Research Journal of Applied Sciences, Engineering and Technology 14(12): 450-457, 2017
DOI:10.19026/rjaset.14.5147
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 17, 2017 Accepted: June 7, 2017 Published: December 15, 2017

Research Article

Conceptual Framework for Knowledge Flow in Software Development Process

Marzanah A. Jabar, Mustafa S. Khalefa, Rusli Hj. Abdulla and Salfarina Abdulla
Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM), 43400 Selangor, Serdang, Malaysia

Abstract: The main aim of this study to merge knowledge flow activities with software development process Past models of flow of knowledge lacked in terms of software development and lacked on sharing right knowledge and lacked in-depth exploration of context modeling, which made those models less applicable. In addition now it attracts much consideration, attention in the knowledge management field. In a knowledge-based organization, knowledge workers need to acquire a variety of knowledge (information) about their tasks. Therefore; many organizations have built knowledge support plat forms to assist workers in meeting their knowledge-needs. These platforms help workers to identify and share knowledge in order to speed up organization innovation and improve employee Productivity. This research proposes a conceptual framework of context oriented flow of knowledge in software development aspects. In this framework, the context is seen as an inseparable element of flow of knowledge, which is regarding the creation, transformation as well as application of knowledge items. One of the main challenges is how to exploring knowledge flow, sharing knowledge in software development process. The main goal for this research to provide a framework to solve this challenge. In this research used questionnaire instrument with 21 people as they are from software development process environment, also in this study want to improve all factors, task, type of knowledge, nature of knowledge used in software development process during the flow of knowledge. Finally in this study validate the framework using experts review, 4 experts from academic and industrial result was great shows all phases in software development process and factors when active, moreover in this research verified the component by using cronpach alpha methods the result was good in acceptable rang 0.711.

Keywords: Artifacts, knowledge flow, knowledge flow element, knowledge flow with software process, roles, software development process

INTRODUCTION

Establish the context of the work being reported. This is accomplished by discussing the relevant primary research literature (with citations) and summarizing the problem you are investigating;

Different definitions of KF have been given by different scholars. Zhuge (2006a) defined knowledge flow as a process of sharing knowledge among individuals or the exchange of instruments used in processing knowledge and alluded to the positive different peculiarities of knowledge flow including direction, content and carrier (Zhuge, 2002). Zhang and Li (2005) described knowledge flow as a technique of creating, sharing and categorizing knowledge among different individuals. There is a great need for knowledge flow in development (Zhang and Li, 2005). In Zhuge (2002), Zhuge (2006a) and Zhuge and Guo (2007) the examination of knowledge flow is centered on collection, transmission and impartation of knowledge within a group. In regards to workflow, work knowledge can be transmitted to workers. Thus, in workflow, level of knowledge cooperation between workers and process is considered a factor influencing the proficiency of workflow. A group which develops software can put together knowledge from one of the members of the group and then pass it on to others. Knowledge partnership is a major way of enhancing teamwork efficiency and also a way to achieving goals easier and faster (Zhuge, 2002; Zhuge, 2006b) suggest an example based strategy in which the merging of codification and personalization could be used derive an efficient knowledge flow system. A knowledge workflow system was designed by Sarnikar and Zhao (2007, 2008) with aim of supporting the computerization of knowledge flow system as well as to mechanize knowledge flow systems by fusing workflow and knowledge. According to Rodríguez

Corresponding Author: Marzanah A. Jabar, Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM), 43400 Selangor, Serdang, Malaysia, Tel.: 0060176484073

et al. (2004) KFs can be used by members belonging to same communities of practice to exchange data and experience in certain areas which can support their work and help them finish their work faster. KFs can make knowledge sharing simple as well as enhance the storage of knowledge for research purpose. KFs can be in form of reference systems which supports knowledge exchange between analyst and researchers.

Recently, there has been a number of KF models proposal. One of such is TKF (Textual Knowledge Flow) proposed by Luo *et al*. (2008) which is a system meant to support the flow of textual knowledge in the area of semantics; a semantic connection system. This kind of proposed knowledge flow system can enable and help the assessment of profits and inputs by clients. Furthermore, KFs can be used for categorizing knowledge needs and knowledge reference patterns which can be easily accessed by specialist when performing a task. Lai and Liu (2009) identified a KF model which can meet the knowledge needs of labourers; workers who use this, will be able obtain knowledge which can satisfy their need for knowledge the knowledge flow found from record access logs. Another system that can be considered a KF is weblog referencing request which is identified as a sender-message-beneficiary or even receiver because sometimes weblogs contain hyperlinks of other weblog post. (Anjewierden *et al*., 2005; Kim *et al*., 2003) suggested a KF system that utilizes a technique to gather, archive and share knowledge. Zhang *et al*. (2008) employed the use of Petri-net to design a KF which can serve as a reservoir which supports sharing, learning, workflow, comprehension and combination of knowledge according to four kinds of flow relations: knowledge creation, combination, reproduction and sharing. Zhao and Dai (2008) showed a method which combines business process and knowledge flow through a technique that incorporates business procedure and Kfs by dividing Kfs into consecution, distribution, blend and reflection toward oneself. The main objective of this study to provide comprehensive conceptual framework and clear model to link knowledge flow activities with software development process.

THEORETICAL BACKGROUND

Knowledge flow is unseen; it is still relevant whether people intentionally or unintentionally make use of it. Work collaboration among a group can be visualized as an accompanying problem of KF. Members of a team can create knowledge and afterwards share it among themselves through the use of a knowledge flow system (KFN). The efficiency of teamwork can be enhanced by proper network planning and implementation of KF (Zhuge, 2002). The major advantages of KF are the cancelation of unnecessary knowledge sharing among team members because different members of the team seek different knowledge to perform certain tasks and less time and effort investment (Lin *et al*., 2012).

Zhuge (2002) shows and improve knowledge flow as a process which involves an exporter and receptor of knowledge; the receptor uses the knowledge to enhance the organization general performance. This could be described as a knowledge-sharing technique which enables knowledge sharing between individuals or even knowledge-processing techniques. Sarnikar and Zhao (2007) and Zhuge (2002) state that, any recipient of knowledge can be described as a "knowledge node". Direction, content and a carrier are the three main components of knowledge flow which determines who sends the message, who receives it, the knowledge content and the way the content is conveyed respectively. The knowledge node might be a member of the team or part of a knowledge process (Sarnikar and Zhao, 2007).

An improvement in organizational knowledge and business processes are evidences of KF optimization. Distinguishing and mapping the KF are relevant to an organization in three real areas (Yoo *et al*., 2007):

- Knowledge flow enhances the transfer of skill that has been created in a sub unit of an organization to other areas within the organization.
- Knowledge flow aids in organizing different workflows of sub units which are scattered geographically.
- Knowledge flow assists organizations to successfully handle business activities that require cooperation of the organization's various sub units.

A clear picture of how knowledge flows across the structure of an organization alongside the features of the workflows. According to Zhuge (2002), KF is a technique of knowledge invention, sharing and replication from a source which is often known as the sender to the recipient. Basically the argument here is that in knowledge flow there is always a sender and a recipient; knowledge flow is not just about knowledge content and direction.

KNOWLEDGE FLOWS ELEMENT

Knowledge nodes: knowledge Nodes (KNs). are known to be sources of knowledge for both the sender and the recipient with recipient. A KN is possibly communicated to a team member or an agent who can create, process and share the knowledge. Senders and recipients of knowledge are also considered as the direction of knowledge flows. In other words, the flow of knowledge can result from the trigger of a source which is known as push strategy. It can also be triggered by a knowledge request made by the recipient; this is known as the pull strategy (Jarrahi and Kangavari, 2012; Zhuge, 2006a). Figure 1 show is that.

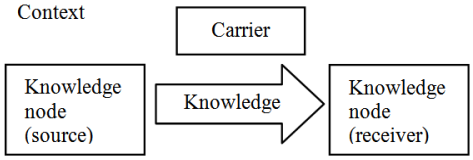

Fig. 1: Element of knowledge flow

Knowledge: The central element is knowledge that reveals specific and sharable knowledge contents (Schutte and Snyman, 2006; Guo and Wang, 2008; Zhuge, 2006b). The knowledge in its pure form-before it is translated and changed to facilitate easy transfer and exchange-as it exists within the creator or knowledge source. Knowledge is cyclic; when it is subjected to the activities listed below it evolves and changes. Different labels are used by different authors to point out these events, processes or activities but usually they consist of the following (Bouthillier and Shearer, 2002; Breedt, 2000; Probst *et al*., 2000; Reinhardt, 2002).

Knowledge creation, identification or discovery: this process involves the creation of new knowledge or identifying existing knowledge to have potential value, Knowledge acquisition, collection or capture: has to do with obtaining the knowledge identified in the previous phase, Knowledge processing, filtering or adaptation: after knowledge is obtained, it is transformed or changed base on the need of the recipients, Knowledge utilization or application: for knowledge to be valuable, it must be used, Knowledge storage or retention: when knowledge is used it becomes part of the user's knowledge base and expertise.

Carrier: The carrier in knowledge flow refers to the medium through which knowledge, which is the content, is shared (Zhuge, 2006a). The medium could be local network, internet or magnetic tapes.

Context: Context refers to the location in which the KF process is carried out. Without a shared context, KNs cannot occur because there won't be shared meaning. This means that for a knowledge flow to be successful there must be a mutual understanding between the source and the recipient; this will in turn yield an understanding of the shared knowledge (Schutte and Snyman, 2006; Guo and Wang, 2008).

INCORPORATING KNOWLEDGE FLOW INTO THE SOFTWARE DEVELOPMENT PROCESS

The following reasons are given by Zhuge (2002), Zhuge (2006b) and Zhuge *et al*. (2001) as the reasons why KF is incorporated into the distributed team Software Development Process (SDP)

The SDP is a Cognitive procedure in which group members can forward their work not just through programming and related software tools but through cognitive operation among themselves.

It is impossible to predesign Cognitive cooperation because it is during the process of development that knowledge of team members is collected and built in the form of experience, methods, decisions and software improvement aptitudes is produced and gathered during the development procedure. Therefore, for dynamic reflection of cognitive process KF is required. An improvement cooperation team can be supported by experience, skill accumulation mechanism as well as aggregate knowledge obtained from past project. Therefore, it is possible for the team members to adapt to change rather than have a redundant.

The development team followed a disciplined software process based on the Unified Process for EDUcation (UPEDU, 2001; Gendreau and Robillard, 2013). Figure 2 depicts a generic process practice. A role is responsible for the outcomes of an activity. An activity needs at least one artifact as an input and will generate at least one output artifact.

METHODOLOGY

The study was conducted in 2013-2017 in Malaysia.

The research strategy for the conceptual framework to make collaborative between knowledge flow and software development process to improve the flow component during software development process. After reviewing the existing various KF framework in SDP area, the following aspect components were identified for the knowledge flow in SDP such as sender, receiver, context, carrier. After that start to develop new questionnaire match for this study, however we send it this questionnaire to the experts to evaluate it before distributed then we start to allot the questionnaire to SDP domain we send it to the 30 person we get complete answer from 21.in addition we need expert's assistance to read the proposed system description To verify the feasibility and applicability of the proposed framework processes, To verify the validity of the proposed framework and model in term of understandability, comprehensive. We used 4 experts from university and industrial.

In the final stage in methodology we start to analyze this pilot study Analysis are mainly conducted using cronbach alph to check the reliability for the element in the questionnaire. In addition we use this measurement to analyzing the respondents and questionnaire items. The main objective purpose of this questionnaire is generally to gain understanding of how knowledge flow clarify or unfold between SDP phases (Fig. 2). The focus will be given on The nature of knowledge flow during SDP phases, The utilization of knowledge flow in SDP, shows how the flow is happened in software development process domain, The mediums used for knowledge flow, The input and output knowledge in SDP, The role of knowledge flow in SDP, The type of knowledge flow in SDP, The task

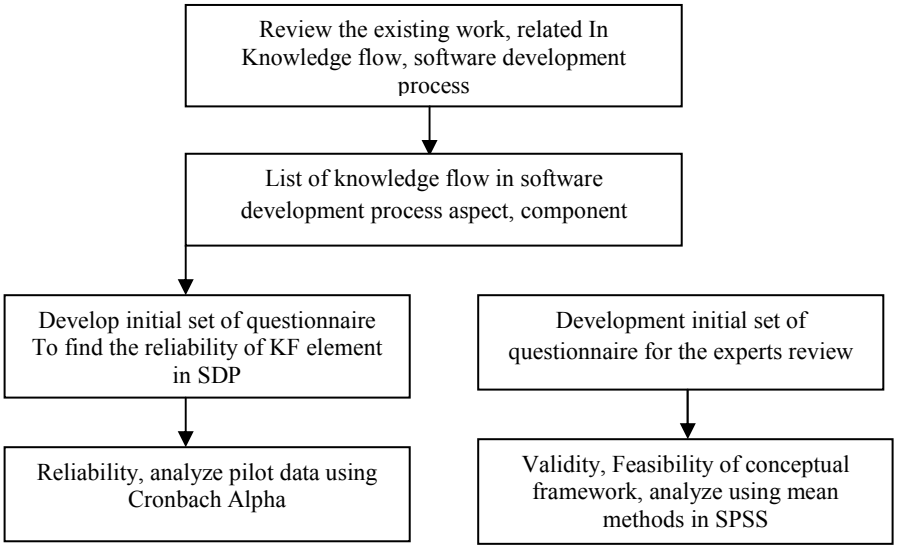

Fig. 2: Methodology

or activities in SDP during the knowledge flow, The sender and receiver in SDP during the flow of knowledge.

The survey that we used in this research not adapted its new survey because our research as exploratory research.

PROPOSE CONCEPTUAL FRAMEWORK

People: Who those they are affected and they responsible to send and receive knowledge who have the knowledge can use it during the flow in SDP. People knowledge on the other hand, accounts the knowledge about leadership, teamwork, communication, negotiation, accepting direction, mentoring and consulting (Bass *et al*., 2008).

Knowledge: In this framework considered the nature of the knowledge are tacit, explicit, information and data. In addition, considered the type of knowledge during SDP are user requirement knowledge, functional domain knowledge, project status knowledge, project management experience knowledge, technical, architecture knowledge, business knowledge, system vision/mission, business rules, quality goal.

Software development process (SDP) will explain the phases and what each phase has and include the six main phases of SDP are considered to be Planning, Analysis, Design, Coding, Testing, Maintenance.

Task: Description of task: Includes description of the steps to be performed for a task, plus their pre-conditions and post condition of a task and checklist to ensure completeness of the task.

Artifacts: Knowledge artifacts are the memories, norms, values and other things that represent the inputs to and products of, the knowledge-enabled activities. When we speak of knowledge artifacts within the context of knowledge flows, we are actually making a simultaneous reference to two important entities. The first is the physical knowledge artifact, which serves as a representation of the associated cognitive knowledge artifact. The second is the cognitive knowledge artifact that makes up our awareness and understanding of a particular aspect of our real or meta-physical world.

Medium or channel: The carrier in knowledge flow refers to the medium through which knowledge, which is the content, is shared (Zhuge, 2006b). The medium could be local network, internet or magnetic tapes. That means the communication way for the knowledge transfer. Figure 3 is show the conceptual framework.

RESULTS AND DISCUSSION

After reviewing the existing various knowledge flow and software development process frameworks in SDP area, the following aspects and the respective components were identified. Required knowledge flow, such as knowledge sender (source), knowledge receiver, media or channel, knowledge, KF Activities are derived from (Zhang, 2002) To support these flow of knowledge during software development process.

This part of the research explains and makes an analysis of the answer obtained from the close ended questionnaire which was conducted among experts using excel tools. The findings indicate that the data obtained was from each question from the questionnaire and close ended questionnaire. This is the result for each question in the questionnaire.

Nature of knowledge: During this section we have some questions ask the expert people during the questionnaire with them. The first question explain

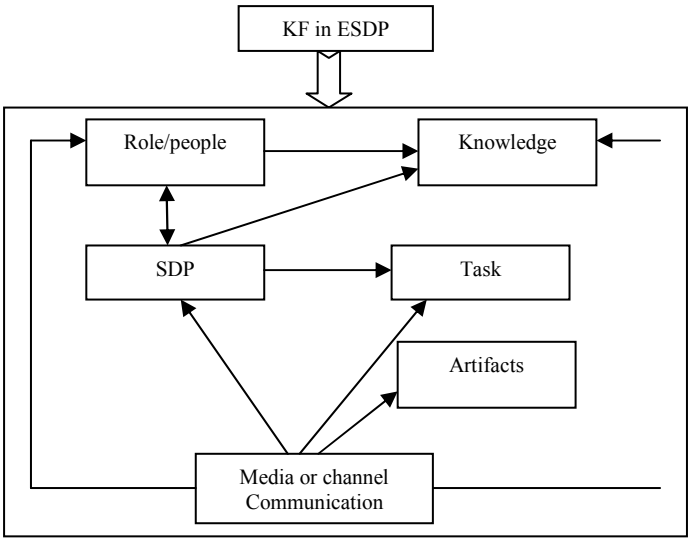

Fig. 3: conceptual framework for knowledge flow in software development process

about the nature of the knowledge inside each phase in enterprise software development process (tacit knowledge, explicit, information and data). The answer from this question. Shows that the most important, effecting phases during this process are analysis, testing, implementation. The other phases have minor and moderate effect. In addition the significant factors that affect each phases and high score during the analysis are explicit knowledge, Information, the others factors have minor and low effecting during all phases.

Media or channel: The media or channel we used to flow of knowledge during enterprise software development process and shows how knowledge transfer. During this phase it Got very high score and significant in design, analysis, testing phase depends on which factors very important and used during those phases. As well as the most important and effective factors during this phase are review meeting, email, workshop, face to face discussion, document preparation, some other factors they didn't effect in all phases like conferencing extranet, social network.

Input artifact: The most significant, active phases during this question (input artifact) and they got high score from the expert's design, testing, maintenance those phase they get top level. On the other hand the most important, significant factors during all phases in this question they are SRS, design document, test cases, Use case and user interface document, Data flow diagram DFD.

Output artifact: In this question the most important phases are analysis phase is got very high score during the output artifacts during SDP. The second important phase got significant in this question the testing got high score and the last phase get significant and high score is the planning phase. On the other hands the effecting factors during all those phases was important and effecting they are test result. Iteration plan, design document, components, meeting agenda, change record.

Task: The effecting of the task in SDP shows in the planning phase is the highest one during the task implementation, design and testing phases they are very significant in during the task and activities in enterprise software development process. As well as the analysis and implementation also they are got high score but less than the previous phase and the maintenance phase got low score during the task and activities in enterprise software development process. In addition the factors effecting during this question as the analysis result shows they analyze use case, design classes, manage product configuration, fix defects.

Roles: show all important people (factors) during each phase. The effecting people in charge during the process of enterprise software are significant and highly score get during the analysis in phases planning, maintenance and got moderate effecting during analysis phase. The main people (factors) involve and active during this phase software engineer, project manager.

Type of knowledge: The type of knowledge during software development process they are very important in this research and as one of the important factors depends of the answers from the questionnaire. The SDP phase the analysis and maintenance got high response and score during the answers from the experts that mean the knowledge during these two phases very important. The second phase got high score and important knowledge during those phases they are

Table 1: Reliability statistics

Cronbach's Alpha	N of Items
0.711	9

planning and testing phase very important type of knowledge implement during these two phases. As well as the important knowledge during all phase they are technical knowledge, quality goals, application domain, user requirement knowledge, functional domain knowledge.

Knowledge sender: During this question the most important people they send knowledge in software development process phases. The highest phase get score during this analysis result from the experts answer is analysis phase have significant and many people involved during this phase in this question. The second phase got high score from the experts answers are maintenance and design phase in this question. The important people (factors) effecting in this phases during the analysis the answers from the experts are project manager, IT manager, analyst. In the other hand the factors have some effecting but its minor effecting during this phases.

Knowledge receiver: The most important phase during the knowledge receiver and the significant roles during all phases are maintenance and design phase they got very high score from the results come from the experts. As well as two other phases they got moderate effecting on knowledge receiver (role/people) they are testing, implementation phase. In addition, the most significant factors were very active during those all phase for knowledge receive are project manager, software engineer, architect, analyst.

Test of Cronbach's alpha has been conducted to examine reliability of the questionnaire. Cronbach's alpha is a reliability test for questionnaire where the same set of variables would result to the same responses if the same set of questions (Bland and Altman, 1997). After performing this test, we received Cronbach's Alpha Based on Standardized Items of 0.711 or 71%. According to Cortina (1993), any score above 0.70 or 70% is an acceptable reliability coefficient score. The output of reliability analysis is shown in Table 1.

This section of the work validates the model. Here four experts were asked to validate the proposed knowledge flow framework in the process of enterprise software development. The objective of this study is to propose a new framework for knowledge flow which addresses and the issues regarding the nature of knowledge flow during phases of ESDP. The role people play in ESDP phases, kind of shared knowledge (low, medium or high) and types of activities that go on, the sender and recipient in ESDP during the flow of knowledge. In addition in this study we validate the conceptual and theoretical model during SDP. All answer come from the experts was positive with high rank will show that in descriptive analysis. We validate by 4 experts they have experience in academic and industrial as they are developer and programmer experts, they have at least 5 years' experience. During the interview with them the selected answer from the experts have been measured using five point likert scale ranging from 1representing " strong disagree" to 5 indicating strong agree.1 = strongly disagree, 2 = Disagree, 3 = Natural , 4 = Agree , 5 = strongly agree. We divided the result in two parts depends on the questionnaire have two parts to show the percentage of excepted and reject in each part in the questionnaire. In each part we have 10 questions (Table 2a and 2b).

CONCLUSION AND RECOMMENDATIONS

One of the difficulties how sharing knowledge and sending knowledge to right person, factor or item or any other side need it, in this study we try to improve sharing knowledge and process of flow this knowledge in software development process. Knowledge flow is the dynamic process occurring among knowledge-processing participants and in certain context wherein relevant knowledge is created, transformed, propagated and applied. Another description of knowledge flow according to the process of knowledge passing between people or knowledge processing mechanism. goal of propose KF framework is the effective KF as well as application of transferred knowledge in the tasks as

Table 2a: Part one

NO	Strongly disagree	Disagree	Natural	Agree	Strongly agree	Result
1	0	0	9	14	17	77.5%

10*4 = 40 is the total of questions
3 = 1+4+3+1 = 9 Natural, 4 = 0+5+4+5= 14 agree,
5 =9+1+3+4 = 17 strong agree
14+17 = 31 that's leads to 31\40*100 = %77.5 the acceptance of part of the theoretical and conceptual framework are valid.

Table 2b: Part two

NO	Strongly disagree	Disagree	Natural	Agree	Strongly agree	Result
1	0	0	7	22	11	82.5%

3 = 2+2+1+2 = 7 natural, 4 = 7+6+5+4 = 22 agree, 5 = 1+2+4+4 = 11 strong agree
22+11 = 33 that's leads to 33\40 * 100 = %82.5 the acceptance of part two the main framework and model from the experts
The total amount of acceptance from the experts for part one and two = 80%

process. Consequent to this, a novel complex concept in KM is the analysis, design as well as implementation of KF in software development team. The advantage of this conceptual framework can give high quality in short time by sharing the knowledge flow during the software development process to the right person at the right time this one of knowledge flow objective, in this study used the survey instrument to validate the components of knowledge flow in software development process, check what this components and factors during the processes however in this study experts review to validate the conceptual framework and what the factors during each phase in software development process. In addition in this research used cronbach alpha to find the reliability between all items, mean methods using descriptive analysis to analyze the experts review future work needs to implement empirically in industrial to achieve the framework objective in another environment.

REFERENCES

Anjewierden, A., R. de Hoog, R. Brussee and L. Efimova, 2005. Detecting knowledge flows in weblogs. Proceeding of the 13th International Conference on Conceptual Structures, Atlanta, pp: 1-12.

Bass, L., P.C. Clements, R. Kazman and M.H. Klein, 2008. Models for evaluating and improving architecture competence. Technical Report, Software Engineering Institude, Carnegie Melon.

Bland, J.M. and D.G. Altman, 1997. Statistics notes: Cronbach's alpha. BMJ, 314(7080): 572.

Bouthillier, F. and K. Shearer, 2002. Understanding knowledge management and information management: the need for an empirical perspective. Inform. Res., 8(1).

Breedt, M., 2000. A framework for knowledge management. M.Sc. Thesis, University of Pretoria, pp: 365-371.

Cortina, J.M., 1993. What is coefficient alpha? An examination of theory and applications. J. Appl. Psychol., 78(1): 98-104.

Gendreau, O. and P.N. Robillard, 2013. A process practice to validate the quality of reused component documentation: A case study involving open-source components. Proceeding of the International Conference on Software and System Process (ICSSP), pp: 61-69.

Guo, J. and Y. Wang, 2008. Context modeling for knowledge flow. Proceeding of the IEEE International Conference on Information Reuse and Integration (IRI), pp: 330-335.

Jarrahi, A. and M.R. Kangavari, 2012. An architecture for context-aware knowledge flow management systems. Int. J. Comput. Sci., 9(2(3): 12.

Kim, S., H. Hwang and E. Suh, 2003. A process-based approach to knowledge-flow analysis: A case study of a manufacturing firm. Knowl. Process Manage., 10(4): 260-276.

Lai, C.H. and D.R. Liu, 2009. Integrating knowledge flow mining and collaborative filtering to support document recommendation. J. Syst. Softw., 82(12): 2023-2037.

Lin, C., J.C. Wu and D.C. Yen, 2012. Exploring barriers to knowledge flow at different knowledge management maturity stages. Inform. Manage., 49(1): 10-23.

Luo, X., Q. Hu, W. Xu and Z. Yu, 2008. Discovery of textual knowledge flow based on the management of knowledge maps. Concurr. Comp-Pract. E., 20(15): 1791-1806.

Probst, G., S. Raub and K. Romhardt, 2000. Managing Knowledge: Building Blocks for Success. John Wiley and Sons, New York.

Reinhardt, R., 2002. Measuring and managing knowledge flows: A profound perspective of intellectual capital management. Proceeding of the Conference on Intellectual Capital: From Potential to Value Creation. Juni, Portoroz, pp: 6-7.

Rodríguez, O.M., A.I. Martínez, J. Favela, A. Vizcaíno and M. Piattini, 2004. Understanding and Supporting Knowledge flows in a Community of Software Developers. In: de Vreede, G.J., L.A. Guerrero and G. Marín Raventós (Eds.), Groupware: Design, Implementation and Use. CRIWG 2004. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3198: 52-66.

Sarnikar, S. and J.L. Zhao, 2007. Automating knowledge flows by integrating workflow and knowledge discovery techniques. Proceeding of the Winter Conference on Business Intelligence. Salt Lake City, UT, pp: 1-21.

Sarnikar, S. and J.L. Zhao, 2008. Pattern-based knowledge workflow automation: Concepts and issues. Inf. Syst. E-Bus. Manage., 6(4): 385-402.

Schutte, M. and M.M.M. Snyman, 2006. Knowledge flow elements within a context – a model. S. Afr. J. Inform. Manage., 8(2): 1-17.

UPEDU, 2001. Unified Process for Education. Retrieved from: upedu.org.

Yoo, K., E. Suh and K.Y. Kim, 2007. Knowledge flow-based business process redesign: Applying a knowledge map to redesign a business process. J. Knowl. Manage., 11(3): 104-125.

Zhang, X. and M. Li, 2005. Workflow-based knowledge flow modeling and control. J. Softw., 16(2): 184-193.

Zhang, Z., Z. Yang and Q. Liu, 2008. Modeling knowledge flow using Petri Net. Proceeding of the IEEE International Symposium on Knowledge Acquisition and Modeling (KAM'08). China, pp: 142-146.

Zhao, W. and W. Dai, 2008. Integrated modeling of business processes and knowledge flow based on RAD. Proceeding of the IEEE International Symposium on Knowledge Acquisition and Modeling (KAM'08), China, pp: 49-53.

Zhuge, H., 2002. A knowledge flow model for peer-to-peer team knowledge sharing and management. Expert Syst. Appl., 23(1): 23-30.

Zhuge, H., 2006a. Knowledge flow network planning and simulation. Decis. Support Syst., 42(2): 571-592.

Zhuge, H., 2006b. Discovery of knowledge flow in science. Commun. ACM, 49(5): 101-107.

Zhuge, H. and W. Guo, 2007. Virtual knowledge service market—for effective knowledge flow within knowledge grid. J. Syst. Softw., 80(11): 1833-1842.

Zhuge, H., T.Y. Cheung and H.K. Pung, 2001. A timed workflow process model. J. Syst. Softw., 55(3): 231-243.