based on OSS [5]. Research on characterising OSS migration initiatives has been performed [22]. They found that software migrations from proprietary to open source depend on organisational and contextual factors such as the IT resources accessibility, organisational climate, organisational complexity, political support, why the change is needed and the project leadership style.

An overview of OSS migration and criteria for migration challenges has been presented [17]. He points out that organisations migrate to OSS from legacy systems because the legacy systems are difficult to integrate with the newer technologies. The OSS migrations can include:

- Language or code migrations;
- Operating systems migrations;
- Data migrations;
- User interface migrations;
- Architecture migrations.

8. Benefits of OSS vs. CSS – A Comparison

Table 3 is a comparison of the benefits of OSS and Closed Source Software by different authors. This Table reveals that there are more profound benefits of OSS than for closed source software.

Table 3. Comparing the Benefits of OSS and CSS

Benefit / Characteristic	Open Source Software (OSS)	Closed Source Software (CSS)	Author
Reliability	OSS has increased reliability over closed source software. The reason is that OSS is usually critically examined by many independent and enthusiastic developers during all its developmental stages.	The reliability of some closed source software is lower than that of OSS. The reason is that CSS is produced by a smaller number of developers who work against tight deadlines under much pressure.	[43] [12] [13]
Sense of Urgency	There is little sense of urgency in OSS projects; there are little or no strict deadlines, and no hierarchical team structure in OSS developments.	Due to stringent deadlines to be met, there is a sense of urgency of CSS projects. There is a hierarchical team structure in closed source projects – the corporate world.	[30]
Quality	The quality of OSS is perceived to be higher than that of CSS. This is because many developers examine the software, facilitating the detection of errors.	CSS is perceived to have a lower quality than OSS. Developers outside the closed group cannot detect errors because the source code is generally not publicly available.	[12] [32] [13]
	The quality of OSS products should be higher than for CSS if there *is* competition between them in the market	Quality of CSS could be higher than quality of OSS if there *is no* competition in the market.	[46]
	Generally there are no formal inspections in the quality of OSS programs and no broad testing. There is little evidence to support rigorous measurements in OSS.	There are formal inspections conducted in CSS projects as well as broad testing. Rigorous measurement is performed in CSS imple-mentations.	[30]

Innovation & Flexibility	OSS has more flexibility than CSS – source code is publicly available. By providing users with the freedom and flexibility, OSS enables innovation to modify the software without any restriction.	CSS has less flexibility than OSS due to its code being closed. Users are not allowed to see the source code and this restricts innovation. But it facilitates the security and reliability of the software. They have targeted innovation that is business focused rather than technology focused.	[12] [13] [8]
Software Requirements	Requirements are mostly absent in OSS projects. There is little systematic effort in addressing Capability Maturity Models (CMMs). There is also little evidence of using the Unified Modeling Language (UML) or any form of systematic modeling in OSS.	Requirements are used in CSS projects. The Capability Maturity Model (CMM) is well addressed in CSS projects. Closed source projects make use of UML or other modeling techniques.	[30]
Vendor-Lock Ins	There is no Vendor-Lock In associated with OSS. The user is independent of the vendor.	CSS is dependent on the Vendor. Therefore, there is vendor-lock in.	[12] [13]
Cost	OSS tends to be free; and have low acquisition cost, except for having to pay for the media on which the software may be distributed (e.g. on a CD). The total cost of ownership may roughly be the same as for some closed source programs.	Most CSS are not free and have a higher acquisition cost than OSS. However, in some situations closed source Total Cost of Ownership (TCO) is lower than that of open source. TCO for closed source and open source software could roughly be the same.	[12] [13] [59] [46] [8]
Adherence to Standards	The use of standards is limited to data formats like the Hypertext markup language (HTML), or the Extensible markup language (XML).	Closed source projects normally adhere to most IT standards during implementation.	[30]
Usability / Ease of code errors identification and problem solving	Most OSS products offer code error reporting tools. These tools assist in the faster detection of errors and the rapid finding of solutions.	Generally, it requires a much longer period to resolve errors in CSS, due to non-availability of code error reporting tools.	[59]

	OSS usually lacks usability because it is developer-centric. Ability to correct errors is limited to users with technical expertise.	Closed source programs do not lack usability. They employ expert usability testing techniques and usability is ranked quite higher than in OSS.	[8] [32]
Operating Systems	OSS products are supported with operating systems that surpass the operating systems that support CSS due to the availability of source code which can be altered. Users can adapt the OSS to their operating systems. The cost of such a diversity of operating systems tends to be higher in closed source systems due to their high development costs.	It is more expensive to change the operating system source code of a CSS. Development costs are generally high. Users usually have to wait for a next release of the software.	[59]
Documen-tation	Most OSS projects are weak on document-tation.	Most CSS projects produce manual and quality documentation.	[59]
	OSS are not legally bound to produce document-tation such as manuals or guides.	Closed source programs are legally required to supply documentation such as user manuals, guides etc.	[13]
Personali-sation	This is the degree to which developers are able to write applications in the way they want the application to look and are used. OSS developers use personali-sation a lot in their work in order to change the look and feel of a product, so that it can integrate seamlessly with their working environment. This enhances their efficiency and mood.	CSS developers are generally not allowed to attach personalisation to their work. Company standards and policies have to be adhered to and CSS is designed to accommodate the generic software market.	[59]
Service and Product Support	OSS products come with many learning materials obtainable from the developer's site or other locations supporting the OSS product. Large community of users and developers support OSS products by designing tutorials and short articles on	Closed source systems are supported by a support team and they usually make use of printed material or books which come at a cost.	[59]

	how the product should be used. User groups are available and support is delivered via forums and blogs. Issues may, or may not be resolved soon.	Closed source programs have a high response service. Ongoing support is provided to the customer. Support to the users of CSS is arguably the greatest advantage of using CSS.	[8]
Plug-in functionality	Is readily available for OSS products. OSS developers and users can extend the functionality of their product by using Plug-ins to write their own modules which can be integrated with the OSS product.	It is more difficult to write Plug-ins for Closed Source systems than OSS because documentation is not as rich as the OSS. The source code is also not readily available.	[59]
Highly specialised Applications	OSS programs are less likely to be used to develop highly specialised applications. There is little evidence that formal specifications are used in OSS projects and this limits the use of OSS in safety-critical software.	CSS can be used effectively to develop highly specialised applications. Formal specifications are used in closed source projects and this enhances their use in safety-critical software.	[46] [30]
Best-practices Project Management	PM practices are usually lacking in most OSS projects and this could undermine the product's quality. Release management guidelines are informal in OSS and there are often version proliferation and implement-ation issues.	Most closed source projects use best-practices project management techniques, all of which enhance a product's quality. Most closed source projects follow release management guidelines.	[46] [30]

Discussion of Table 3

The reliability of some CSS may be lower than that of OSS owing to fewer programmers that develop closed source software, working against tight deadlines and under a fair amount of pressure [12] [13] [43]. Closed source software is perceived to have a lower quality and lower flexibility than OSS due to the non-availability of the source code [12] [13] [32]. However there are arguments that CSS is of a higher quality than OSS, provided that there is no competition in the market [30] [46].

Most CSS implementations make use of a modeling language like Unified Modeling Language (UML), as well as incorporating the Capability Maturity Model (CMM). In contrast, OSS implementations usually do not make use of any modeling techniques like UML; neither do they use the CMM [30].

The Total Cost of Ownership (TCO) of both OSS and closed source software are roughly the same [8]. Closed source programs do not lack usability, documentation or service/product support, whereas OSS programs usually lack usability and documentation [8] [30]. There is no vendor lock-in associated with OSS but closed source software is characterized by vendor lock-ins [12] [13].

According to Raghunathan, the comparisons of open source and closed source are not conclusive, or in a finer analysis are slightly in favour of open source [46]. This is also the view of Khanjani, namely, that OSS yield more benefits than CSS [32].

More enthusiastic developers are involved in developing, testing and evaluating the code of OSS programs.

9. Comparing OSS and CSS Security

The importance of analyzing a whole OSS system when performing an extensive security investigation has been emphasised [20]. Such analyses include the application software, its source code, and the tools used for developing the object code. Examples are compilers, operating systems, hardware and the whole development environment.

Different authors have different perceptions when they compared OSS security with that of CSS as shown in Table 4. The table reveals that the security of OSS is roughly of the same quality as that of a CSS system.

Table 4. Comparing OSS and CSS Security

Characteristic	OSS security	CSS security	Author
Publishing of Designs and Protocols	OSS designs and protocols are published and these contribute to the security of the systems. This may reveal logical errors in the security of the system.	Closed source designs and protocols are not published.	[25]
Finding and correcting security vulnerability	It is easier to find and correct code errors in OSS than in CSS owing to the openness factor.	Open and closed approaches to security are rather similar. Correcting errors in CSS is dependent on the programming team that developed the program – the source code is not publicly available.	[10]
Checking and Testing of code	OSS users have the freedom to validate and test the code of the OSS product that they want to use so as to ascertain its quality and security.	Because users do not have the choice to validate and test the code in closed systems, the author stresses that OSS initial coding tends to be of a higher quality than CSS.	
Controlled Environment Development	OSS is often viewed as having security issues because OSS is not necessarily developed in a controlled environment.	CSS is perceived as being more secure because it is developed in a controlled environment by a concentrated team with a common direction. The source code may be viewed and edited only by this team. The software is comprehensively audited, eliminating the risk of back door Trojans and reducing the risk of code errors or other software issues.	[8]
Closeness or openness of software code – security through obscurity	It is maintained that OSS improves software transparency, security and trustworthiness because users and developers can validate an OSS program's functionality and security, due to the availability	The authors stress that the security of software is dependent on the user and not necessarily its closedness or openness. CSS can also be as secure as OSS.	[20]

	of its source code. They highlight that it is easier to correct bugs in OSS systems thereby enhancing the quality of code. This could also lead to the use of better project management and quality control. Open source users can independently evaluate the security for themselves. The real exposure of the system can be assessed and the gap between perceived and actual exposure is diminished.	CSS does not allow users of such software to evaluate its security for themselves. This does not allow users to easily discover weaknesses and 'patching' is not possible by users.	[25]
Analysis of published vulnerabilities	There are no significant differences in terms of vulnerability severity found between open source and closed source. More and faster patches can be found in open source systems. Patches for open source systems are released faster than for closed source systems.	The vulnerability severity found between open source and closed source are perceived to be the same. Patches for vulnerabilities of closed systems are released weeks or months after the discovery of the vulnerabilities and this increases the risk of using the system.	[47] [25]
	Patch management is harder to coordinate in open source systems because OSS comes in many different versions. Patches will not be available for some distributions and they may be vulnerable to attacks while others are being patched. OSS products are more secure than CSS products. However, their general pattern of vulnerability detection is similar.	The authors claim that it is easier to manage patches in a closed source system than in an open source system. CSS products are less secure than OSS products.	[7] [61]

Discussion of Table 4

Closed source designs and protocols are not published, whereas the OSS designs and protocols are published enhancing the security of OSS programs since logical errors may be revealed [25]. This is also the view of Dwan that due to the openness of OSS code, it is easier to find and correct errors in OSS than in CSS [10]. This is also pointed out by Hoepman that more and faster patches are found in OSS whereas patches are not released as fast in CSS, thereby increasing the risk of using the system securely [25].

OSS users have the freedom to validate and test the code in order to ascertain its quality and security, therefore OSS initial coding tends to have higher quality and security than CSS [34]. However, Daniel

argues that CSS is perceived to be more secure than OSS because it is developed in a controlled environment by a dedicated team of developers with a common direction [8].

The view of Hansen is that CSS can be as secure as OSS because the security of software is dependent on the user and not on its openness or closedness [20]. The severity of vulnerabilities found between OSS and CSS are similar as pointed [47]. While our view is that OSS is more secure than CSS, there are, however, security challenges that have to be overcome when migrating from a closed system to an open system [17].

10. Security Challenges during Migration to OSS

A list of items that can be migrated is presented by Geetha and these are: (a) Language or code migrations, (b) Operating system migrations, (c) Data migrations, (d) User Interface migrations and (e) Architecture migrations [17]. He points out that the challenges to migration from Legacy systems to OSS include: (i) Qualification and selection of OSS, (ii) Human factors such as: Fear of the new software; Knowledge is power; Cost of training personnel for the new tools; reduced productivity of the personnel and (iii) Technical challenges. The technical challenges include: Usability; Software Development Service and support; Security; Data migration; and OSS Code Maintenance and Management [11].

According to Geetha and ElHag, the security challenges during migration to OSS are: (a) Detecting security risks, bugs, and errors, (b) Eliminating the bugs and errors and (c) Obtaining metrics for measuring software security for real-time and mission critical software [11] [17].

11. A Model for Addressing the Security Challenges during Migration to OSS

Summative content analysis was used as the research method to explore the model for addressing the security challenges during migration to OSS. During summative content analysis, the keywords (derived from review of literature) are identified before and during data analysis [26]. Keywords are extracted from the literature and mostly from the two articles written by Anner and Ajigini [1] [3]. An open source assessment framework and a threat modelling methodology, pioneered by Microsoft since 1999 have been highlighted, this is then proposed by Anner to overcome the security challenges of OSS [3] [53]. The aim is to reduce the risks to confidentiality, integrity and availability and to identify and reduce threats, vulnerabilities and risks to an acceptable level. They mention that alternative methods to reduce risks include: (a) Code auditing (b) Penetration testing, and (c) Using Statistical analysis tools.

As per Anner, the threat modelling process consists of four stages, viz: (i) Application Analysis/Diagramming (ii) Threat Enumeration, (iii) Threat Rating, and (iv) Mitigation Options [3]. They point out that the threat modeling approach with slight modifications can assist with the identification of security vulnerabilities, as well as investigating coding issues and implementation mistakes.

A Rudimentary Management Framework to protect sensitive information during the migration to an open source system is suggested [1]. The model we propose in this section for addressing the security challenges discussed in this paper in migrating to OSS, is based in part on the threat-modeling framework in Anner and the sensitive information migration framework [1] [3].

Our model is illustrated in Fig. 1 and is discussed below:

During the Application Analysis/Diagramming phase (A), the applications are analyzed from a flow of data perspective. All the aspects that make up the applications are catalogued and the relationships between the assets in terms of data exchange are identified through a UML Class-oriented structure.

The Threat Enumeration phase (B) consists of analyzing each element in the Class-oriented UML against a list of potential threats depending on the element type using the STRIDE Taxonomy [28]. STRIDE is used as a classification schema to characterize known threats in accordance to the attacker motivation.

The risk levels for each of the enumerated threats are determined and ratings of all threats are established during the Threat Rating phase (C).

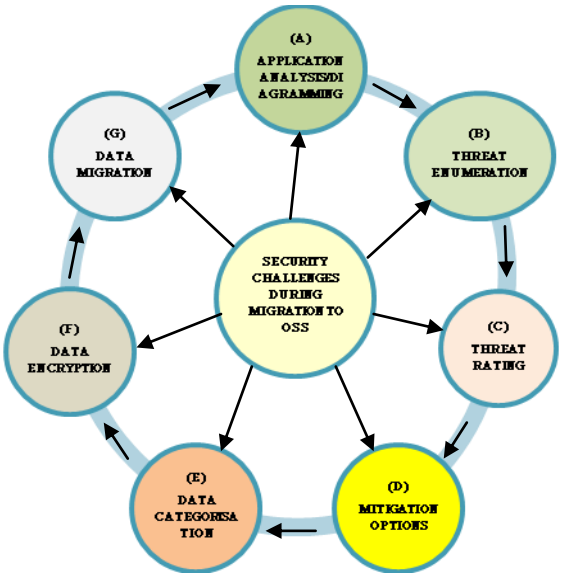

Figure 1. Modelling security challenges during OSS migration

During the Mitigation Options phase (D), all functionality and patching are removed and other security controls are added and redesigned.

The business rules and the data classification system are used to classify migrated data during the Data Categorisation phase (E).

Data protection tools and Privacy enhanced technologies are used to encrypt the data during the Data Encryption phase (F).

The encrypted data is now migrated during the Data Migration phase (G).

Implementing the Proposed Model

The following processes are proposed to implement the model in Figure 1:

Phase A: Application Analysis/Diagramming Phase –

a) Identify security objectives – user identity protection, privacy and regulation, availability guarantees of applications.
b) Catalogue all the applications.
c) Analyse all the application designs and architectures to identify the components using Data Flows.
d) Identify UML component diagrams.
e) Identify the relationships between the assets using data exchange by using Class-oriented UML structures.

Phase B: Threat Enumeration Phase –

a) Analyse each element in the Class-oriented UML diagram against potential threats by using the STRIDE Taxonomy.
b) Analyse data movement across trust boundaries (e.g. from Internet to Web tier).
c) Identify the features and modules with a security impact that need to be evaluated.
d) Investigate how data enters modules, how modules validate and process the data, where the data flows to, how the data is stored and what fundamental decisions and assumptions are made by the modules.

Phase C: Threat Rating Phase –

a) Identify threats using Bugtraq tools and techniques. Bugtraq is a mailing list containing information on how to exploit and use intrusion detection systems vulnerabilities in defending networks.
b) Determine the risk levels of each threat.
c) Establish the ratings of all the threats.
d) Use either a threat graph or a structured list to write out the threats.

Phase D: Migrations Options Phase –

a) Remove the functionality and patching.
b) Add other security controls.
c) Redesign other security controls.

Phase E: Data Categorisation Phase –

a) Develop business rules.
b) Develop a Data classification system.
c) Classify data based on business rules and the above data classification system.

Phase F: Data Encryption Phase –

a) Deploy Data Protection Tools.
b) Deploy Privacy Enhancement Technologies.
c) Use secure Tools to encrypt the data.

Phase G: Data Migration Phase –

a) Ensure that data to be migrated are encrypted by using verification techniques.
b) Migrate the encrypted data.

12. Conclusions

In this paper we investigated the notions of closed source software (CSS) and open source software (OSS); the security issues and challenges of migrating from CSS to OSS were investigated, we discussed the respective advantages of each and considered comprehensively the security aspects underlying each approach to software development.

A comparison of the benefits of OSS and closed source software by different authors was explored. The comparisons of the benefits of open source and closed source are slightly in favour of open source. Additionally, a comparison of OSS and CSS security was undertaken and our view is that OSS is more secure than CSS.

Using summative content analysis, the challenges in migrating from a closed system to an open system were identified, and these, together with two frameworks – one for threat modelling and another for protecting sensitive information during system migration were used to propose a model for addressing the various security aspects in migrating from an open system to a closed one [1] [3]. Our model is based on a seven-phase process as presented in Figure 1. It is anticipated that this model may be useful as a basis for mitigating the security challenges in moving from a closed (CSS) to an open (OSS) system.

13. Future Work

Future work in this area may be pursued along a number of lines: The framework proposed for protecting sensitive information during system migration has to be further integrated with the security-protection model proposed in this paper [1]. In particular the classification of sensitive information in phase 5 – Data Categorisation has to be further developed. Having implemented our

model, we have to validate it in industry at companies that have migrated to OSS, as well as those who are yet to undertake such migration.

14. References

[1] Ajigini, O. A., Van der Poll, J. A. and Kroeze, J. H. (2012) 'Towards a management framework to protect sensitive information during migrations', in Proceedings of the 2nd International Conference on Design and Modeling in Science, Education and Technology (DeMSet), pp. 6-13.

[2] Allen, J. P. (2012) 'Democratizing business software: Small business ecosystems for open source applications', Communications of the Association for Information Systems, 130 (28), pp. 483-496.

[3] Anner, Y. and Cid, C. (2010) Open-Source security assessment, Royal Holloway Series.

[4] CENATIC, (2008) 'Open source software for the development of the Spanish public administration: An overview'; www.cenatic.es (7 April, 2011).

[5] CENATIC, (2009). 'Study on the situation of open source software in universities and R&D centers', Report, Almendralejo; www.cenatic.es (7 April, 2011).

[6] Chengalur-Smith, I., Nevo, S. and Demertzoglou, P. (2010) 'An empirical analysis of the business value of open source infrastructure technologies', Journal of the Association for Information Systems, vol. 11, Special issue, pp. 708 - 729.

[7] Clark, R., Dorwin, D. and Nash, R. (2009) 'Is open source software more secure?' Homeland Security / Cyber Security; http://www.cs.washington.edu/education/courses/csep590/05au/whitepaper_turnin/oss%2810%29.pdf. (2 February, 2003).

[8] Daniel, J. (2009) 'Open source vs. closed source software: The great debate'; http://www.articlesbase.com/internet-articles/open-source-vs-closed-source-software-the-great-debate-1040071.html (1 February, 2013).

[9] Drozdik, S. and Kovacs, G. L. (2005) 'Risk Assessment of an open source migration project', In Proceedings of the First International Conference on Open Source Systems, Geneva, M. Scotto & G. Succi (eds.) pp. 246 – 249.

[10] Dwan, B. (2004) 'Open source vs. closed', Network Security, vol. 5, pp. 11-13.

[11] ElHag, H. M. A. and Abushama, H. M. (2009) 'Migration to OSS: readiness and challenges'.

[12] Fitzgerald, B. (2006) 'The transformation of open source software', MIS Quarterly, 30(3), pp. 587 - 598.

[13] Gallegoa, M. D., Lunab, P. and Bueno, S. (2008) 'User acceptance model of open source software', Computers in Human Behaviour, 24(5), pp. 2199 - 2216.

[14] Gallopino, R. (2009) 'Open Source TCO: Total Cost of Ownership and the Fermat's Theorem'; http://robertogaloppininet/2009/01/08/open-source-TCO-total-cost-of-ownership-and-the-Fermats-theorem/(ed.) (21 March, 2012)

[15] Gartner, (2008) Gartner highlights: Key predictions for IT organisations and users in 2008 and beyond.

[16] GCIS, (2007) 'Cabinet Statement.' Feb 22; www.gcis.gov.za/media/cabinet/2007/070222.htm (2 February, 2012.

[17] Geetha, S. (2012) 'Possible challenges of developing migration projects', International Journal of Computers & Technology, 3(3), pp. 463 - 465.

[18] Gwebu, K. L. and Wang, J. (2010) 'An exploratory study of free open source software users' perceptions', The Journal of Systems and Software, 83(11), pp. 2287 - 2296.

[19] Gwebu, K. L. and Wang, J. Wang (2011) 'Adoption of open source software: The role of social identification', Decision Support Systems, vol. 51, pp. 220 - 229.

[20] Hansen, M., Kohntopp, K. and Pfitzmann, A. (2002) 'The open source approach – opportunities and limitations with respect to security and privacy', Computers & Security, 21(5), pp. 461 - 471.

[21] Hauge, O., Ayala, C. and Conradi, R. (2010) 'Adoption of open source software in software-intensive organisations – A systematic literature review', Journal of Information and Software Technology, vol. 52, 1133 – 1154.

[22] Heredero, P., De, C., Berzosa, D. L. and Santos, R. S. (2010) 'The implementation of free software in firms, an empirical analysis', The International Journal of Digital Accounting research, 10(6).

[23] Hedgebeth, D. (2007) 'Gaining competitive advantage in a knowledge-based economy through the utilization of open source software', VINE: The Journal of Information and Knowledge Management Systems, 37(3), pp. 284 – 294.

[24] Hislop, R. (2004) 'Mossel Bay adopts Linux on the desktop', Electronic Government Africa, 1(1), pp. 14.

[25] Hoepman, J. and Jacobs, B. (2007) 'Increased security through open source', Communications of the ACM, 50(1), pp. 79 - 83.

[26] Hsieh, H. and Shannon, S. E. (2005) 'Three approaches to qualitative content analysis', Qualitative Health Research; http://qhr.sagepub.com/content/15/9/1277 (19 September, 2014).

[27] Lewis, J. A. (2007) 'Government open source policies'; http://www.csis.org/media/csis/pubs/070820_open_source_policies.pdf, (29 January 2012).

[28] Little, G. and Stergiades, E. (2009) Worldwide Open Source Services, 2009-2013 Forecast.

[29] James, S. and Van Belle, J. (2008) 'Ensuring the long-term success of OSS migration: a South African exploratory study', 6th Conference on Information Science Technology and Management, New Delhi, India.

[30] Kamthan, P. (2007) 'A perspective on software engineering education with open source', IGI Global.

[31] Kemp, R. (2009) 'Current developments in open source software', Computer Law & Security Review, vol. 25, pp. 569 - 582.

[32] Khanjani, A. and Sulaiman, R. (2011) 'The aspects of choosing open source versus closed source', IEEE Symposium on Computers & Informatics, pp. 646-649.

[33] Kovacs, G. L., Drozdik, S., Zuliani, P. and Succi, G. (2004) 'Open source software for the public administration', In Proceedings of the 6th International Workshop on Computer Science and Information Technologies, Budapest, Hungary.

[34] Manfield-Devine, S. (2008) 'Open source: does transparency lead to security?', Computer Fraud & Security, pp. 11 - 13.

[35] Miscione, G. and Johnston, K. (2010) 'Free and open source software in developing contexts, from open in principle to open in the consequences', Journal of Information & Ethics in Society, 8(1), pp. 42 - 56.

[36] Mtsweni, J. and Biermann, E. (2010) 'A roadmap to proliferate open source software usage within SA Government servers', Third International Conference on Broadband Communications, Information Technology & Biomedical Applications, IEEE Computer Society, pp. 430 - 436.

[37] Mutula, S. and Kalaote, T. (2010) 'Open source software deployment in the public sector: a review of Botswana and South Africa', Emerald, 28(1), pp. 63 – 80.

[38] Nagler, M. (2005) 'Open Source adoption of the German Federal Office for information security'; http://ec.europa.eu/idabc/servelets/Doc?id=21394 (25 January 2012).

[39] Open Source Report, Coverity Report, (2008); http://scan.coverity.com/report/ (1 February, 2013).

[40] Open Source Security Study, Fortify Report, (2008); www.fortify.com/ossreport.html (1 February, 2013).

[41] Oram, A. (2011) 'Promoting open source software in Government: The challenges of motivation and follow-through', Journal of Information Technology & Politics, 8(3), pp. 240 – 252.

[42] Otter, A. (2007) 'SA government gets serious about ODF, IOIL Technology'; http://www.ioltechnology.co.za/article_page.php?iArticleId=4126700&iSectionId=2888, (21 February 2012).

[43] Pearson, H. E. (2000) 'Open source licenses, open source – The death of closed source Systems?' Computer Law & Security Report, 16(3), pp. 151 - 156.

[44] Poulter, A. (2010) 'Open source in libraries: an introduction and overview', Emerald, 59(9), pp. 655 – 661.

[45] Rafiq, M. and Ameen, K. (2009) 'Issues and lessons learned in open source software adoption in Pakistani libraries', The Electronic Library, vol. 2794, pp. 601 – 610.

[46] Raghunathan, S., Prasad, A., Mishra, B. K. and Chang, H. (2005) 'Open source versus closed source: Software quality in monopoly and competitive markets', IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 35(6), pp. 903 - 918.

[47] Schryen, G. (2009) 'Security of open source and closed source software: An empirical comparison of published vulnerabilities', AMCIS Proceedings, Paper 387.

[48] Scola, N. (2009) 'Why the White House's embrace of Drupal matters', Personal Democracy Forum techPresident.

[49] SERPRO, (2005) 'Fast move to free software in Brazil'; http://ec.europa.eu/idabc/en/documents/5131/528, (20 January, 2012).

[50] Shaikh, M. and Cornford, T. (2012) 'Strategic drivers of open source software adoption in the public sector: Challenges and opportunities', European Conference on Information Systems AIS, Paper 237.

[51] Sharma, A. and Adkins, R. (2006) 'OSS in India', In DiBona, C., Cooper, D. and Stone, M. (eds.), Open Sources 2.0, O'Reilly Media, Sebastopol, CA, pp. 189 - 196.

[52] Shaw, A. (2011) 'Insurgent expertise: The politics of free/live and open source software in Brazil', Journal of Information Technology & Politics, vol. 8, pp. 253 – 272.

[53] Shostack, A. (2008) 'Experiences threat modelling at Microsoft', In Modelling Security Workshop, Dept. of Computing, Lancaster University, UK; http://blogs.msdn.com/sdl/attachment/8991806.ashx (2 February, 2013).

[54] Stol, K., Babar, M. A., Russo, B. and Fitzgerald, B. (2009) 'The use of empirical methods in open source software research: Facts, trends, and future Directions', FLOSS, Vancouver, Canada, pp. 19 - 24.

[55] Swiderski, F. and Snyder, W. (2004) Threat Modelling, Microsoft Press.

[56] The Guardian, 2004 'Open invitation taken up at last'; http://society.guardian.co.uk/epublic/story/0,,1362744,00.html. (2 February, 2013).

[57] Thomas, J. (2007) 'Malaysian public sector OSS program Phase II: Accelerated adoption';

http://www.oscc.org.my/documentation/phase2_launching/OSS-Phase-2Strategy-Plan-Launch.pdf (5 June, 2013).

[58] TMPSOSSSMP, (2008) 'The Malaysian public sector open source software master plan: Phase II – Accelerated adoption'; http://www.mampu.gov.my/seminar%20ict/kk2-OSS.pdf (19 February, 2012).

[59] Vintila, B. (2010) 'Citizen oriented open source security', Open Source Science Journal, 2(3), pp. 57 - 64.

[60] Vital W. (2006) 'The South African adoption of open source', White paper created by Vital Wave Consulting; www.vitalwaveconsulting.com/insights/South-African-Adoption-of-Open-Source.pdf, (20 February, 2012).

[61] Walia, N., Rajagopalan, B. and Jain, H. (2006) 'Comparative investigation of vulnerabilities in open source and proprietary software: An exploratory study', Americas Conference on Information Systems (AMCIS), vol. 108, pp. 848 - 857. Weber, T., 2004. *The success of open source*. Harvard University Press, New York, NY.

[62] Weber, T. (2004) The success of open source. Harvard University Press, New York, NY.

[63] Weilbach, L and Byrne, E. (2010) 'A human environmentalist approach to diffusion in ICT policies – A case study of the FOSS policy of the South African Government', Journal of Information Communication & Ethics in Society, 8(1) pp. 108 – 123.

[64] Yeo, B., Liu, L. and Saxena, S. (2006) 'When China dances with OSS', In DiBona, C., Cooper, D. and Stone, M. (eds.), Open Sources 2.0, O'Reilly Media, Sebastopol, CA, pp. 197 – 210.

