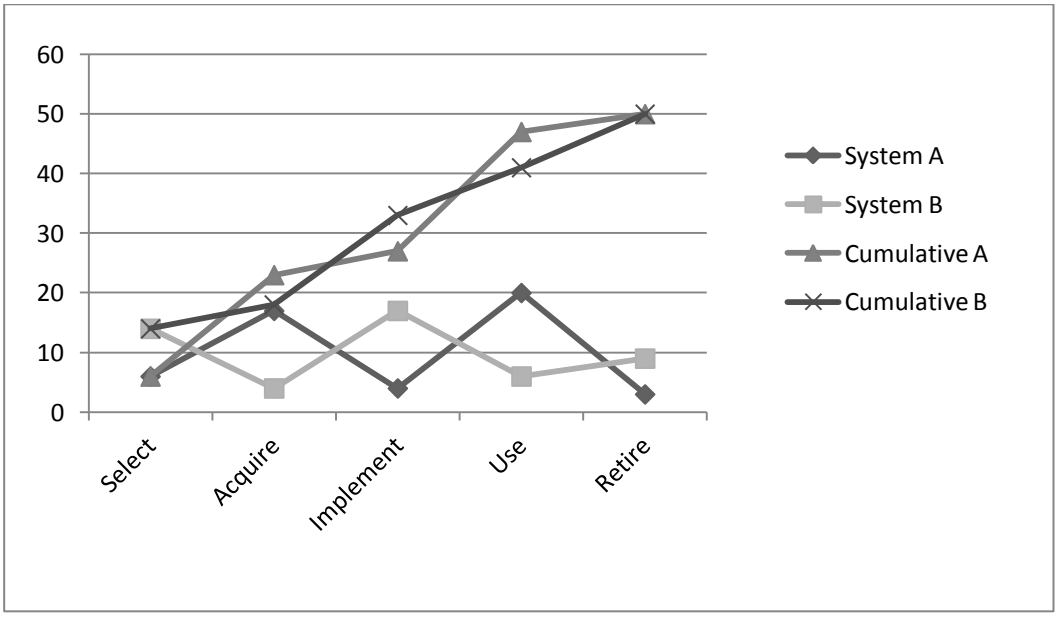

Figure 1. A simple TCO plot for two software products

13. In this report we develop Lerner and Schankerman's[17] definition of total cost of ownership (TCO). These authors propose TCO as "a measure of the total cost of owning and operating a piece of software, including both the initial monetary cost of purchasing it and any associated costs of implementing it". This definition encompasses both the capital costs of acquisition and the operational costs (CAPEX and OPEX). It does not, however, include the exit costs which we consider as potentially significant. Equally we are explicit as to the search costs that precede acquisition.

★★★★★★★★★★

In many relevant respects open source software is different to proprietary software. It places different demands on, and offers different benefits to, the host organisation, is embedded in different software ecosystems and is served by different supply chains

★★★★★★★★★★

14. Our developed definition is thus, "TCO reflects a measure of all the costs of identifying and acquiring software, installing it and operating it and the exit costs found in migrating away from the software. TCO reflects not just the direct qualities of a software product (price, functionality, reliability) but also the relationship of the software to the organization's broader set of technology platforms, installed systems, skills and strategic goals, as well as available market and community based services."

15. TCO is a useful and indeed operational concept. But it does not directly address the full picture of how useful or effective any given software may be – that is the benefits that need to be set against costs. Its primary role is to allow comparisons between various software products and services through like-termed costs. TCO is, in its purest form, mute as to relative benefits. For this reason Return on Investment (ROI) is often used in conjunction with TCO studies to provide the benefit versus costs perspective.

16. Were we simply to assume that the cost of open source software is zero (or close to) and that in other respects it was no different to any other genre of software, then the task of TCO analysis would be easy[18].

17. However this is not the case. In many relevant respects open source software is different to proprietary software (both parties would probably agree on that). It places different demands on, and offers different benefits to, the host organisation and it is embedded in somewhat different software ecosystems and is served by different supply chains[19]. The question of TCO thus becomes less focused on what software costs *per se* to purchase or over its lifetime. TCO has to consider questions of how software fits into the organisation and relates to the other resources including legacy systems, technology platforms and infrastructures, skill sets and management style, as well as business strategy [Figure 2].

18. We thus can see TCO as essentially addressing three domains from which costs derive. First is the characteristics of the software itself, the second the resources of the organisation, and third the environment (software eco-system) they work within.

19. In the data collection for this study we adopt a targeted version of the software lifecycle (Table 1), seen from a software perspective and based on prior work by Deloitte and OpenForum Europe [see Appendix E]. In this model we identify 14 specific cost drivers within the five broad life cycle stages – Selection, Acquisition, Integration, Use, Retirement.

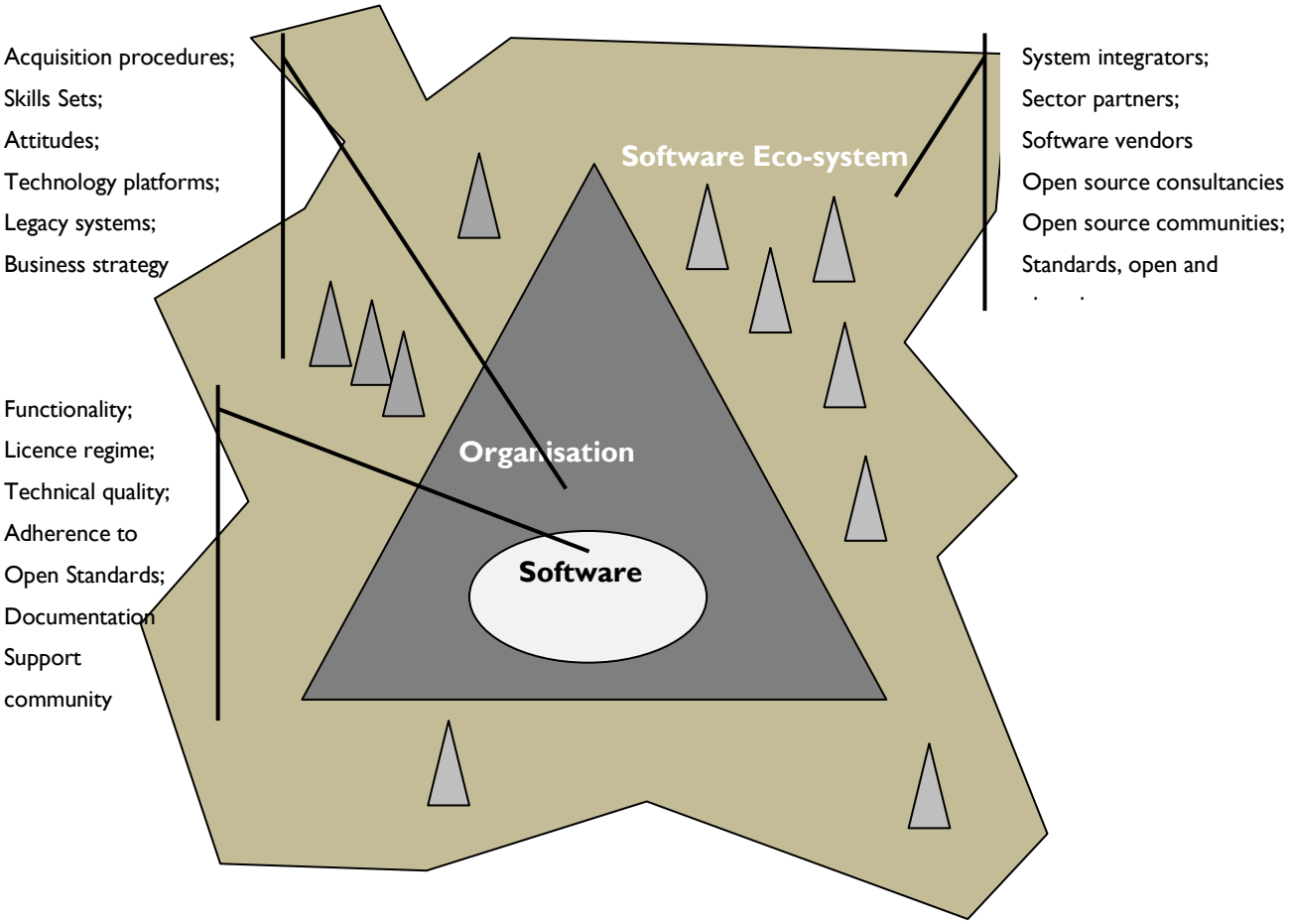

Figure 2. TCO as three elements

20. The value of the TCO model is that it allows some disaggregation of cost components. In doing so it allows consideration of a number of dimensions that influence choice and to some degree explicit cost. In this study we identified the set of 12 such strategic factors for open source adoption given in Table 2 below. These are expressed in general in a positive way and in the pro-forma (discussed below) we asked experienced people who had made such decisions or worked with open source software to identify and rank these factors in terms of importance.

Cost Category	
Search	Cost of up-front evaluation study
	Cost of up-front proof of concept implementation
Acquisition	Cost of Software
	Cost of Customisation for business needs
	Cost of Integration (to current platform)
Integration	Cost of Migration (data and users)
	Cost of Training
	Cost of Process and Best Practice change
Use	Cost of Support services - in house
	Cost of Support services – contracted
	Cost of Maintenance and Upgrades
	Software scaling (for change in user or transaction volumes)
Retire	Exit costs (in relation to hardware and software)
	Exit costs (in relation to changeover, re-training)

Table 1. 14 Cost Drivers for software acquisition and TCO

Factor of Influence
Reducing Vendor Lock-In
Ability to Experiment or Innovate
Value for Money
Access to Knowledge and Skills
Building Business Agility
Support for Incremental Development of Solutions
Ability to build and work with a peer community to re-use and share code
Ability to work with Local/SME Service Providers
Access to a wider choice of Support Service Providers
Ability to work with Sector Peers on Common Areas of Interest
Better adoption of Open Standards
Access to Code (e.g. for worst case)
Ability to Modify Code (e.g. for customization and solving critical defects)
Ability to change Support Service Providers

Table 2. 14 Factors of influence for open source software adoption

Methodology

21. This study was structured in two phases. Phase I was based around a pro-forma data collection instrument that was completed and returned by 32 people (see Appendix C). The pro-forma was set up for access as an electronic and printable document (available in odt, pdf and doc formats) and an online version in SurveyMonkey. The pro-forma was online for a period of two months. We received twenty-five responses online plus seven returned by email.

22. Phase 2 was concerned with in-depth interviews with 20 people in 14 organisations. In both phases a mix of public sector and private sector organisations were included, with the in-depth studies (phase 2) involving a preponderance of public bodies.

Validating the TCO model

23. Of the total pro-formas for Phase I we received about half were filled in by small to medium sized (SME) private companies (44%). Just under a quarter were completed by employees of large, and in some cases global companies. Public sector replies made up about one third. In phase II we covered a larger portion of the public sector respondents, but at this stage we were interested in capturing the generic open source experience.

24. For a number of questions respondents were asked to add comments or spell out the category of 'other' in more detail. Responses to such questions produced refinements of the initial model and became part of the interview guide and informed Phase II.

25. The respondents were purposively sampled, chosen for their extensive experience of OS products. The majority had experience with OS applications (a broad category), and about half with infrastructure focused systems. At a finer level of detail we see that OS platforms, web services, database and networking are the top four categories reported (Figure 3).

26. Over half of the respondents gave examples of situations where OSS had replaced proprietary software. And there was strong agreement that these uses of OSS had saved the organisation money.

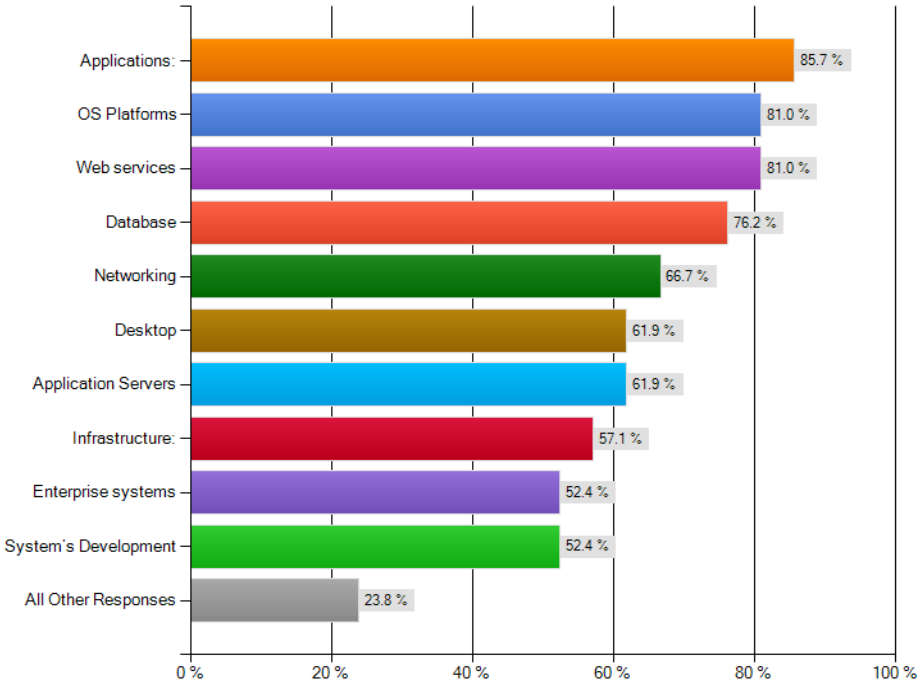

Figure 3: Principal Open Source software implemented in the organisation

Findings

27. Among the respondents just over half of their organisations had a policy in respect of open source. Comments added to this question described policies ranging from mild endorsement to strong presupposition – see Table 3.

Does your organisation have a policy in support for open source??
Internally always has to be used except when there is no alternative. To our customers has to be delivered always when enough tested internally previously.
It is always considered first, and proprietary software only purchased when no other option exists.
No formal policy but attempt to purchase the best solution, which often happens to be open source, based on various criteria and existing experience.
Unwritten policy -- we prefer OSS when sensible
We only use open source software, unless we need an alternate to communicate/ collaborate with clients and partners
We prefer open source software whenever choosing so doesn't directly reduce our efficiency compared to proprietary.
Use whenever reasonably possible. Even when usage means a bit of re-training and technical effort.

Table 3: Comments on OS Policy

28. Respondents spilt almost 50:50 as to whether they had substantial contracted support for open source or not. From the comments made it is clear that many see this as essentially an in-house task, and some saw it as a part of their wider commitment to the OS community. See Table 4.

Do you have substantial contracted support? If so is it from your usual systems integrator/partner?
One specific to open source (multiple answers)
Systems Partner
Our support is provided in house by a team of system administrators
Both, some support through our hosting/infrastructure partners, some through solution/product-specific partners.
We are a consultancy and we provide our own support except for few open source services that are subcontracted.
We don't, but central IT does. They contract with Red Hat for RH support. We have worked with vendors who support software like Sakai, Moodle, and Drupal, with mixed success.
We have an internal team separated from our corporate IT department
We are self-supporting with OSS. I and others participate in the OSS community.

Table 4: Comments on OS support practices

29. We also asked respondents, "Is software choice usually made by some formal assessment method or TCO study?" Of 18 respondents who answered this question 13 said "No". Comments accompanying this question often emphasised that 'it depended' or that decision making was addressed through the invitation to tender criteria rather than any separate TCO analysis.

30. When asked about strategic drivers for open source adoption the highest score was for *reduced vendor lock in* – echoing the origins of open source in the open source initial definition of 1998. A close second was *value for money*. Figure 4 gives a summary view, table 8 the full figures.

★★★★★★★★★★

The highest score for strategic drivers was for reduced vendor lock in. A close second was value for money

★★★★★★★★★★

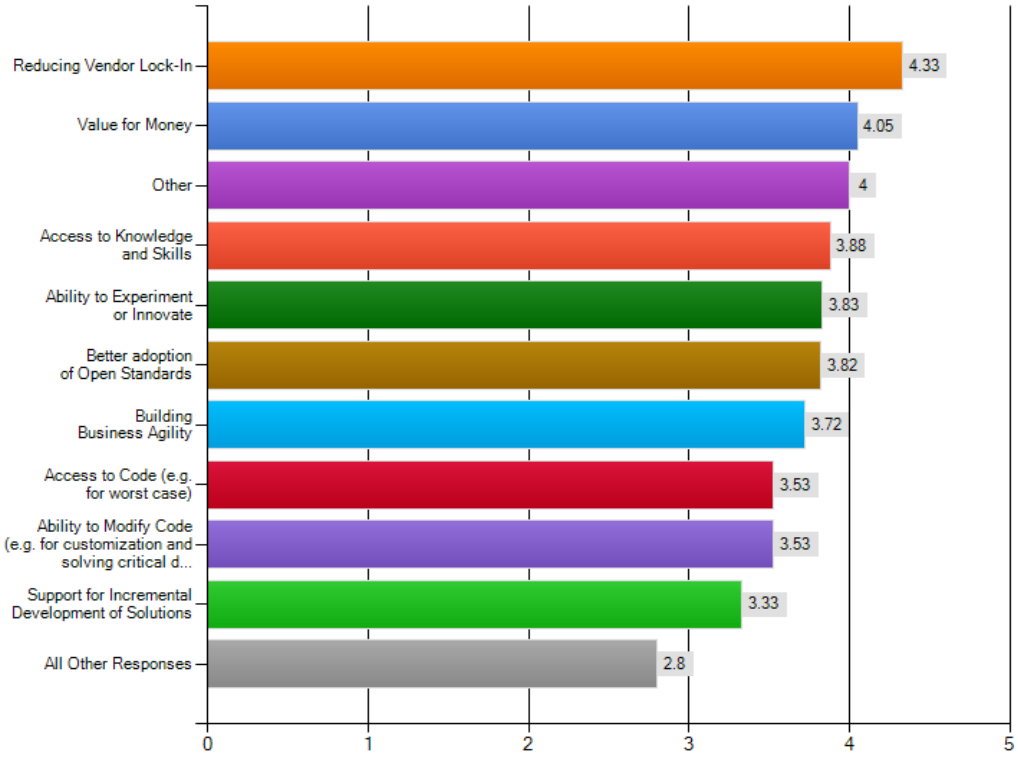

Figure 4: Summary view of Significant Strategic Drivers

31. Respondents were also positive in identifying relevant examples of these factors at work from within their own organisations. As Figure 5 shows, once again *reduced vendor lock in* was the most common, but among the top 5 are *open standards support*, *business agility* and *support for incremental development*.

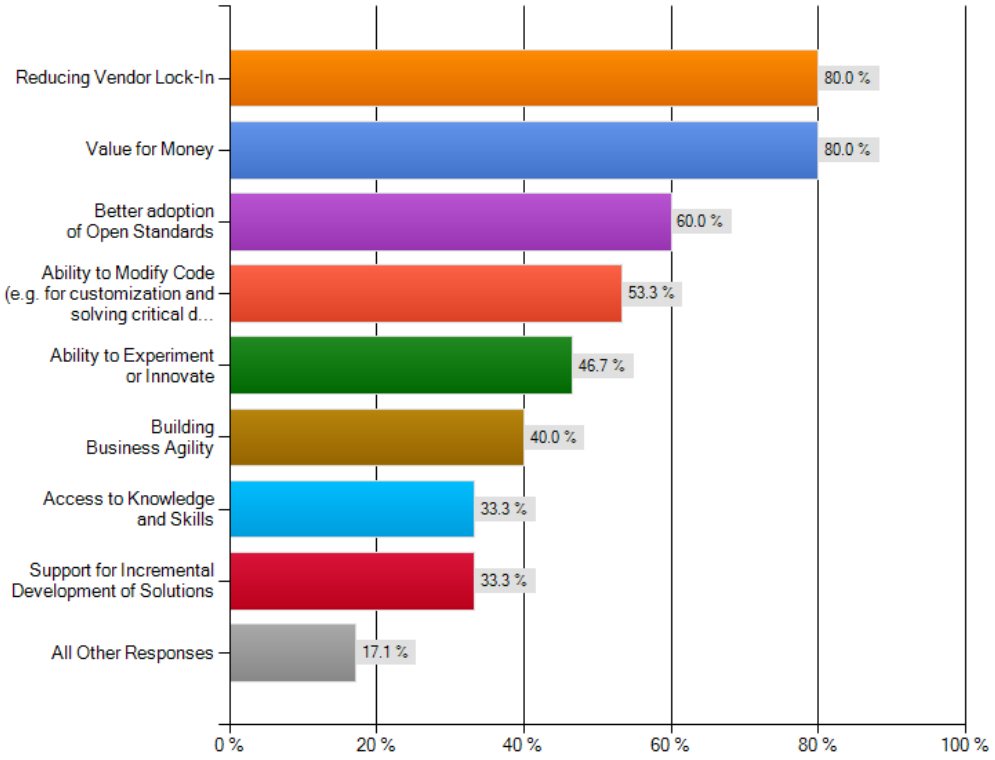

Figure 5: Relevant examples of strategic drivers among respondents

32. Respondents were asked in a free text response what they considered to be the primary factors holding back their organisation from using more open source software. A summary of their responses is given in Table 5 below.

What is holding your organisation back from using open source?
OS related issues
Understanding Licences and license compliance
Availability of specific apps
Some OSS is very immature, inferior user interfaces
Sometimes proprietary alternatives are simply better
Feature completeness
[Lack of a] community backing the open source project
Product related issues
Poor coverage in ERP arena ; Lack of availability of open source software for our industry
Incomplete implementations; Not working correctly
Very complex code bases (and communities)
Organisation related issues
Procurement policy
Value to money
Misinformation among upper level management; Lack of knowledge of key technical

	decision makers; Time availability
Support issues	
	Lack of in-house support; lack of in-house knowledge;
	Understanding by staff; Poor support of open standards by our business partners; Support worries; Requirements for external support contracts
Environment issues	
	Desire to have specific software; SAP Legacy
	Compatibility with Microsoft proprietary file formats
	Perceived Lack of acceptance of OSS for Public sector solutions
	Proprietary standards used by environment (govt & clients)

Table 5: Summary of comments on factors hindering adoption of open source.

33. When considering the component parts of the TCO model our respondents recognised a range of important cost categories and as expected no single one stood out. As Figure 6 shows the top six covered various aspects of the life cycle of a system (full data is given in Appendix B, Table 8). Slightly to our surprise *cost of software* was number one. This suggests that software license payments, particularly 'per seat' payments, are recognised as being significant – see also *software scaling* the number five cost identified. *Cost of software* is followed closely by *cost of customization for specific business need*, suggesting that organisations do indeed appreciate the availability of code and the use of open standards and scripting possibilities, as a way to build tailored applications.

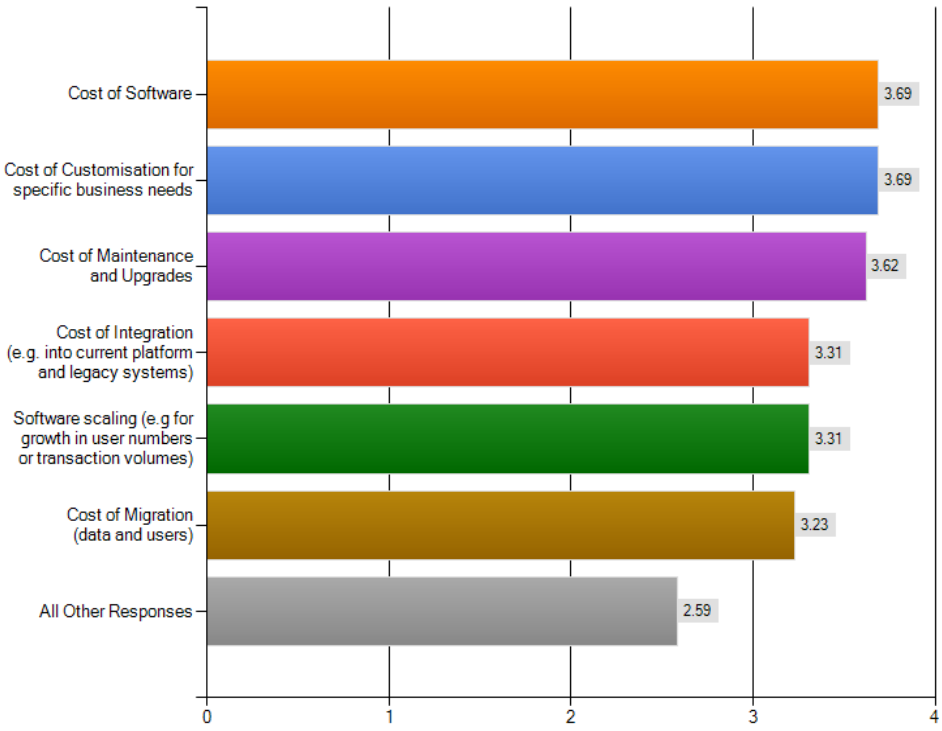

Figure 6: The top 6 cost categories for TCO

34. When considering the ease or difficulty of estimating these cost factors respondents were equally clear. As Figure 7 shows (full details in Appendix B, Table 10), once again *cost of software* was judged the easiest to estimate with cost *of maintenance and upgrades* as number two and *cost of support-contracted* as number three. The lowest categories were in the domain of *organisational change* and *exit costs* and establishing *in-house services* including *customisation.*

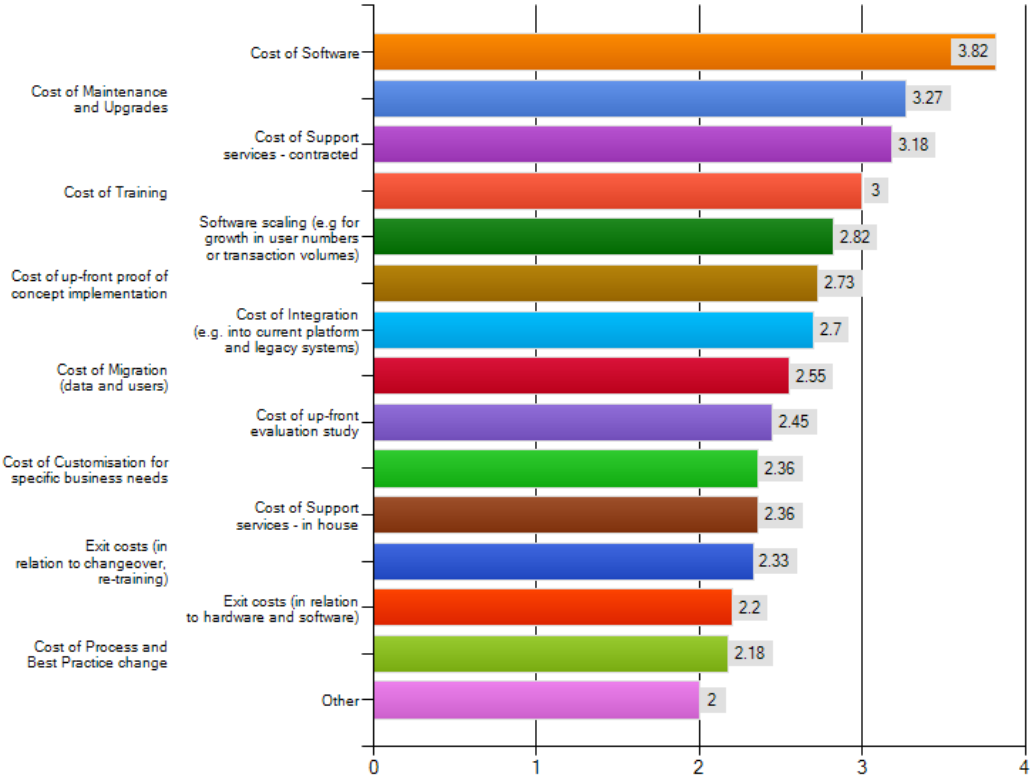

Figure 7: Ease of estimation of cost categories

Case Studies and In-depth interviews

35. The second phase of this study was qualitative, and included analysis of a number of cases of open source adoption by public sector bodies. We conducted in-depth semi-structured interviews lasting an hour or more with 20 respondents (see Table 6). The interviewees were each identified as key informants engaged with open source within their organisation.

36. The material from these interviews was analyzed for the main lessons, decisions, challenges, strengths, advice, best practices, consequences and other elements. The elements covered included cost of adoption, business benefits, key challenges and best practices as well as strategies employed. Drawing on Figure 2 and its framing of TCO as relating to three elements, software itself, the organisation and the software eco-system, we similarly structured our

analysis. In presenting these findings we use the 5 stage life cycle model discussed above above – Selection, Acquisition, Integration, Use and Retirement.

Interviewees		
Organisation	**Country**	**Date**
Opentia	Spain	27/04/2011
Solihull Council	UK	28/04/2011
Ars Aperta	France	29/04/2011
Socitm	UK	04/05/2011
Fedict	Belgium	05/05/2011
Ars Aperta	France	08/05/2011
Connectathon	UK	09/05/2011
TfL/Oyster	UK	11/05/2011
Value Decision	UK	12/05/2011
Cenatic	Spain	13/05/2011
Schoten Govt	Belgium	13/05/2011
Brazilian Govt	Brazil	19/05/2011
Munich City Govt	Germany	20/05/2011
Freelance consultant	Italy	23/05/2011
Andalucía Govt	Spain	30/05/2011
Andalucía Govt	Spain	09/06/2011
Andalucía Govt	Spain	09/06/2011
Camden Council	UK	20/07/2011

Table 6: Key informants for phase 2

37. Table 7 gives an overview of the analysis developed from our interview data on TCO issues, based on the software life cycle.

Selection	Acquisition	Integration	Use	Retirement
Formal TCO assessment	Initial purchase price	Interoperability	Learning	Data migration costs
Open source policy	Monetary costs of set-up	Customization expense	Training	Re-training
In-house and contracted expertise	Access to upgrades	Software scaling cost	Cost of evaluating software (tinkering)	Switching costs
Assessment and broader decision practices			Support services	

Table 7: SAIUR applied to OSS

Software issues

:- Selection

38. Making a selection from a choice of software products is seldom straightforward but with the added variety of open source products it becomes more complex. Open source software, beyond the well known offerings, is often not well marketed and it takes time and effort to uncover what is available and its provenance. Decision makers need to understand the various risks and opportunities that, for example, co-development or community involvement strategies can bring – and to some degree balance them in cost terms. These issues reflect distinctive qualities and opportunities that are part of the open source world but not necessarily as relevant for proprietary options.

★★★★★★★★★

A customer needs to match their system need and its strategic significance to a software's origins

★★★★★★★★★

39. The projects behind individual examples of open source software range from small groups developing simple applications through larger communities with more varied interests represented, and finally to projects with substantial company support and well aligned service offerings from

established providers. In selecting software a customer needs to match their system needs, its strategic significance, and available skills to the software's origins. For example a customer with extensive in-house expertise might take on more responsibility for support and customisation, while a small stand alone application may be suitable for a minimally supported but mature product.

40. Reducing dependency on in-place software or service providers, so called vendor lock-in, is a major concern for companies and the public sector bodies. In this respect open source is seen as offering a different relationship to the source of supply, one that allows far more freedom of movement. In the ultimate, source code is available and open to read and amend, and stemming from this openness alternative service arrangements can conceived of.

41. Open source software can allow an organisation to manage risk during the selection process in that ideas can be tested, others experiences can be freely borrowed, and since software direct costs are low and standards compliance generally high, mock-ups and prototypes can be freely built. Thus procurement decisions do not have to be made in one action, as is often the case with proprietary software, but can be 'worked up to'.

42. Evidence and opinion would seem to concur that desktops are perhaps the riskiest open source projects with the most stakeholders to consider. Selection in this domain is to be approached with care. At the least such a migration needs planning that moves servers and other applications over first and actual desktops last, and that in a very gradual manner with sympathetic training for the users, incentives and senior manager support.

★★★★★★★★★★

Desktops are perhaps the riskiest open source projects with the most stakeholders to consider

★★★★★★★★★★

43. One issue identified in a number of contexts is the perceptions of security risks with open source, in particular given open and available code. This is a debate that both academics and practitioners have engaged with – is open code be more or less secure than closed proprietary code? There are many factors that might give an answer in any particular context of use, but it is evidence of a proof of faith when governments (such as the City of Schoten) encourage the use of open source software in the development of e-ID systems for their citizens.

44. Our respondents understood that an open source choice could complicate the question of liability – for example, who can you sue for infringement of third party rights? This leaves the question of how risk should be assessed in such a case, and the extent to which system integrators or other contracted suppliers could or would carry such risks? In many cases this is an exaggerated concern, which suggests that education and a developed understanding is needed.

:- *Acquisition*

45. Confirming the finding reported for phase 1 of this study, interviewees were clear that real cost savings are possible from adoption of open source software - license fees and other acquisition costs are close to zero. More significantly perhaps, the upgrade costs (re-acquisition) are equally low and upgrade 'surprises' less common.

★★★★★★★★★★

Poor documentation is one serious concern reported

★★★★★★★★★★

46. Poor documentation is one serious concern reported with open source software. This was seen as particularly problematic because the open source supply chain was seen to underestimate the need for documentation targeted for different stakeholders. Documentation, as it is produced within open source communities, is seen essentially as a necessary part of the archive of expertise held on the code, not a user/adopter resource, and this can create a form of reverse lock in, or lock-out from open source.

Government of Andalusia (Spain)

The regional government of Junta de Andalucía has been involved with OSS since 2005. Its 250, 000 employees have all been involved with open source at some level. Interest in OSS in Andalusia can be traced even further back to OSS use in public schools and government supported Internet access centres.

Junta de Andalucía now has a large repository of software and encourages exchange of advice and experience within local authorities. It also has its own GNU/Linux distribution called Guadalinex.

According to Junta de Andalucía open source has saved them millions of Euros. But it has also brought other benefits. The benefits of the repository include; reduced duplication of effort; local private and public authorities can reuse and build on existing software; encouraging other expressions of openness; competition in the software market. They also see this initiative as providing local companies with an opportunity to become more competitive and provide better support through their detailed knowledge of open source software and development processes, and thus nurture the local IT software sector.

:- *Integration*

47. In public sector bodies open source software often has to find its place alongside the proprietary code of legacy systems and infrastructures. It is no easy task to move away from

software that has been used for years and which may interface with many other parts of the organisation and beyond to partner organisations. Interfaces to systems are thus another form of lock-in that can be created when software is used that is not based on open standards, or more charitably may pre-date them.

★★★★★★★★★★

Adherence to open standards is seen to reduce lock-in, allow for innovation opportunities, support a more agile development process, and provides a safeguard for the sustainability of code

★★★★★★★★★★

48. Some UK local authorities explained how open source supported agility. They saw the response time for queries and bug fixes as astounding compared to working with proprietary software companies who worked to their own cycles of product development.

49. Our respondents also suggest that a 'deeper' form of support is at times possible with open source software as compared to proprietary products. Empowered by access to the code base more can be done; some problems can be quickly and directly resolved, and developers are free to innovate and train with the code, and if appropriate fork it.

50. However, the idea of deep support is not accepted by all. Some question the availability, type and reliability of the support services that are offered for open source products. There are, of course, a number of open source products where we see a small industry established to develop and support the code and its users. But this is not true of all open source software, and in the public sector, with rather distinct needs for applications, such levels of support may be less easy to sustain.

51. It is then important to think through the product that will be adopted and the support service it can be expected to have. If using a niche open source product where there is only one plausible support offering, then you could be almost as locked in as you would be with a proprietary product.

:- *Use*

52. Many interviewees explained that open source cost savings materialize mid to long term rather than in the short-term. Thus, they report, it is important to manage expectations to ensure that a project that chooses open source software is not considered a failure prematurely if it does not deliver excellent service at substantially lower cost on day one.

53. As with any other software open source applications can incur excessive costs if implementation is badly planned, and expectations of short-term efficiencies and returns are

misconceived. As a number of people reported, acceptance by employees, and so training and education, becomes a significant factor. Similarly the involvement of a wide range of users from the earlies stages can help to communicate the reasons for an open source choice.

54. There are many examples that show that a change to open source software can face a high level of resistance by the organisation's own staff because of a fear of deskilling or gaining non-transferable skills. Thus the concerns of employees when being asked to change to open source software need to be directly addressed.

55. Training on the job can also help to build up local expertise with the full range of IT professionals and engaged users. However, examples from the public sector indicate that employees with substantial open source expertise may be quickly swallowed up by the private sector (where larger salaries are offered) or are promoted away from technical roles. This can leave the open source project without a champion and leader and can severely hobble a potentially successful project.

★★★★★★★★★★

A change to open source software can face a high level of resistance from the organisations own staff because of a fear of deskilling or gaining non-transferable skills

★★★★★★★★★★

:- Retirement

56. The final phase of the software life cycle is that of retirement. Retirement costs include all the expenses of switching from one software to another, various interoperability expenses, costs related to collateral legacy systems, retraining staff and teething issues with successor systems. This is an area where costs are not specific to open source. As would also be the case with proprietary software, such costs are seldom clear – certainly not at the time of acquisition. Nonetheless, to the extent that open source code is based on open standards, and that open source enterprise systems are usually well modularized, so the process of migration could be expected to be easier.

57. When migrating between open source products it is understood that costs may be lower because adherence to open standards allows greater interoperability. The organizations contributing to this study who had a more long term view of their open source software adoption gave comments to the effect that the migration costs (exit costs) were more favourable for open source and so this had for them become one of the deciding factors in favour of OSS.

★★★★★★★★★★

Open source can help achieve strategic independence, for example becoming less vulnerable to forced upgrades that are not useful but still very costly execute.

★★★★★★★★★★

Organization

58. One unsurprising but perhaps disquieting finding from interviews is that open source and its support service providers are used on occasion as a tactical device by both the public sector and private companies to manoeuvre a better deal from their current proprietary supplier. There may be no intention to change to open source but open source suppliers spend energy and resources on such customers. This is perhaps indicative of the half-way attitude of such organisations – recognising open source as a serious enough option to influence incumbent suppliers, but not serious enough to ever be taken up.

59. Respondents suggested that when an organisation is tied into a support contract then upgrades can place vendors in a position to exploit their power. This, they suggest, is far less possible with open source software.

60. A facet that is seen as beneficial but rather unexpected is how a culture of innovation and more risk taking behaviour can be promoted as open source is used. Open source adoption has, for example, forced local authorities to become more accepting of 'mistakes' that can be identified and rectified quickly by hands-on access to code and configurations. Experience of such agility and empowerment can spur the change in favour of open source.

★★★★★★★★★★

61. One respondent suggested that public sector employees can be loathe to do anything out of the ordinary as they fear a loss of their job and reputation. (A contemporary version of the 'nobody lost their job by buying IBM' of a generation ago, though these days IBM is a player in the open source world!) The continued use of Microsoft products, for example, provides employees of the public sector (and private) with a clear limit to their responsibility – they call MS for all help – and if things fail they can blame MS. In contrast, if *you* promoted the use of open source then *you* are directly responsible for its failures.

A culture of innovation and more risk taking behaviour can develop in the public sector as open source is used

★★★★★★★★★★

Transport for London/Oyster Card (TfL) (UK)

TfL moved to open source use in early 2007 in large part to avail itself of benefits to business agility, its' ability to innovate, and to take increased control of its software and support services. In collaboration with the consultants Deloitte, TfL began to use well established open source software such as Apache for its web server, JBoss and Red Hat Enterprise Linux (RHEL). In the years since their use of open source has deepened and widened as has their in-house expertise in the area. They have also come to play a part in some relevant open source communities.

TfL reports an 80% operational saving and staff efficiencies through this work, and this has contributed to meeting substantial budgetary cost reduction targets over ten years - £20 million achieved so far. Part of this has been achieved through greater transparency and audit-ability which allows more informed decisions to be made about systems over time. The example that TfL/Oyster offers is not essentially that open source has benefits for direct technology costs but that it can lead to more subtle benefits over time which keep cost reduction benefits flowing.

62. Put another way, most organizations we spoke to manage their IT resources more or less on the basis of pragmatism. If something is good enough and not broken then it will continue to be used. Open source alternatives have to argue their way past this understandable inertia. This requires imagination and forward thinking as well as clear messages from the open source world. It requires that the value proposition made by open source is clearly stated including, for example, the basis of licensing, liability, and support structures and commitments. In the public sector in the UK we found that while local authorities may begin to gravitate towards open source because of promised lower costs, this is not in the end enough and a better understanding of the organisational consequences (both benefits and organisational changes) of open source is needed.

Software Eco-System

63. Open source is driven by the pooling of resources, expertise and especially code for sharing, improvement, customization, and novel use. The Brazilian public sector's open platform and repository showcases how the use and creation of open source software can lead to innovation, savings on costs and expenses, and bring the citizens closer to the government providing a more democratic channel of communication. Everybody in theory is able to contribute to the archive, and many citizens have found ways other than software code to contribute and become a part of the platform co-creation.

64. Unsurprisingly, most of our interviewees were clear that open source adoption does overall help reduce costs. But some went beyond simply stating that open source saves money; they

believe that open source is also more able in the long run to be able to provide the types of systems and services that the public sector needs, and in a way that reflects the common interests of the sector.

65. From a more macro perspective it is argued by some that, as in the case of Spain and Brazil, open source can nurture and build up strength in local supplier companies and support economic growth. The software industry in both nations, it is argued, has flourished with open source development, in particular with the development of a larger base of SMEs. Similar ideas are heard from UK local authorities who see the link between their systems strategies and local economic development. The UK does not, however, restrict international companies and major players in the industry from bidding for the same tenders, thus SME's in the UK with an interest or commitment to open source find themselves in competition with well-resourced international companies. The former are often out-bid which leads to some suggesting that there is work to be done to promote the local and SME based IT industry in the UK.

66. This is reinforced by the more general observation of our public sector respondents that while it is easy enough in theory to take code and customize it to individual needs, this not so easy in practice. To do so they need to hire experts and look for support outside their organisation. As discussed above, open source adoption is often a mix of ambitions including control of code and configuration, and equally about taking more direct control of systems and their management. Thus the support desired is less the 'outsourcing' as practiced in the current Systems Integrator marketplace, but a more targeted and technically focused support that might be delivered well by smaller and perhaps more local suppliers.

67. A related problem identified by the respondents is that public bodies (as also private companies it must be said) seldom contribute code back to the open source community. This is usually not perceived as a major problem in open source, since even simple on-line download 'users' contribute by building critical mass and awareness, acting as de facto testers and general feedback providers. However, the attitude of taking and not giving back can upset some open source developer communities, and probably does not ensure the best synergy between code and practice.

68. Open source can, however, be a vehicle for community building. The City of Schoten, for example, promotes the 'principle of mutuality'. This is expressed as a culture of knowledge sharing across different local authorities based on a sharing of investment costs that helpes to create the knowledge embodied in software.

69. Indeed, the real value found in open source adoption, the real step change in total cost of ownership, may be found in the collaborative and co-creative process and spirit it encourages. This in turn leads to value creation, innovation, and a stronger ecosystem[20].

Lessons for the Public Sector

70. In this final section we highlight some of the important lessons learned by different organisations through their experience with open source adoption, development and use.

71. Some core themes are apparent across the cases we explored. First and perhaps most importantly, each and every interviewee repeated to us that pragmatism needs to guide open source adoption and not ideology. This is perhaps surprising given the 'ideological' dimension to much debate within open source communities, but users in business or in the public sector want for the most part to look beyond this.

72. This pragmatism is reflected in the wide understanding of the need for a strategy and a plan to work towards open source options. It is not a quick fix, must-do, low hanging fruit or an easy win. Any organisation interested in adopting open source software must not foster false hopes, and then suffer premature feelings of failure.

73. More positively, adoption of open source can be part of building a more agile organisation able to innovate and respond to change, and at the same time (re)building in-house expertise.

74. Migrating to open source is more likely to be successful if it is done when there is a real and present need for change, rather than simply on the basis of finding open source attractive on infrastructure cost arguments.

75. Organisations need to be clear on how 'benefit realization' will occur and how it will be monitored – will all parties realize (work for, appreciate and understand) the benefits and when should they be expected? These questions need a clear and honest answer so as not to nurture false hopes and unrealistic expectations.

76. In many respects adoption of open source is no different to any other change programme. It needs strong champions, wide commitment, planning and follow through. And like other areas of innovation it can be risky, but also rewarding.

77. The second overarching theme that emerges from the case studies is that open source is not just or only or always about 'cheap'. But it can bring a number of distinct and enduring benefits when contrasted to strategies based around proprietary software.

78. Respondents often emphasised that if open source is to become an accepted part of information systems work within the public sector then it needs government-level policies to sustain such a change including an overhaul of procurement processes. Most see this as an essential levelling of the playing field that can provide open source service companies, particularly in the SME sector, with a chance of survival and growth.

79. The wider policy regime and how it is communicated is important too. Change in government agencies is seldom driven by 'bottom-up' efforts, particularly if risk to projects or to careers is perceived. Strong policy from above provides support to such efforts and legitimises change. A similar concern to legitimise and even promote open source code is expressed by those who identify a need for some form of certification or kite marking, both to provide greater legitimacy to products and help purchaser make decisions and choices, but also to help focus expertise and know-how and establish common benchmarks.

80. The final theme we identify is that of collaboration and community. Adoption and development of open source can support the sharing of both expertise and expense between government bodies, for example among local authorities. Just as the business sector at times uses open source models as a flexible and efficient way to work with rivals on mutually beneficial projects, so for public bodies it can serve as a flexible route to collaboration.

81. A number of our respondents were keen to emphasise that organisations using open source software products do need to contribute to the relevant community and the code base. Ideally, this becomes a contribution of some substance, though this may be only able to occur when a substantial part of the organisation becomes users of open source software, and the staff costs associated with open source activism can be seen as proportionate.

References

[1] **G. Archer** 2010. "Open Source Software Policy: Australian Government Information Management Office (AGIMO) Circular." Australian Government: Department of Finance and Deregulation http://fwd4.me/wY6

[2] **V. Kundra, D. Gordon and V. A. Espinel**. 2011. "Technology Neutrality: Memorandum for Chief Information Officers and Senior Procurement Executives." Washington DC Executive Office of the President: Office of Management and Budget. http://fwd4R.A. .me/wY5

[3] **R. A. Ghosh, R. Glott, P.-E. Schmitz and A. Boujraf**. 2010. "Guideline on public procurement of Open Source Software." Brussels, Belgium: UN University/MERIT and Unisys Belgium. http://fwd4.me/wXx

[4] **R. Burkhardt**, "Seven Predictions for Open Source in 2009," http://drdobbs.com/open-source/212700284, Ed., 2008

[5] **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., 2011.

[6] **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., 2009.

[7] The interviewee from Cenatic in Spain quoted this in the interview we conducted for Phase II for our current study.

[8] **M. Ballard**, "Government IT suppliers claim procurement system excludes open source," in ComputerWeekly.com http://www.computerweekly.com/Articles/2011/02/25/245598/Government-IT-suppliers-claim-procurement-system-excludes-open.htm 2011.; **C. Saran**, "Government plans procurement overhaul to slash IT spend," in ComputerWeekly.com http://www.computerweekly.com/Articles/2010/09/24/242958/Government-plans-procurement-overhaul-to-slash-IT-spend.htm 2010.; **K. Hall**, "Government tells major IT suppliers - we want more open source software," in ComputerWeekly.com http://www.computerweekly.com/Articles/2011/02/23/245555/Government-tells-major-IT-suppliers-we-want-more-open-source.htm 2011.

[9] **C. Saran**, "Government plans procurement overhaul to slash IT spend," in ComputerWeekly.com http://www.computerweekly.com/Articles/2010/09/24/242958/Government-plans-procurement-overhaul-to-slash-IT-spend.htm 2010.; **R. A. Ghosh, R. Glott, P.-E. Schmitz, and A. Boujraf**, "Guideline on public procurement of Open Source Software," Brussels, Belgium: http://fwd4.me/wXx: UN University/MERIT and Unisys Belgium, 2010.; **M. Ballard**, "Government IT suppliers claim procurement system excludes open source," in ComputerWeekly.com http://www.computerweekly.com/Articles/2011/02/25/245598/Government-IT-suppliers-claim-procurement-system-excludes-open.htm 2011.; **C. Saran**, "Government plans procurement overhaul to slash IT spend," in ComputerWeekly.com http://www.computerweekly.com/Articles/2010/09/24/242958/Government-plans-procurement-overhaul-to-slash-IT-spend.htm 2010.; **K. Hall**, "Government tells major IT suppliers - we want more open source software," in ComputerWeekly.com http://www.computerweekly.com/Articles/2011/02/23/245555/Government-tells-major-IT-suppliers-we-want-more-

open-source.htm 2011.; **S. Phipps**, "Open Source Procurement: Subscriptions," in ComputerWorldUK http://blogs.computerworlduk.com/simon-says/2011/03/open-source-procurement-subscriptions/index.htm 2011.

[10] House of Commons Public Administration Select Committee Report Government and IT – "a recipe for rip-offs": time for a new approach. 18 July 2011. HC715- http://www.parliament.uk/business/committees/committees-a-z/commons-select/public-administration-select-committee/publications/

[11] **B. Russo and G. Succi**, "A Cost Model of Open Source Software Adoption," IJOSSP, pp. 60-82, 2009.

[12] **P. Agerfalk and B. Fitzgerald**, "Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy," MIS Quarterly, vol. 32, pp. 385-400, 2008.

[13] **R. Sutor**, "Managing open source adoption in your IT organization," http://www.sutor.com/newsite/blog-open/?p=3260 Ed., 2009.; **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.

[14] **Michael Skok -** 2011 5th Annual North Bridge Future of Open Source Survey
http://www.futureopensource.net/2011-future-open-source-survey

[15] http://www.osor.eu/news/de-launch-of-the-open-source-integration-initiative

http://www.osor.eu/news/de-updated-vendor-independent-e-id-application-based-on-open-source

http://www.osor.eu/news/es-open-source-software-widely-used-by-the-regional-government-of-extremadura/?searchterm=extremadura

http://www.osor.eu/news/using-free-software-improves-relation-between-state-and-society/?searchterm=brazil

http://www.osor.eu/news/misc/DenHaag26072010-v2.pdf/view?searchterm=andalucia

[16] **L. M. Ellram**, "A Taxonomy of Total Cost of Ownership Model," *Journal of Business Logistics,* vol. 15, pp. 171-191, 1994.; **L. M. Ellram**, "Total cost of ownership an analysis approach for purchasing. ," *Journal of Physical Distribution and Logistics* vol. 25, pp. 4-23, 1995**.; L. M. Ellram and S. P. Siferd**, "Total Cost of Ownership: A Key Concept in Strategic Cost Management Decisions," *Journal of Business Logistics,* vol. 19, pp. 55-84, 1998.; **and K. Hurkens, W. v. d. Valk, and F. Wynstra**, "Total Cost of Ownership in the Services Sector: A Case Study," *The Journal of Supply Chain Management,* vol. 42, pp. 27-37, 2006.

[17] **J. Lerner and M. Schankerman**, The Comingled Code: Open Source and Economic Development. Hong Kong: MIT Press, 2010.

[18] **B. Russo** and **G. Succi**, "A Cost Model of Open Source Software Adoption," *IJOSSP,* pp. 60-82, 2009.

[19] **A. MacCormack**, "Evaluating Total Cost of Ownership for Software Platforms: Comparing Apples, Oranges and Cucumbers," mimeo 2003.

[20] **J. Dinkelacker, P. Garg, R. Miller, and D. Nelson**, "Progressive Open Source," in Proceedings of the 2002 ACM International Conference on Software Engineering (ICSE'02), 2002, pp. 177-184.

Appendix A Check list of costs and benefits for software adoption

	Cost Category	Comment
Software Specific:		
Search	Cost of up-front evaluation study	
	Cost of up-front proof of concept implementation	
Acquisition	Cost of Software	
	Cost of Customisation for business needs	
	Cost of Integration (to current platform)	
Integration	Cost of Migration (data and users)	
	Cost of Training	
	Cost of Process and Best Practice change	
Use	Cost of Support services - in house	
	Cost of Support services - contracted	
	Cost of Maintenance and Upgrades	
	Software scaling (for change in user or transaction volumes)	
Retire	Exit costs (in relation to hardware and software)	
	Exit costs (in relation to changeover, re-training)	
Organisation Specific:		
Strategic lever	The use of a cheaper option to help create competition	
Dependence	Do future upgrades lock you into a particular vendor?	
Empowerment	Does the software encourage empowerment and ability to change software as needed?	
Innovation driver	Does it help to inspire and drive innovation because it is accessible to view and change?	

Software Eco-System Specific:		
Platform co-creation	Can the software be pooled, shared and built upon to create a platform which encourages reuse and co-creation?	
Collaborative competition	Will the adoption of this software help to nurture the local IT industry by levelling the playing field, and encourage collaborative competition?	
Building in-house expertise	Will the software empower the organisation and help develop in-house expertise through access to a knowledgeable community, source code, and an environment which implies sharing and reciprocity?	
Principle of mutuality	Can the use, adoption and development of this software create experts which can then be used as a shared resource across local authorities and central government?	

Appendix B Data Tables

	Low	Low to Medium	Medium	Medium to High	High	Rating Average
Reducing Vendor Lock-In	5.6% (1)	0.0% (0)	16.7% (3)	11.1% (2)	**66.7% (12)**	4.33 (18)
Ability to Experiment or Innovate	5.6% (1)	5.6% (1)	22.2% (4)	**33.3% (6)**	**33.3% (6)**	3.83 (18)
Value for Money	0.0% (0)	5.3% (1)	21.1% (4)	**36.8% (7)**	**36.8% (7)**	4.05 (19)
Access to Knowledge and Skills	0.0% (0)	0.0% (0)	35.3% (6)	**41.2% (7)**	23.5% (4)	3.88 (17)
Building Business Agility	5.6% (1)	5.6% (1)	22.2% (4)	**44.4% (8)**	22.2% (4)	3.72 (18)
Support for Incremental Development of Solutions	5.6% (1)	16.7% (3)	22.2% (4)	**50.0% (9)**	5.6% (1)	3.33 (18)
Ability to build and work with a peer community to re-use and share code	11.8% (2)	**29.4% (5)**	11.8% (2)	17.6% (3)	**29.4% (5)**	3.24 (17)
Ability to work with Local/SME Service Providers	23.5% (4)	23.5% (4)	**29.4% (5)**	23.5% (4)	0.0% (0)	2.53 (17)
Access to a wider choice of Support Service Providers	23.5% (4)	**35.3% (6)**	11.8% (2)	23.5% (4)	5.9% (1)	2.53 (17)
Ability to work with Sector Peers on Common Areas of Interest	**29.4% (5)**	23.5% (4)	17.6% (3)	23.5% (4)	5.9% (1)	2.53 (17)
Better adoption of Open Standards	11.8% (2)	5.9% (1)	17.6% (3)	17.6% (3)	**47.1% (8)**	3.82 (17)
Access to Code (e.g. for worst case)	11.8% (2)	0.0% (0)	29.4% (5)	**41.2% (7)**	17.6% (3)	3.53 (17)
Ability to Modify Code (e.g. for customization and solving critical defects)	5.9% (1)	0.0% (0)	**47.1% (8)**	29.4% (5)	17.6% (3)	3.53 (17)
Ability to change Support Service Providers	11.8% (2)	11.8% (2)	**35.3% (6)**	29.4% (5)	11.8% (2)	3.18 (17)
Other	0.0% (0)	0.0% (0)	25.0% (1)	**50.0% (2)**	25.0% (1)	4.00 (4)

Table 8. Strategic Drivers for Open Source Adoption

	Low	Low to Medium	Medium	Medium to High	High	Rating Average
Reducing Vendor Lock-In	5.6% (1)	0.0% (0)	16.7% (3)	11.1% (2)	**66.7% (12)**	4.33 (18)
Ability to Experiment or Innovate	5.6% (1)	5.6% (1)	22.2% (4)	**33.3% (6)**	**33.3% (6)**	3.83 (18)
Value for Money	0.0% (0)	5.3% (1)	21.1% (4)	**36.8% (7)**	**36.8% (7)**	4.05 (19)
Access to Knowledge and Skills	0.0% (0)	0.0% (0)	35.3% (6)	**41.2% (7)**	23.5% (4)	3.88 (17)
Building Business Agility	5.6% (1)	5.6% (1)	22.2% (4)	**44.4% (8)**	22.2% (4)	3.72 (18)
Support for Incremental Development of Solutions	5.6% (1)	16.7% (3)	22.2% (4)	**50.0% (9)**	5.6% (1)	3.33 (18)
Ability to build and work with a peer community to re-use and share code	11.8% (2)	**29.4% (5)**	11.8% (2)	17.6% (3)	**29.4% (5)**	3.24 (17)
Ability to work with Local/SME Service Providers	23.5% (4)	23.5% (4)	**29.4% (5)**	23.5% (4)	0.0% (0)	2.53 (17)
Access to a wider choice of Support Service Providers	23.5% (4)	**35.3% (6)**	11.8% (2)	23.5% (4)	5.9% (1)	2.53 (17)
Ability to work with Sector Peers on Common Areas of Interest	**29.4% (5)**	23.5% (4)	17.6% (3)	23.5% (4)	5.9% (1)	2.53 (17)
Better adoption of Open Standards	11.8% (2)	5.9% (1)	17.6% (3)	17.6% (3)	**47.1% (8)**	3.82 (17)
Access to Code (e.g. for worst case)	11.8% (2)	0.0% (0)	29.4% (5)	**41.2% (7)**	17.6% (3)	3.53 (17)
Ability to Modify Code (e.g. for customization and solving critical defects)	5.9% (1)	0.0% (0)	**47.1% (8)**	29.4% (5)	17.6% (3)	3.53 (17)
Ability to change Support Service Providers	11.8% (2)	11.8% (2)	**35.3% (6)**	29.4% (5)	11.8% (2)	3.18 (17)
Other	0.0% 0)	0.0% (0)	25.0% (1)	**50.0% (2)**	25.0% (1)	4.00 (4)

Table 9: Type of Cost: Importance

	Low	Low to Medium	Medium	Medium to High	High	Rating Average
Cost of up-front evaluation study	**30.8% (4)**	23.1% (3)	15.4% (2)	23.1% (3)	7.7% (1)	2.54 (13)
Cost of up-front proof of concept implementation	23.1% (3)	23.1% (3)	**30.8% (4)**	7.7% (1)	15.4% (2)	2.69 (13)
Cost of Software	7.7% (1)	15.4% (2)	7.7% (1)	**38.5% (5)**	30.8% (4)	3.69 (13)
Cost of Customisation for business needs	0.0% (0)	15.4% (2)	**30.8% (4)**	23.1% (3)	**30.8% (4)**	3.69 (13)
Cost of Integration (to current platform)	7.7% (1)	7.7% (1)	**38.5% (5)**	**38.5% (5)**	7.7% (1)	3.31 (13)
Cost of Migration (data and users)	0.0% (0)	23.1% (3)	**38.5% (5)**	30.8% (4)	7.7% (1)	3.23 (13)
Cost of Training	15.4% (2)	30.8% (4)	**46.2% (6)**	7.7% (1)	0.0% (0)	2.46 (13)
Cost of Process and Best Practice change	15.4% (2)	30.8% (4)	**38.5% (5)**	15.4% (2)	0.0% (0)	2.54 (13)
Cost of Support services – in house	15.4% (2)	**30.8% (4)**	**30.8% (4)**	23.1% (3)	0.0% (0)	2.62 (13)
Cost of Support services – contracted	23.1% (3)	15.4% (2)	**38.5% (5)**	23.1% (3)	0.0% (0)	2.62 (13)
Cost of Maintenance and Upgrades	15.4% (2)	7.7% (1)	7.7% (1)	**38.5% (5)**	30.8% (4)	3.62 (13)
Software scaling (for growth in users or transaction volumes)	15.4% (2)	0.0% (0)	**38.5% (5)**	30.8% (4)	15.4% (2)	3.31 (13)
Exit costs (in relation to hardware and software)	30.8% (4)	7.7% (1)	15.4% (2)	**38.5% (5)**	7.7% (1)	2.85 (13)
Exit costs (in relation to changeover, re-training)	**30.8% (4)**	23.1% (3)	23.1% (3)	23.1% (3)	0.0% (0)	2.38 (13)
Other	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.00 (0)

Table 10: Type of Cost – Ease of estimation

Appendix C Methodology

C. Methodology

This study was structured to take place in two phases. Phase I included a pro-forma of twenty-five questions. These questions covered the basics of the company size, name and focus, but then went on to ask some very detailed questions about the various applications that are open source, why they were chosen, if they replaced proprietary software, was any difference in cost experienced, and finally, what prompted this change or need to adopt open source software. The responses to the pro-forma, especially to the last question were fed into an interview guide. This then took us to phase II where we conducted between 17 in-depth interviews.

C.1 Phase I – Data Collection Pro-forma

Phase I involved the creation of the pro-forma which was based on literature and documentation that helped to understand TCO models used in companies. The pro-forma was set up for access in two ways, document form (available in odt, pdf and doc formats) and an online version set up in SurveyMonkey. The aim of this study was to make sense of adoption of open source software by both private companies and public sector organizations. Though funded by the UK Cabinet Office to assess and evaluate the costs and issues involved in open source adoption by government agencies we decided that a more sound methodology would involve a balanced mix of commercial and public sector organizations. 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[i] but there are far more 'success' stories of open source adoption by commercial companies[ii]. The factors that encourage private companies to adopt open source software, especially considering most business models of such adoption indicate that the software itself does not lead to value creation or capture directly[iii], make some of the lessons translatable across both sectors.

The pro-forma was put online for a period of two months (and is still online but for the sake of this report we only took into account the pro-formas completed in the first two months) and we received twenty-five responses. We also received seven paper based pro-formas sent back to us as scanned documents via email. This made a total of thirty-two pro-formas. We had set the pro-forma to ensure that details of the respondent was a required category. This was done to be able to filter out any responses that were biased, duplicate or simply not completed with any seriousness. Of the 25 online pro-formas two were filled in by people calling themselves 'test' and 'anonymous'. We discounted the results from both these pro-formas. We also had two incomplete pro-formas online. Incomplete pro-formas were those where some questions were

skipped. As this exercise was carried out as a precursor and data gathering exercise more for the interest of creating a strong and clear interview guide for phase II we accepted the results of the incomplete pro-formas. Phase II is where the researchers involved in this study gained a more detailed understanding of TCO models and the decision-making process in organizations so it was felt that so long as the pro-formas were recognized to be valid (not anonymous or biased) and useful (filled in 75% of the pro-forma and added some non-mandatory comments that helped us to evaluate the experience of the company with open source) we would include the results to help shape the interview guide for phase II.

C.2 Phase I Leading to Phase II

Of the total pro-formas we received the majority of them were filled in by small to medium sized private companies (44%). Small to medium sized enterprises included all those with a number of employees ranging between1-100. We had 24% of the pro-formas completed by employees of large, and in many cases global companies (employees ranging from 101 and above). Public sector replies made 32% of the total. In phase II we covered a larger portion of the public sector.

The pro-forma had a number of questions where respondents were asked to add comments or spell out the category of 'other' in more detail. Responses to such questions gave rise to some very interesting issues which became a part of the interview guide and informed the researchers involved. Key personnel of organisations were interviewed for Phase II. We chose them on the basis that they were heavily involved in making procurement decisions and strategy of open source use in the organization we chose to conduct our study in.

The main ideas the Phase I respondents focused on included the lack of maturity level of open source software, license confusions and lack of knowledge about the implications of various open source licenses. Other ideas which arose were somewhat more surprising, most organizations do not even attempt a TCO study before making procurement decisions because of the expense such studies involve. The models used to assess TCO are also more suited for proprietary software and companies are not comfortable or skilled to tweak them for open source. And lastly, there is no policy in most companies for open source adoption. These decisions are made more ad-hoc and usually based on pragmatic decisions of use and need rather than cost.

For Phase II we conducted in-depth semi-structured interviews, lasting an hour or more with each interviewee. The interviewees included key, and engaged members with open source in the various local authorities and counties. Strategy leaders in the same organizations were also interviewed. The material from the interviews, along with the pro-formas, was analyzed systematically for the main lessons, decisions, challenges, strengths, advice, best practices, consequences and other interesting elements that emerge from the interviews to help mould the framework we have built that can then guide public sector adoption of open source. The key elements covered include total cost of adoption but also reach beyond simple cost analysis to

also include benefits, key challenges and the best practices (coping strategies employed by the member organizations to ameliorate the situation).

---

[i] **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.; and **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.

[ii] **J. Dinkelacker, P. Garg, R. Miller, and D. Nelson**, "Progressive Open Source," Hewlett-Packard, Palo Alto, California HPL-2001-233, September 28th, 2001.; **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.; **B. Fitzgerald**, "The Transformation of Open Source Software," *MIS Quarterly,* vol. 30, pp. 587-598, September, 2006.; and **S. O'Mahony, F. C. Diaz, and E. Mamas**, "IBM and Eclipse (A)," *Harvard Business Review Case Study,* vol. December 16, 2005.

[iii] **J. West and S. Gallagher**, "Challenges of Open Innovation: The Paradox of Firm Investment in Open Source Software," *R&D Management,* vol. 36, pp. 315-328, 2006; **J. West**, "How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy,* vol. 32, pp. 1259-1285, 2003.; **A. Osterwalder, Y. Pigneur, and C. L. Tucci**, "Clarifying Business Models: Origins, Present, and Future of the Concept," *Communications of the Association for Information Systems,* vol. 15, pp. 1-40, 2005.; and **S. L. Vargo and R. F. Lusch**, "Evolving to a New Dominant Logic for Marketing," *Journal of Marketing* vol. 68, pp. 1-17, 2004.

Appendix D Pro-forma used for Phase I[xxiv]

<table>
<tr><th colspan="2">Section A: Organisation and Respondent Details</th></tr>
<tr><td>Organisation</td><td></td></tr>
<tr><td>Size and Sector of the Organisation including approximate number of employees</td><td></td></tr>
<tr><td>Name of Respondent</td><td></td></tr>
<tr><td>Current Position</td><td></td></tr>
<tr><td>Contact details:
phone

email</td><td></td></tr>
<tr><td>Your role(s) in relation to software procurement and/or open source software policy.</td><td></td></tr>
<tr><th colspan="2">Section B: Use of Open Source</th></tr>
<tr><td>Please briefly list principal open source software (OSS) that is implemented in the organisation with a rough time period (e.g. OpenOffice implemented 2008 and still in use; Eclipse in Java development since 2009):
Applications:
Enterprise systems
Vertical/line of business
Desktop
System's Development
Infrastructure:
OS Platforms
Application Servers
Web services
Networking
Database</td><td>
<table>
<tr><th>Name Software/Systems</th><th>In your view has this system saved the organisation money?</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Are there examples where OSS has directly replaced proprietary software? If yes, please give examples.</td><td></td></tr>
</table>

Does your organisation have a person or unit that takes specific responsibility for promoting or supporting open source software?	
Does your organisation have a policy in respect of open source software? *For example when it should always or never be considered.*	
Does your organisation have substantial contracted support for your open source software? *If yes, is it through your usually systems integrator/partner, or one specific to open source?*	
Is software choice usually made by some formal assessment method or TCO study. *If yes, can you explain when this is done and how useful it proves to be?*	

Section C: Strategic Drivers		
Please indicate your relative judgement as to the rank importance of the **strategic factors** shown below in influencing decisions to adopt open source software in your organisation. In the second column please indicate if there is a relevant example of the factor in your organisation.		
Factor of Influence	***Importance of Factor*** ***Low Medium High***	***Are there relevant examples in your organisation?***
Reducing Vendor Lock-In	*1 2 3 4 5*	*Yes { } No { }*
Ability to Experiment or Innovate	*1 2 3 4 5*	*Yes { } No { }*
Value for Money	*1 2 3 4 5*	*Yes { } No { }*
Access to Knowledge and Skills	*1 2 3 4 5*	*Yes { } No { }*
Building Business Agility	*1 2 3 4 5*	*Yes { } No { }*
Support for Incremental Development of Solutions	*1 2 3 4 5*	*Yes { } No { }*
Ability to build and work with a peer community to re-use and share code	*1 2 3 4 5*	*Yes { } No { }*
Ability to work with Local/SME Service Providers	*1 2 3 4 5*	*Yes { } No { }*
Access to a wider choice of Support Service Providers	*1 2 3 4 5*	*Yes { } No { }*
Ability to work with Sector Peers on Common Areas of Interest	*1 2 3 4 5*	*Yes { } No { }*
Better adoption of Open Standards	*1 2 3 4 5*	*Yes { } No { }*
Access to Code (e.g. for worst case)	*1 2 3 4 5*	*Yes { } No { }*
Ability to Modify Code (e.g. for customization and solving critical defects)	*1 2 3 4 5*	*Yes { } No { }*
Ability to change Support Service Providers	*1 2 3 4 5*	*Yes { } No { }*
Other *(explain)*	*1 2 3 4 5*	*Yes { } No { }*
Which <u>three</u> of the above strategic factors would you consider the most influential in driving open source into your organisation? *Very brief answers are sought here (a word, a phrase) to allow follow up in interview.*	1. 2. 3.	
Which <u>three</u> strategic factors would you consider the most significant in <u>holding your organisation back</u> from using open source software? *Very brief answers are sought here (a word, a phrase) to allow follow up in interview.*	1. 2. 3.	

Section D: Total Cost of Ownership Models		
Please indicate your relative judgement as to the rank importance of the **cost category** shown below when making software acquisition decisions and in particular when comparing open source software with proprietary solutions. E.g. which ones would you really want to focus upon? In the second column please indicate your relative judgement as to how easy it is to estimate these costs		
Type of Cost	***Importance of Cost Category*** *Low Medium High*	***Ease of estimation of Cost Category*** *Low Medium High*
Cost of up-front evaluation study	*1 2 3 4 5*	*1 2 3 4 5*
Cost of up-front proof of concept implementation	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Software	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Customisation for specific business needs	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Integration (e.g. into current platform and legacy systems)	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Migration (data and users)	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Training	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Process and Best Practice change	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Support services - in house	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Support services - contracted	*1 2 3 4 5*	*1 2 3 4 5*
Cost of Maintenance and Upgrades	*1 2 3 4 5*	*1 2 3 4 5*
Software scaling (e.g for growth in user numbers or transaction volumes)	*1 2 3 4 5*	*1 2 3 4 5*
Exit costs (in relation to hardware and software)	*1 2 3 4 5*	*1 2 3 4 5*
Exit costs (in relation to changeover, re-training)	*1 2 3 4 5*	*1 2 3 4 5*
Other *(explain)*	*1 2 3 4 5*	*1 2 3 4 5*
Which <u>three</u> of the above cost categories would you consider the hardest to quantify, and why? *Very brief answers are sought here (a word, a phrase) to allow follow up in interview.*	1. 2. 3.	
Do you have a specific acquisition decision in mind when you give these answers? *If so, we will be pleased to talk further about this specific case in interview*		

Section D: Your Questions and Other Comments	
Please add any comments, suggestions or points here that you believe to be important to consider when discussing software acquisition in relation to return on investment (ROI), TCO or related ideas.	

Thank you very much for completing this survey. We will be in touch with you to arrange an interview.

In our interview we hope to ask you a number of more detailed questions about the issues raised above, and to explore some other areas. As food for thought we indicate some of the topics we hope to ask you about.

- Has conducting a TCO study before migrating to open source prove useful? In what way?
- What has been the most surprising cost of using/migrating to open source, and why?
- Is evaluating the cost of open source software adoption made more or less complicated when it is used alongside proprietary software?
- Where, and in which areas, do you find that open source provides the greatest cost advantages.
- What would be your answer to the same question in relation to proprietary software?
- What aspects of open source do you find to be least supported, and how does this affect your TCO evaluations or acquisition decisions?
- How, if at all, do exit costs differ in relation to open source as opposed to proprietary software?

[xxiv] Online version is available at https://www.surveymonkey.com/s/RNZS7J5

Appendix E TCO Matrix created for OpenForum Europe by Deloitte

	Rank	Consideration	Description	Proprietary	Open Source with vendor support via subscription	Open Source without vendor support
Evaluation	L	Cost of evaluation the solution (paper based)	In arriving at the right solution option it is necessary to evaluate the various options. This is not typically considered part of the TCO but has been included for completeness. While the cost of the research for Open Source verses Proprietary will differ to an extent an important point to note is that Open Source options could be overlooked if traditional sources of information (e.g. analyst reports) are used exclusively. This is more likely to be the case if someone who does not have an IT background carries out the initial research.	Literature (analyst material, technical documentation, marketing material etc.) is generally more developed for the established proprietary and Open Source products. However, analyst material carries a cost where Open Source material is generally free and available online.		Paper based evaluation is harder to carry out and therefore more expensive.
	L	Cost of evaluation the solution (proof of concept)	Prior to making a decision on the solution it may be necessary or desirable to carry out a proof of concept. Alternatively if the solution decision is being driven by the IT department the first step of an evaluation may be to download the most promising looking Open Source software solution and "play around with it" to see if it is suitable. This can be considered as "adoption led selection" as opposed to "procurement led selection" and while this can lead to the best solution being selected it can equally led to a solution with a high TCO being adopted without due consideration every being given to lifetime cost.	High as licences typically need to be purchased to support the proof of concept. Pre-sales effort can reduce this.	Free to download versions are available to support proof of concept work. However, it should be noted that where Proprietary vendors typically have extensive pre-sales support this is not generally available for Open Source solutions. This can have an adverse impact on the cost of evaluation.	
Upfront	H	Capital cost of software	One of the most important considerations with respect to upfront cost.	High.	Zero although there may be some capital cost for proprietary extensions.	

	H	Cost of functional customisation	One of the most important considerations with respect to upfront cost. It is dependent on: - The functional fit of the software to the business's requirements - The ease of customising the software to meet the requirements - The cost and availability of skills to make the customisations	Dependent on functional fit not on licence model. Factors to consider are ease of implementation and availability / cost of resource to carry out customisation
	M	Cost of integration	Similar to, or though usually less significant, functional customisation. It is dependent on: - The ease of integration. Partly based on integration options and partly on the quality of design (i.e. is there a well documented and designed API) - The cost and availability of skills to carry out the integration	Dependent on integration complexity not on licence model. Factors to consider are: how well the solution is architected with respect to integration, whether standard integration patterns and formats are supported, and cost of resource
	M	Migration cost	Cost of migrating data and users from the legacy solution to the new solution. The cost of migration may be reduced if the source and target solutions conform to documents and data standards (something that is more common in Open Source solutions).	Dependent on data volume and complexity not licence model.
	M	Cost of infrastructure	Determined by a number of factors that are not greatly influenced by the licence model: - The expected load that the solution will place on the infrastructure - The Operating System and RDBMS that the solution requires (Open Source solutions are less likely to be restrictive)	Dependent on quality of design and non-functional requirements not licence model.
	L	Training cost	This is usually a small component of the upfront cost but for some desktop applications or core business applications it may be significant. Where	Dependent on quality and familiarity of User Interface not licence model.

			users are very familiar with existing solutions (e.g. Microsoft Office) and the new solution involves a transition then this may become a deciding factor in the TCO.		
	L	Tooling cost	The customisation of the solution to meet the business' needs and the data migration effort may require specific tools that have independent cost. These costs should be considered part of the TCO. Typically the tools used to customise and develop Open Source solutions are free of charge where those to customise proprietary solution often carry additional cost.	Not dependent on licence model.	
Support	M	First line support cost	This is most likely to be a cost that is shared across multiple applications / solutions that require support. It will be influenced by how intuitive the solution is to use (the quality of the UI) and by the administrative tools available to support staff (e.g. admin consol to reset user passwords or change access rights).	Assuming that first line support is largely non-technical then this is not dependent on licence model.	
	M	Second line support cost	The magnitude of this cost will be directly proportional to the level of customisation carried out and the capabilities of the supporting organisation. Solutions that are largely "out the box" and that align with an organisation's IT Strategy are likely to have a lower cost of ownership.	Vendor certification for support teams drive quality but leads to a closed / expensive market.	Second line support can be sourced from a competitive market complicating procurement but potentially driving down cost.
	H	Third line support cost (software)	Proprietary solutions typically come with support contracts that provide a route to resolve defects. However, careful consideration needs to be given to determine whether the support actually provides any real assurance, for example are binding SLAs included the support contract and	Defects are resolved for fixed (known) annual price. However, as vendor delivered fix may be slow to arrive an allowance should be made for the cost of developing workarounds. Typically there is a view that there is "on throat to	Defects will be resolved at cost (probably on a T&M basis) using community support. This drives variable cost but does mean client controls time

			what are the penalties of missing these.	choke" but need to question what this real gives you verse what the perception might be.		to develop. Competition can be introduced into the support as not tied to a single vendor.
			This row considers risk for the different support models	Vendor drives speed of defect resolution. Risk is that vendor does not value customer and is slow	Vendor will provide fix in most instances but customer can develop own fix if required	Capability of support provider drives speed of defect resolution. Risk is that provider is not capable
	L	Third line support cost (infrastructure)	Infrastructure support includes the cost of resolving infrastructure failures, restoring backups, changes to firewalls etc.	Not dependent on licence model		
Maintenance	H	Software maintenance (of the vanilla product)	Enterprise level software typically requires defect and security releases to address identified defects and security flaws. These releases can be provided as patches or complete releases.	Vendors typically provide defect releases and security patches as part of annual maintenance fee or subscription fee. In addition to this fee the cost of applying defect fixes and security patches should be factored into the TCO as should the cost of periodically moving to a major new release		Published vulnerabilities and release of defect fixes / security patches need to be monitored. The cost of this and the cost of applying fixes needs to be considered.
	H	Software maintenance (of the bespoke or customised parts of the solution)	Where the solution includes a significant proportion of bespoke code or customisation of the base software product then there is a need to provide ongoing maintenance for the bespoke element. This will not be provided by the software vendor.	The software vendor will not usually take responsibility for the maintenance of bespoke code and therefore a separate support structure will have to be put in place.		There is the opportunity to bring some bespoke code into the Open Source project and therefore reduce the maintenance cost.
	H	Software upgrades	For solutions that will be in place for a number of years there is a need to consider the cost of moving to new releases of the software if this is required. The key determinate is whether it is required at all.	Ongoing support agreements may force software upgrades even if it is not required for business reasons. This may drive both increased licence cost and upgrade costs.		Customer is free to choose when to upgrade.

	H	Enhancements	The cost of making functional enhancements to meet the changing needs of the business.	Not dependent on licence model		
	H	Software scaling	The cost of scaling the solution to support higher user or transaction volumes.	Meeting growing demands likely to require the purchase of additional software licences / capital investment. Reducing scale is unlikely to release capital.	Meeting growing demands can drive additional support cost / operational cost. Reducing scale will equally drive reduced operational cost.	Zero software cost to scale
	H	Infrastructure scaling	Cost of scaling infrastructure to support more (or less) scale	Not dependent on licence model. Quality of design and "heaviness" does impact. Light-weight solutions (majority of Open Source and some proprietary solution fall into this category) will have lower infrastructure scaling cost.		
Exit	M	Termination	The potential commercial exit costs	Commercial agreements may mean that support costs do not stop if the solution is retired.		Zero.
	M	Migration	The cost of moving data and content from the proposed to solution to a future solution that is replacing it.	Not dependent on licence model but is dependent on the extent to which the solution implements open standards and provides open access to data.		