



A Framework of Collaborative Knowledge Management System in Open Source Software Development Environment

Modi Lakulu (Corresponding author)

Faculty of Information Technology and Communication, Sultan Idris University of Education

35900, Tg Malim, Perak, Malaysia

Tel: 60-19-2894-877 E-mail: modi@ftmk.upsi.edu.my

Rusli Abdullah, Mohd Hasan Selamat, Hamidah Ibrahim & Mohd Zali Mohd Nor

Faculty of Computer Science and Information Technology, University Putra Malaysia

43400, Serdang, Selangor, Malaysia

Tel: 60-3-8946-6557 E-mail: rusli, hasan, hamidah {@fsktm.upm.edu.my}, mohd.zali@my-newstar.com

Abstract

The global economy crisis reveals the advantages of Open Source Software (OSS). Software developers benefit not only from reduced cost of acquisition, but also access to source code and components. In this aspect, knowledge sharing among developers are immensely important in all facets of System Development Life Cycle (SDLC). Feller and Fitzgerald (2000) raised the critical questions on what life-cycle underpins the OSS model and what is the best methodology to support the OSS as well as what toolkit support OSS methodology. This paper shall discuss the formulation of Knowledge Management System (KMS) framework for sharing knowledge in OSS using SDLC from the planning phase until the maintenance phase. An initial fact finding survey was conducted on selected OSS developers in Malaysia to analyze the current usage and acceptance of OSS. The results are quite unexpected, with many OSS developers are still not fully using OSS tools in SDLC stages. The proposed KMS model is envisaged to allow OSS Community-of-Practice to share the OSS knowledge for the whole SDLC.

Keywords: Open source software, Knowledge, Knowledge management, Community of practice, KMS framework

I. Introduction

The OSS has become increasingly important and has attracted developers from both public and private sectors. Open source model, as a radically new software development model, begins in the mid-90s. Since then, many software created by open source model have been widely adopted and used by various industries. OSS is released under license conforming to the Open Source Definition (OSD) as articulated by the Open Source Initiative (OSI) [Feller and Fitzgerald, 2000]. OSS source codes are published and made available to the public, allowing everyone to copy, modify, and redistribute the source codes without paying royalties or fee [Jennex, 2006]. The most important requirements of an OSS system is that the source codes must be freely available to anyone who wish to examine or change for their own purpose [Godfrey and Tu, 2001]. This creates software which are reliable, high quality, inexpensive and timely [Feller and Fitzgerald, 2000].

Managing knowledge in OSS development is crucial to allow developers to capture, locate and share the knowledge on codes and methods. Knowledge is defined as a fluid mix of frame experience, values, contextual information and expert insight that provides a framework for evaluating and incorporating new experience and information. It often embed not only in documents or repositories but also in organizational routines, processes, practice; and norm [Davenport and Prusak, 1998]. There are two types of knowledge; explicit and tacit. Explicit knowledge is clearly formulated or defined, easily expressed without ambiguity or vagueness, and codified and store in a database. Tacit knowledge is the unarticulated knowledge that is in a person's head that is often difficult to describe and transfer [Bollinger and Smith, 2001].

Knowledge Management (KM) in technical perspective, refers to the methods and tools for capturing, storing, organizing, and making accessible the knowledge and expertise within and across communities [Canfora, Cimitile, Visaggio, 2002]. The idea of a KM system is to enable employees to have ready access to the organization's based documented facts, sources of information, and solutions for a specific Community-of-Practice (CoP) members such as software engineer, database administrator, programmer and users.

In this paper we focus on System Development Life Cycle (SDLC) as a method in software development, using

information available throughout SDLC, as the main source of knowledge. Hence, the CoP can share knowledge throughout the development process such as the OSS tools they used in each phase, documentation, source code, support and best practice. Besides, to increase the participation of CoP, the communication infrastructure is very important as open source projects rely heavily on tools as modes of communication [Barnet and Schwaber, 2004].

Why we choose software development in Open Source? Based on the preliminary survey we found most of open source practitioners do not fully use OSS tools throughout the processes of software development. For example in planning phase only 8.3% of respondents use OSS tools, in design phase only 16.7% and in testing phase only 8.3% use OSS tools. They prefer to use OSS tools for web server, programming language and database, but do not prefer to use OSS tools in planning phase, designing phase and testing phase. (Refer to section 4, Result and Discussion for more details.)

The main motivation of this paper is to formulate a Knowledge Management System Framework for Open Source in Software Development. The function of this framework is to enable the CoP to share the knowledge of Open Source especially in Software Development. The proposed framework is based on Hahn and Subramani (2000), Meso and Smith (2000) Tiwana's KMS Architecture (2002), Abdullah, Sahibuddin, Alias, Selamat (2006) and Shankar, Acharia, Baveja (2009) and consist of five main parts; layers, functionality, process, knowledge and Community-of-Practice.

The strategies of managing knowledge according to ontology give the opportunity to CoP members to be knowledgeable in all facets of System Development Life Cycle. Knowledge sharing and collaboration among the CoP create the knowledge networks and can be considered to be part of the third generation knowledge management initiatives. The benefits of the above framework is its ability to provide a mechanism to share the knowledge throughout the development processes, from the planning phase until the maintenance phase in terms of OSS tools, experience, best practice cases, research, documentation, source code, support and many others. Why users should change his/her mind to use the OSS? There are many reasons why the open source model has been successful and popular with developers and the benefits for the users/organizational in terms of Access to source code, broad rights, encourage software re-use, can increase code quality and security, decrease vendor lock-in, reduce cost of acquisition, increase customizability and community [Olla and Crider, 2006]. The other of the benefits of open source first is the low cost or free access to knowledge, technologies, and/or other product, second is a counter to quality issue and finally the initiative cut across organizational boundaries and improve and/or create communication paths and protocol leading to improved cooperation [Jennex, 2006].

The remainder of the paper is structured as follows: Section 2 discusses related works. Section 3 reviews the methodology, and Section 4 presents result and discussion, and finally the conclusion is presented in Section 5.

2. Related Works

In this section, the discussion shall focus on Software Development and Existing KMS Frameworks. The Software Development is focus on the phases in software development, while Existing KMS Framework is describing and evaluating five KMS frameworks. The details of software development and existing KMS framework are in section 2.1 and 2.2 respectively.

2.1 Software Development (SD)

Software engineering knowledge are acquired and generated in order to produce high quality software, shorter development period and lower development costs. There are many software engineering methods used in industry. Among the well known methodologies are SDLC, Rapid Application Development (RAD), Agile Methods, OOAD and RUP. In this paper, we choose SDLC starting from planning, Analysis, design, implementation and finally to maintenance [Hoffer, George, Valacich, 2005](refer to Figure 1), the activities for each phase of SDLC are shown in table 1.

2.2 Existing KMS Frameworks

Five existing KMS frameworks were evaluated and analyzed, as follows:

- KMS architecture by Meso and Smith (2000).
- Framework for KM Support by Hahn and Subramani (2000).
- Tiwana's KMS Architecture (2002).
- KMS framework for Higher Learning Institution (HLI) by Abdullah, Sahibuddin, Alias and Selamat (2006).
- Soft-system Knowledge management framework by Shankar, Acharia and Baveja (2009).

Meso and Smith (2000) proposed a KMS architecture which comprises of technology, function and knowledge (see Figure 2). They proposed that the technology should include computer-mediated collaboration, electronic task management, and messaging. The functions utilize KM processes in using, finding, creating and packaging knowledge. Finally, the knowledge content includes how to answer the questions such as the know-how, know-what, know-why and so on.

Hahn and Subramani (2000) proposed a framework for KM Support (see Figure 3) which suggests two important considerations in managing knowledge, as follows:

- Where the knowledge resides and
- The extent to how the knowledge is structured.

The locus of the knowledge determines how the KMS connects a user (with a problem or question regarding an artifact) to an expert. Meanwhile, the level of a priori structure determines the extent to which KMS usage imposes the burden of a translation (or transformation of the problem or question) to a form that corresponds to implicit logic underlying the priori structure.

Tiwana (2002) has proposed KMS architecture consists of seven layers, which are interface, access, collaborative, application, transport, integration and repositories.

Meanwhile, Abdullah, Sahibuddin, Alias and Selamat (2006) proposed a KMS framework for HLI which includes 6 components: Architecture, Functionality and Application, Taxonomy and Processes, Culture, Psychological and Audit, as depicted in figure 4.

Likewise, Shankar, Acharia and Baveja (2009) proposed a Soft-System Knowledge Management framework (see figure 5), which propose an approach for KM system development to ensure a fit between the organizational needs on new product development (NPD) and KM initiatives. This framework consists of three main components, as follows:

- Knowledge Sharing Methods
- Organizational Level
- Key Enablers

The comparison on the major components in all of the above frameworks is depicted in Table 2. In summary, the existing frameworks overlaps on some general components such as infrastructure, technology, repository and collaboration. However, these combined frameworks are still not sufficient to support the Open Source Software Knowledge Management System (OSSKMS) requirements.

3. Methodology

There are three steps taken in the methodology, first conducted a survey and analyzed, second analyzed KMS framework and lastly the process of the formulation of the proposed framework. We choose this method because we stated with identified the problems and then analyzed several KMS frameworks as a reference to formulate a proposed framework. The formulation of the proposed framework is take into account the problems identified from the survey result and existing KMS framework. This is very important to make sure the proposed framework is based on the objective and the limitation.

A survey was conducted in selected software development organizations in Klang Valley Malaysia to gauge the developer's awareness of OSS and to ascertain the usage of OSS tools in OSS development. Fifty questionnaires were distributed to the respondents, but only twelve questionnaires were returned. The questionnaire data were verified and was analyzed using Rasch Method and SPSS. The result of the survey contributed to the formulation of the proposed framework. The main components derived from the questionnaire are open source infrastructure, knowledge and CoP. Finally we analyze five existing KMS frameworks and formulate a proposed KMS framework. The details of the formulation of the framework is discussed in section 3.1.

3.1 Formulation of the Framework

A new framework shall be synthesized from previous frameworks by Hahn and Subramani (2000), Meso and Smith (2000), Tiwana's architecture (2002), Abdullah *et al.*(2006), and R. Shankar *et. al.*(2009) as follows:

- Structured knowledge, which includes document repository, data warehousing, databases, repositories and data mining are derived from all from frameworks. This knowledge shall be mapped as OSS knowledge onto each phase of software development.
- The collaborative environment elements are derived from Meso and Smith (2000), Hahn and Subramani (2000), Tiwana (2002), and Abdullah *et. al.*(2006).
- Functions or process are derived from Meso and Smith (2000) and Abdullah *et. al.*(2006).
- Knowledge are derived from Meso and Smith (2000), and Shankar *et. al.* (2009).

The result of proposed KMSOS²oD Framework is shown in Figure 6. The proposed framework is formulated especially to support SDLC using OSS and consists of five main components: layers, components, process, knowledge, and CoP (see Figure 6).

The layers consists of application, collaboration tools, open source infrastructure, technologies, and repositories.

Meanwhile, the functionality layer includes portal, collaboration tools, OS infrastructure, technologies, and repositories. Basically the OS infrastructures are to act as a bridge between repositories, technologies, collaboration tools and application to CoP.

The Processes are similar to KM processes for acquiring or collecting, organizing, disseminating or sharing knowledge, using knowledge and store the knowledge. The knowledge content focuses on all facets of SDLC, starting from planning, analysis, design, implementation and finally maintenance. There are two categories of knowledge tacit and explicit. Based on the SECI model by Nonaka and Takeuchi (1995), there are four conversion patterns of knowledge are Socialization, Internalization, Combination, and Externalization. Finally, the CoP consist of students, academicians, programmers, system analyst and users.

OSS, Software Development (SD) and Knowledge Management (KM) are the main conceptual foundation for the proposed framework. These three foundations are collaborating among each other during KM activities. The OSS will provide the knowledge taxonomy as knowledge content either as a tacit or explicit knowledge. This knowledge will be map and manage into SDLC phases and activities. Finally, KM will act as tools for capturing, storing, organizing, and making accessible knowledge and expertise within and across communities.

How the proposed framework can support the knowledge and use OSS tools? Basically, in this case the CoP can only disseminate/acquire the OSS knowledge and mapped into software development process either in planning, analysis, design, implementation or maintenance through the portal. In this case, software development process refers to SECI model. All the OSS knowledge will be stored according to the ontology.

4. Result and Discussion

Figure 7 shows the summary of Person and Item data, where the Persons are (developers) and Items are the OSS tools in software development. For person reliability, the Cronbach Alpha of 0.87 logit and the item reliability of 0.91 logit are good and very good (Fisher, 2007), respectively. The persons summary indicates a good separation of $G=1.05$ logit and good item spread of 2.54 logit. Meanwhile, the item summary reveals a good separation of $G=2.81$ logit and but a poor item spread of 8.28 logit. The poor item spread (2.54 logit) could be contributed by item Q5, Q11, Q6, and Q7.

Based on Person-Item MAP (Figure 10), the two person scoring the highest point are E2134 and L2132. Both of them score 3.06 logit. Both of them are male and their age group is between 31 to 35 years old. There are one person (K4124) score the lowers point. For items, the most difficult item is Q18, measured at 4.17 logit. This means 25% strongly do not agree and 75% do not agree that products of OSS are very hard to commercialize. Nevertheless, the easier item is Q5 and the min measure is -4.09 logit, where 8% agree and 92% strongly agree that OSS tools are easy get in the market / in the Internet. There is a big gap between item Q18 and Q15 and Q12. Q15 is about OSS tools are difficult to learn and Q12 is about OSS tools are suitable for non-profit company. For both questions, most of respondents are not agree that OSS tools are difficult to learn and suitable for non-profit company. Besides that, there are also exist big gap below the item mean, there are Q5 and Q11, Q6, Q7, Q13 and Q10 with contributes a poor item spread of 8.28 logit. Q11 is about OSS tools should be taught in the Higher Learning Institution in Malaysia, Q6 is regarding OSS tools are totally reducing cost of software development, Q7 is about OSS tools are always available in the internet, Q13 is about OSS tools should be transfer to knowledge for sharing purpose and Q10 is about companies/organizations should use OSS tools in software development. Based on the responds from the respondents, according to items Q11, Q6, Q7, Q13 and Q10, most respondents agree to the questions asked.

Nevertheless, 91.7% (see Figure 8) of the OSS developers do not use OSS tools in planning phase. Surprisingly, 83.3% did not use OSS tools in design phase. This situation same in the testing phase, where 91.7% did not use OSS tools in testing.

Figures 9 show that the OSS developers fully utilized OSS tools in Operating System, database, programming language and web server. Linux (75%) is the popular operating system compared to the others. MySQL (75%) is the most popular database and PHP (75%) is the most well known programming language. While, Apache (83.3%) is the highest usage for web server.

As a conclusion, usage of OSS tools in software development in planning, design and testing phases are very low. However, the majority of OSS developers fully utilize OSS tools in database, programming language and web server.

5. Conclusion

Knowledge sharing and collaboration among the community-of-practice create the knowledge networks and can be considered to be part of the third generation knowledge management initiatives. In a survey carried out to gauge on usage of OSS tools in OSS development reveals that OSS developers not fully utilized OSS tools throughout the system SDLC. Hence, information and knowledge passed down from various stages of SDLC are not captured nor flow smoothly between stages. As such, we propose the KMSOS2oD framework to provide a mechanism to share the knowledge throughout the development processes, from the planning phase until the maintenance phase in terms of OSS tools,

experience, best practice cases, research, documentation, source code, support and many others. This framework attempts to adapt the software development life cycle to develop application system. This framework gives the opportunities and guidelines for CoP to share their knowledge and encourage people to use OSS in software development throughout the development phase.

References

- Abdullah, R, Sahibuddin, S, Alias, R. A., & Selamat, M.H. (2006). Knowledge Management System Architecture For Organizational Learning With Collaborative Environment, *International Journal of Computer Science and Network Security*, 6(3A), pp. 237-246.
- Barnett, L., & Schwaber, C.E. (2004). Applying Open Source Process in Corporate Development Organizations, Forrester Research, Inc, Technology Square Cambridge, MA, USA..
- Bollinger, A. S., & Smith, R.D. (2001). Managing Organizational Knowledge as a Strategic Asset, *Journal of Knowledge Management*, 5(1), pp. 8-18.
- Canfora, G, Cimitile, A., & Visaggio, C. A. (2002). From Knowledge Management Concepts Towards Software Engineering. Product Focused Software Process Improvement, 4th International Conference PROFES 2002, Rovaniemi, Finland, December 9-11, 2002, pp. 407-422.
- Davenport, T.H., & Prusak, L. (1998). *Working Knowledge: How Organisations Manage What They Know*, Boston, MA: Harvard Business School Press.
- Feller, J., & Fitzgerald, B. (2000). A Framework Analysis of The Open Source Development Paradigm, Proceedings of the twenty first international conference on Information systems, December 2000, ACM, Brisbane, pp. 58-69.
- Fisher, W.P. (2007). Rasch Measurement Transactions. [Online] Available:<http://www.rasch.org/rmt/rmt211m.htm>(8 April 8, 2009)
- Godfrey, M.W., & Tu, Q. (2001). Evolution in Open Source Software: A Case Study, University of Waterloo. [Online] Available: <http://plg.uwaterloo.ca/~migod/papers/2001/iwpse01.pdf> (April 8, 2009)
- Hahn, J., & Subramani, M. (2000). A Framework of Knowledge Management System: Issues and Challenges For Theory and Practice, Proceedings of the twenty first international conference on Information systems, ACM, Brisbane.
- Hoffer, J. A., George, J. F., & Valacich, J. S. (2005). *Modern system Analysis and Design: Fourth Edition*, Pearson Prentice Hall, New Jersey.
- Jennex, M. E. (2006). Open Source Knowledge Management. [Online] Available: [http://www.igi-global.com/files/prefaces/IJKM%20preface%20\(4\).pdf](http://www.igi-global.com/files/prefaces/IJKM%20preface%20(4).pdf) (8 April 2009)
- Meso, P., & Smith, R. (2000). A Resource-Based View of Organizational Knowledge Management Systems, *Journal of Knowledge Management*, 4(3), pp.224-234.
- Mettam, G. R., & Adams, L. B. (1994). How to prepare an electronic version of your article. In B. S. Jones, & R. Z. Smith (Eds.), *Introduction to the electronic age* (pp. 281-304). New York: E-Publishing Inc.
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-creating Company*. New York, Oxford: Oxford University Press.
- Olla, P., & Crider, R. (2006). Open-Source Online Knowledge Portals for Education”: Benefits and Risks. In A. Tatnal, *Encyclopaedia of Portal Technology and Applications*. IDEA. In pres
- Shankar, R., Acharia, S., & Baveja, A. (2009). Soft-system Knowledge Management Framework for New Product Development, *Journal of Knowledge Management*, 13(1) , pp.135-153.
- Tiwana, A, (2002). *The Knowledge Management Toolkit: Orchestrating IT, Strategy and Knowledge Platforms*. USA: Prentice Hall.

Figures and Tables

Table 1. Phases and activities of SDLC (SD using OSS)

Phases	Activities
Planning	<ul style="list-style-type: none"> • Project Identification and Selection • Project Initiation and Planning
Analysis	<ul style="list-style-type: none"> • Requirements Determination • Requirements Structuring
Design	<ul style="list-style-type: none"> • Files and Databases • Forms and Reports • Interfaces and Dialogues • System and Program Structure
Implementation	<ul style="list-style-type: none"> • Distributed Systems • Coding • Testing • Installation • Documentation • Training • Support
Maintenance	<ul style="list-style-type: none"> • Obtaining Maintenance Requests • Transforming Requests into Changes • Designing Changes • Implementing Changes

Table 2. Comparisons Between the Frameworks

Item/ Framework	Knowledge Management Support (Hahn and Subramani, 2002)	KMS framework for HLI (Abdullah, 2003)	KMS Architecture (Meso & Smith, 2000)	TIWANA'S KMS Architecture (2002)	Soft-system Knowledge Management framework (R. Shankar, 2009)	KMSOS ^o D (Suggestion)
Application	NA	Y	NA	Y	Y	Y
Technology	Y	Y	Y	Y	NA	Y
Infrastructure	Y	Y	Y	Y	Y	Y
Repositories	Y	Y	NA	Y	Y	Y
Knowledge	NA	NA	Y	NA	Y	Y
Functionality/KM	NA	Y	Y	NA	NA	Y
Process						
Collaboration	Y	Y	Y	Y	NA	Y
Tools						
Open Source	NA	NA	NA	NA	NA	Y
Infrastructure						

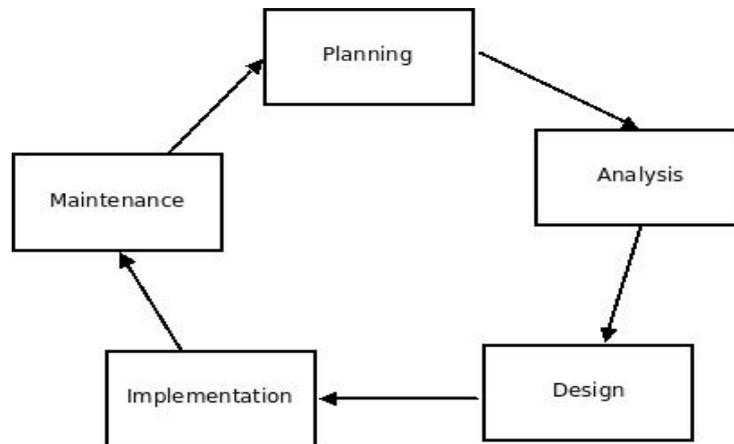


Figure 1. The System Development Life Cycle

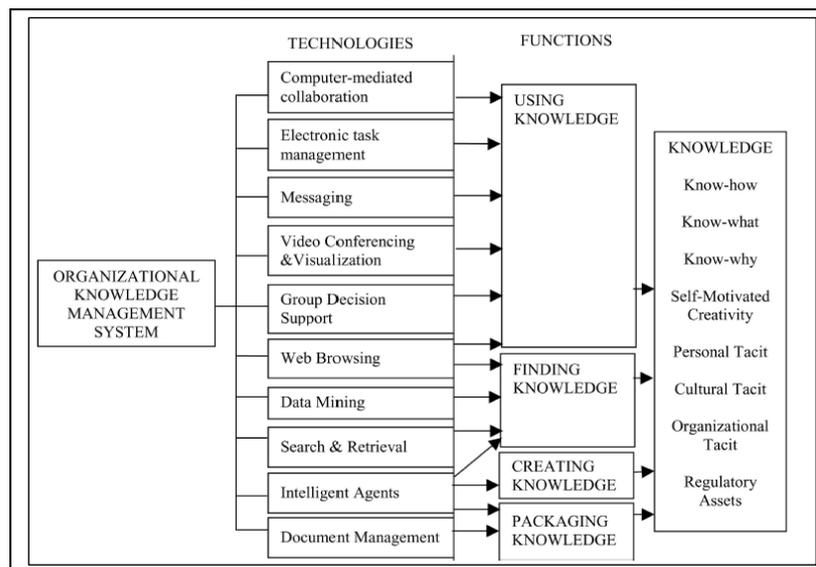


Figure 2. The Technical Perspective of a KMS

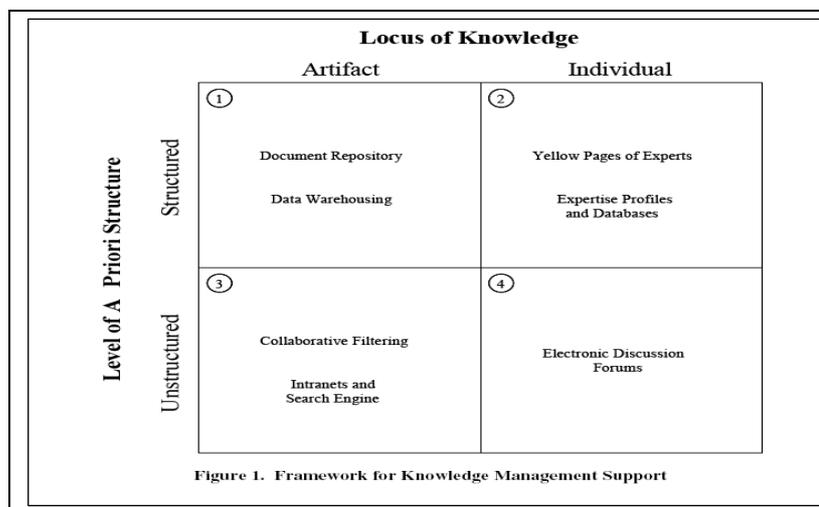


Figure 1. Framework for Knowledge Management Support

Figure 3. Framework for Knowledge Management Support

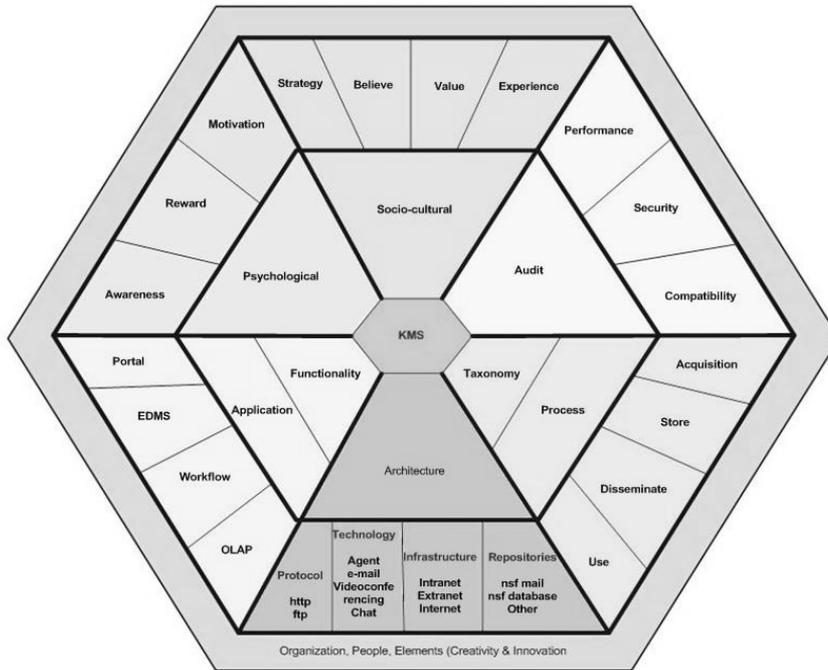


Figure 4. KMS framework for HLI

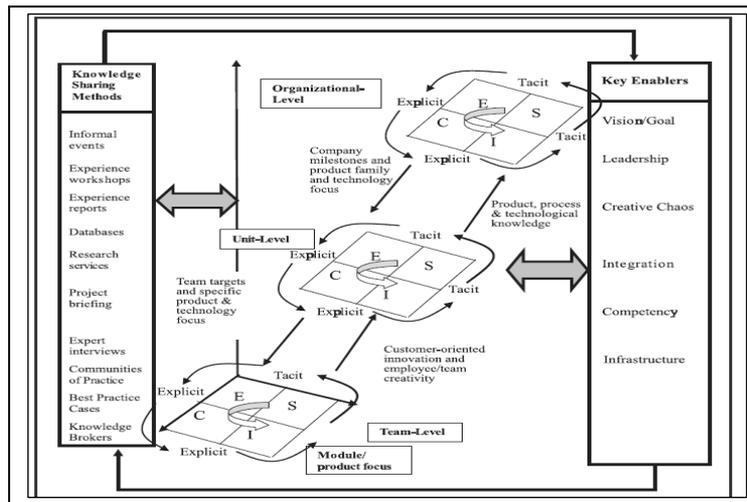


Figure 5. Soft-system Knowledge Management Framework

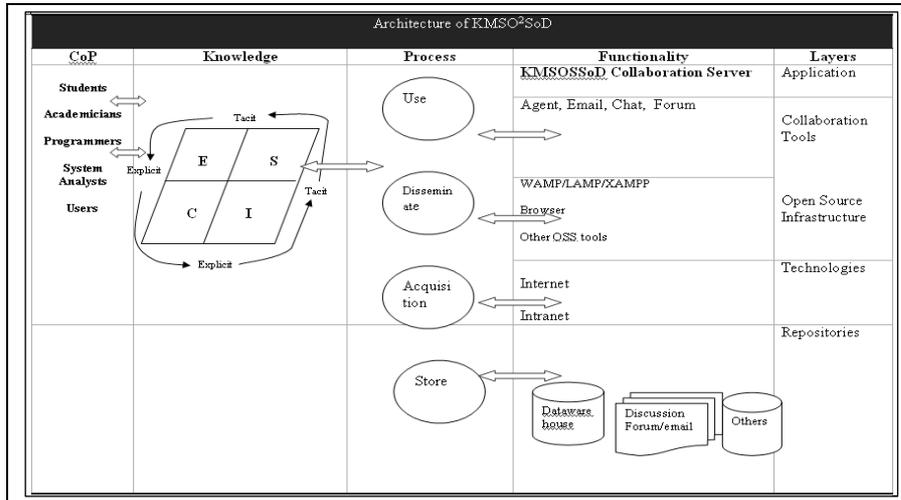


Figure 6. Proposed KMSOS²oD Framework

SUMMARY OF 12 MEASURED Persons									
	RAW SCORE	COUNT	MEASURE	MODEL ERROR	MNSQ	INFIT ZSTD	MNSQ	OUTFIT ZSTD	
MEAN	59.9	19.7	1.79	.44	1.02	-.3	1.01	-.4	
S.D.	5.7	1.1	.73	.02	.76	2.0	.67	1.7	
MAX.	67.0	20.0	3.06	.47	2.84	3.9	2.33	2.2	
MIN.	44.0	16.0	.52	.42	.23	-3.3	.22	-3.1	
REAL RMSE	.51	ADJ. SD	.53	SEPARATION	1.05	Person RELIABILITY	.52		
MODEL RMSE	.43	ADJ. SD	.58	SEPARATION	1.31	Person RELIABILITY	.63		
S.E. OF Person MEAN = .22									
VALID RESPONSES: 98.3%									
Person RAW SCORE-TO-MEASURE CORRELATION = .90 (approximate due to missing data)									
CRONBACH ALPHA (KR-20) Person RAW SCORE RELIABILITY = .87 (approximate due to missing data)									
SUMMARY OF 20 MEASURED Items									
	RAW SCORE	COUNT	MEASURE	MODEL ERROR	MNSQ	INFIT ZSTD	MNSQ	OUTFIT ZSTD	
MEAN	36.0	11.8	.00	.59	.98	-.3	1.01	-.2	
S.D.	6.7	4	1.99	.12	.73	1.5	.75	1.5	
MAX.	47.0	12.0	4.17	1.06	3.28	3.4	3.12	3.2	
MIN.	21.0	11.0	-4.09	1.49	.20	-2.6	.18	-2.7	
REAL RMSE	.67	ADJ. SD	1.87	SEPARATION	2.81	Item RELIABILITY	.89		
MODEL RMSE	.61	ADJ. SD	1.89	SEPARATION	3.13	Item RELIABILITY	.91		
S.E. OF Item MEAN = .46									
UMEAN=.000 USCALE=1.000									
Item RAW SCORE-TO-MEASURE CORRELATION = -.99 (approximate due to missing data)									
236 DATA POINTS, APPROXIMATE LOG-LIKELIHOOD CHI-SQUARE: 339.34									

Figure 7. Reliability of Person-Item

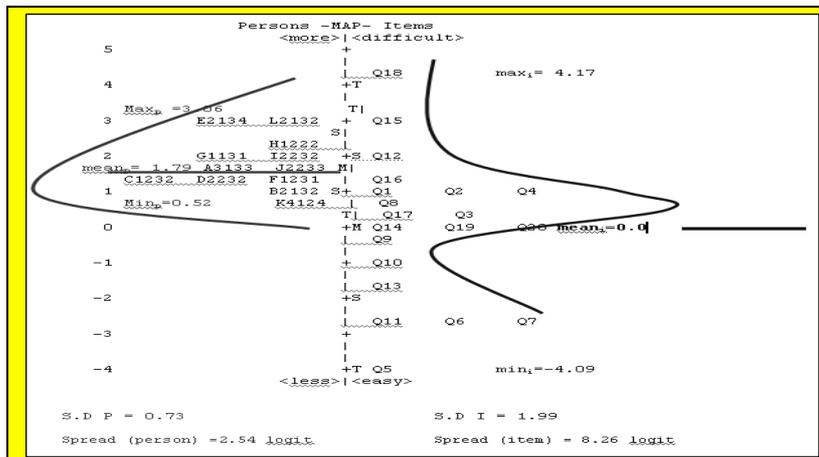


Figure 8. Person-Item MAP

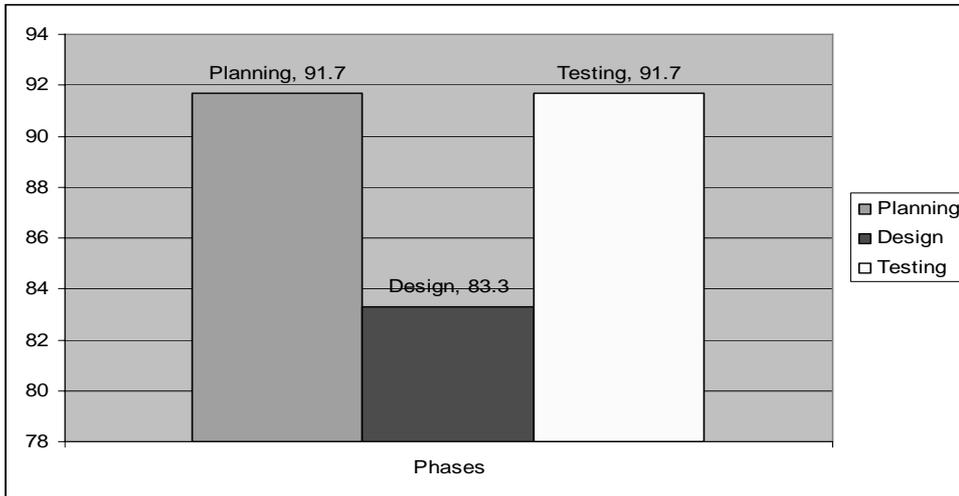


Figure 9. Did not used OSS tools

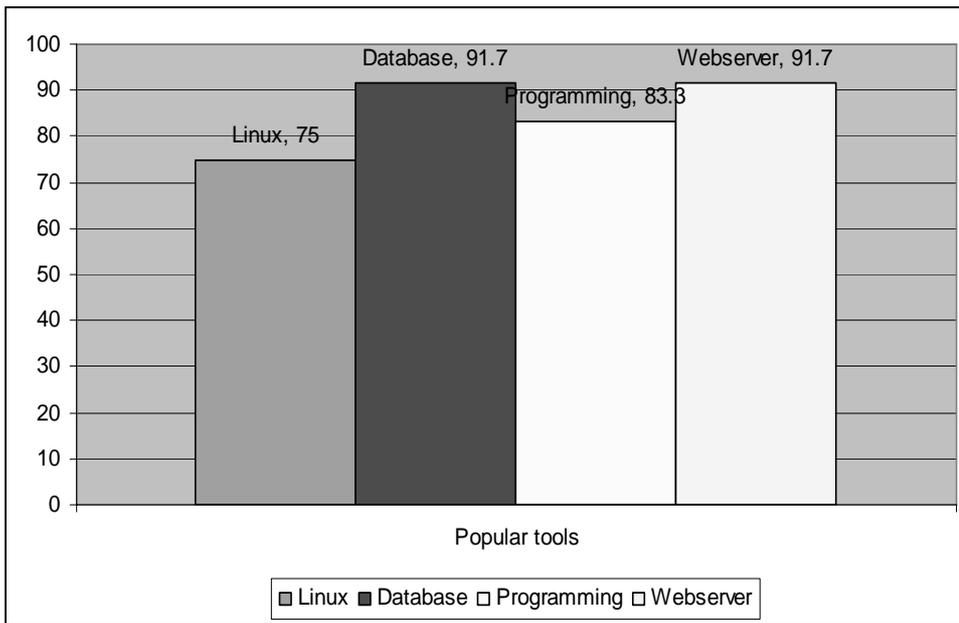


Figure 10. Popular OSS tools