A Model for Assessing Organizational Learning in Software Development Organizations

Oumout Chouseinoglou, Semih Bilgen

A Model for Assessing Organizational Learning in Software Development Organizations

Oumout Chouseinoglou[1] and Semih Bilgen[2]

[1] Statistics and Computer Science Department, Başkent University,
06810, Ankara, Turkey,
umuth@baskent.edu.tr
[2] Electrical and Electronics Engineering Department, Middle East Technical University,
06531, Ankara, Turkey,
semih-bilgen@metu.edu.tr

Abstract. In order to keep up with the continuously increasing competition and to obtain competitive advantage, software developer organizations (SDO) need to possess the characteristics of Learning Software Organizations (LSO). Maturity is directly related to both learning and knowledge management (KM). However, the major software process improvement (SPI) approaches do not explicitly address how learning capabilities of a SDO can be assessed or what knowledge needs to be managed and how, when, where, or by and for whom. This paper introduces a model for evaluating the organizational learning characteristics of a SDO. We report the results of applying the model in a university course on software development.

Keywords: Learning software organization, software process improvement, SQ4R

1 Introduction

None of the well-known models currently in use for evaluating the level of quality in software products and the maturity of development processes (ISO15504, CMMI) explicitly focuses on organizational learning; that is the process of learning by individuals and groups in a software organization through the development process, even though CMMI provides an infrastructure for organizational learning and systematic improvement [1].

Learning is the necessary prerequisite of knowledge as well as maturity. The basis for increasing the level of maturity is the ability for organizational learning. Knowledge is one of the most important assets of an organization. The importance of knowledge as an asset increases for organizations that use knowledge-intensive processes. In order to keep up with the continuously increasing competition and to obtain competitive advantage, Software Development Organizations (SDO) need to *obtain* the correct knowledge, *use* it efficiently and *pass* it to future projects. These three constitute the major process areas of knowledge management (KM). A SDO that manages the processes of obtaining, using and passing knowledge, and learns while

developing software, is a Learning Software Organization (LSO). As software process improvement (SPI) is a knowledge-intensive task, many SDOs have recognized the importance of administrating knowledge effectively, productively and creatively at both the individual and organizational levels [2]. Organizations with greater learning-related scale, knowledge, and diversity are more likely to initiate and sustain the assimilation of complex technologies such as SPI [3].

Below, we introduce a model that will allow SDOs to assess their knowledge management activities in all process areas, identify those that need improvement and monitor their continuous improvement.

The rest of the paper is organized as follows: First, we briefly review the basic literature on KM and LSO. Section 3 presents our model for representing and assessing organizational learning in SDOs. A case study is presented in Section 4 and its results are discussed in Section 5 providing a validation of the model based on expert opinions. The last section concludes the paper.

2 Learning Software Organizations

Learning organizations are defined in [4] as organizations where people continually expand their capacity to *create* the results they truly desire, where *new and expansive* patterns of thinking are nurtured, where *collective aspiration* is set free, and where people are *continually* learning how to learn together and in [5] as a group of people who systematically extend their capacities so as to accomplish *organizational goals*. Therefore the learning process should be tailored, designed and applied accordingly to serve the overall goals of the organization, resulting in "organizational learning".

Focusing specifically on SDOs [5], and similarly [6], define an LSO as an organization that learns within the domain of *software development*, *evolution* and *application* where the objects of learning can consist of models, knowledge and lessons learned related to the different processes, products, tools, techniques and methods applied during the different stages of the software development process.

Regarding the significance of KM in software engineering, it is stated in [7] KM acknowledges the importance of *individuals* having access to the correct information and knowledge when they need to complete a task or make a decision and works toward SPI by explicitly and systematically addressing the management of organizational knowledge. The main shortcoming of the major SPI approaches such as the CMMI is that they do not explicitly state what knowledge needs to be managed and how, when, where, or by and for whom. Therefore, KM needs to address this limitation of the existing SPI approaches in order to support the establishment of a LSO.

[8] and [9] review the KM literature, showing that only a few studies are related to SPI, and that there is a need for different KM insights within the domain of software engineering. Various authors ([11,12,13]) group the processes proposed in each model under four stages; namely the stages of creation, storage, dissemination and utilization with the supplementary phase of measurement. [10] and [14] define the

knowledge evolution cycle which consists of five phases of organizational knowledge[1], linked to each other in a cyclic fashion.

Investigating the proposed KM models and schemes, two important conclusions may be drawn: firstly, that KM is not a monolithic process but instead it consists of several different processes that need to be addressed and measured separately and secondly that the KM process is of a continuous nature.

3 The Proposed Model

Based on the literature survey on LSO and KM, which is provided in detail in [15], we propose a model for the assessment of organizational learning in SDOs. The main aims of this model are a) to provide a framework for comparison between SDOs with respect to their organizational learning capabilities, b) to allow SDOs to identify their deficiencies and shortcomings, and c) to provide a starting point for SPI and to measure the realized improvement. The model consists of 3 major process areas that map to the three major objectives of a LSO and are connected to each other in a continuous fashion to depict the continuity of the learning activity, which can be assessed with respect to 12 core processes that are an elaboration of the 3 major process areas. In contrast to surveyed KM models in [11], the proposed model focuses on the human factor and not on knowledge stored in tools and knowledge bases, acknowledging the importance of humans and groups in the organizational learning process [16]. With that viewpoint, the model tries to capture and assess the organizational learning realized in human agents and teams but also on human developed artifacts, such as documents, practices and processes. The basic structure of the model is shown in Fig. 1.

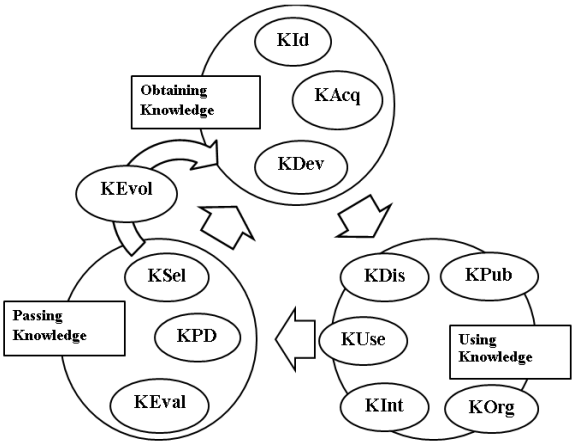

Fig. 1. The proposed model with 3 major process areas and 12 core processes

[1] Namely originate/create knowledge, capture/acquire knowledge, transform/organize knowledge, deploy/access knowledge and apply knowledge [14]

In [15], we describe, in detail, the following 12 core processes constituting the 3 major process areas of LSOs.

- **Obtaining Knowledge**
  - Knowledge Identification (Discovery or Capturing) - *KId*
  - Knowledge Acquisition (Buying) - *KAcq*
  - Knowledge Development (Creation or Construction) - *KDev*
- **Using Knowledge**
  - Knowledge Organization - *KOrg*
  - Knowledge Dissemination (Sharing or Distribution) - *KDis*
  - Knowledge Publication - *KPub*
  - Knowledge Usage (Application or Utilization) - *KUse*
  - Knowledge Integration (Routines) - *KInt*
- **Passing Knowledge**
  - Knowledge Preservation (Retention or Archiving and Deleting) - *KPD*
  - Knowledge Evaluation (Valuation) - *KEval*
  - Knowledge Selling - *KSel*
  - Knowledge Evolution - *KEvol*

In order to assess a SDO within the proposed model, appropriate indicators are necessary. The importance of measuring the KM process is discussed in [17] and a list of measurement models in literature related to a KM process is provided in detail in [11]. In order to identify and define the appropriate measurements for the proposed model we used the Goal/Question/Metric (GQM) approach [18]. The full description of the undertaken GQM approach and its resulting metrics are given in [15]. Generic measures of the model are listed in Table 1.

4 The Case Study

A case study was conducted to validate the proposed model in the context of a one semester software engineering course, İST478, in Başkent University. 15 undergraduate and 4 graduate level students were enrolled and 4 software development groups were formed, with each graduate student assigned as team leader. The course followed a customization of the outline provided by CSCI577ab[2] [19] applying the Incremental Commitment Spiral Model (ICSM) which consisted of the Exploration, Valuation, Foundations, Development, and Transition phases (phases 1,2,3,4 and 5 respectively). The deliverable deadlines and the items to be delivered for each of these phases were predefined. The tasks and artifacts to be developed were based on specific templates and they were described in detail in the ICSM – Electronic Process Guide[3]. A detailed list of deliverable phases, dates, deliverable packages and artifacts is provided in [15].

[2] Software Engineering I – Fall 2011, http://greenbay.usc.edu/csci577/fall2011/index.php
[3] http://greenbay.usc.edu/IICMSw/index.htm

Table 1. The proposed model and the relative generic measures

Major Process Area	Core Process	Generic Measure	Short Name
Obtaining Knowledge	Knowledge Identification	Internal Trainings	KId1
		Tasks Completed Internally	KId2
		Documents Completed Internally	KId3
	Knowledge Acquisition	External Trainings	KAcq1
		Utilized External Communication	KAcq2
		Trained Topics	KAcq3
		Utilized External Documents	KAcq4
	Knowledge Development	Creative Idea Development	KDev1
		Creative Idea Evaluation	KDev2
Using Knowledge	Knowledge Organization	Horizontal document linking	KOrg1
		Vertical document linking	KOrg2
	Knowledge Dissemination	Information messages from management	KDis1
		Amount of meetings	KDis2
		Length of meetings	KDis3
		Meeting Discussion Efficiency	KDis4
	Knowledge Publication	Internally Distributed Guidelines	KPub1
		Externally Distributed Guidelines	KPub2
		Academic Publications	KPub3
	Knowledge Usage	Creative Idea Application	KUse1
		Deliverable Quality	KUse2
		Meeting Functional Efficiency	KUse3
	Knowledge Integration	Task Differentiation within phases	KInt1
		Deliverable Differentiation within phases	KInt2
		Deliverable Correction	KInt3
Passing Knowledge	Knowledge Preservation and Deleting	Knowledge evaluation and assessment	KPD1
		Task differentiation from guidelines	KPD2
		Deliverable differentiation from templates	KPD3
	Knowledge Evaluation	Valuated Items	KEval1
	Knowledge Selling	Shared Documents	KSel1
		Shared Tasks	KSel2
		Trainings Given	KSel3
	Knowledge Evolution	Guideline Evolution between Projects	KEvol1
		Task Evolution between Projects	KEvol2
		Deliverable Evolution between Projects	KEvol3

To determine whether the proposed model assesses the difference of organizational learning between different groups, two of the groups were assigned a differentiated development method, SQ4R (Survey, Question, Read, Recite, Review, and wRite) [20], based on critical thinking, to enhance their learning experience. The SQ4R approach was undertaken by two randomly assigned groups (namely Group 2 and 3) in all five phases of the software development lifecycle of İST478 course. The details on how the SQ4R approach was utilized are given in [15].

The core processes in Table 1 were investigated with respect to their applicability to the course structure. The non-applicable core processes were not assessed. Moreo-

ver, the generic measures proposed were refined with respect to course characteristics, the artifacts produced and the deliverables developed by the project groups, and were transformed into actual metrics. The evaluation period of the measures has been identified as the predefined five development phases. The detailed list and explanation of each generic and applied metric, the calculation formula and the interpretation of each result are presented in [15].

5 Results of the Case Study

The learning ability of the groups based on the measures was calculated and assessed at the end of each phase by individual questionnaires [15] and the meeting minutes. Exit interviews were conducted with each group to resolve any inconsistencies or anomalies in the questionnaires.

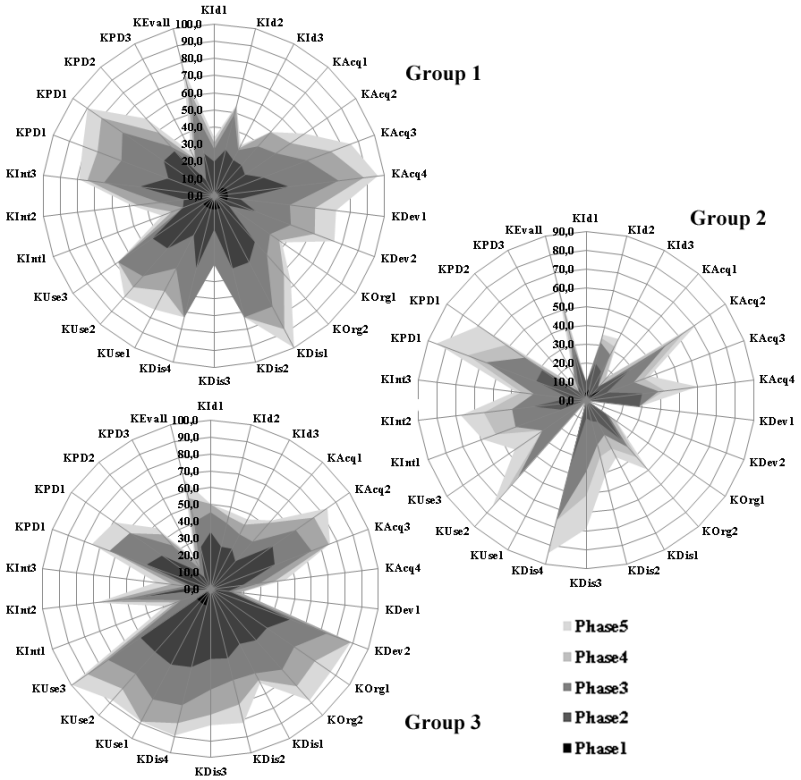

Fig. 2. The Organizational Learning footprint of the groups in the case study

The students undertook 7 in-class examinations to measure the amount of preserved knowledge within the group (KPD1), and the submitted documents were graded to assess the product quality of the deliverables (KUse2), but also to identify the document defects and the defect removal ratio (KInt3). All measures were normalized

appropriately [15]. Fig. 2 displays the organizational learning progress of each group with respect to the measured key process areas.

Among the groups, only Group 1 had not undertaken the SQ4R approach. From the footprints it can be seen that Group 1 scores low in knowledge identification, organization, integration and preservation. On the other hand Group 2 scores low almost in all key processes, except knowledge acquisition and integration. Group 3 also scores low in knowledge identification and integration. As the majority of the students in these groups lack any professional software engineering development practice experience or relative knowledge, we were expecting low scores in knowledge identification and development, but higher in knowledge acquisition. The results of Group 2 can be justified by a communication problem between group members. On the other hand Group 1 has scored high due to the high cohesion between its members. The high scores of Group 3 have resulted from the SQ4R approach that allowed the members to build a commitment towards the software development process.

As it can be seen from the footprints, with the use of appropriate and correct metrics, the organization can easily identify its weak learning process areas and thus develop a strategy to provide a solution for these weaknesses. Although the metrics were coined to meet the specific requirements of an in-class software development group, they can be easily modified to match the needs of any SDO.

The exit interview with team leaders led to the conclusion that learning ability assessment enabled by this model can be fully used in SPI.

6 Conclusions

We have introduced a model that allows SDOs to measure and assess their learning capabilities, identify their strengths and weaknesses in terms of learning and to proceed with building a competitive advantage by becoming a LSO. Based on the case study results and the expert opinions it is evident that the proposed model is a step towards this goal. Although the findings show that it can be applied in the organizational context, further implementations of the SQ4R approach in the business environment are currently under way to provide better insights of its value for software development activity.

As stated in [21] the validity of the model and of the embedded formulations must be strengthened through numerous case studies. As an extension of this study, it is of crucial importance to continue with the integration of the identified core processes with existing software maturity models.

References

1. Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, S.: CMMI or Agile: Why Not Embrace Both! Software Engineering Institute of Carnegie Mellon University, (2008) accessible at http://www.sei.cmu.edu/publications/documents/08.reports/08tn003.html
2. Santos, G., Montoni, M., Figueiredo, S., Rocha, A.R.: SPI-KM-Lessons Learned from Applying a Software Process Improvement Strategy Supported by Knowledge Management,

8th Int. Conference on Product Focused Software Process Improvement (PROFES'2007), Latvia (2007)
3. Fichman, R., Kemerer, C.: The Assimilation of Software Process Innovations: An Organizational Learning Perspective. Management Science, vol. 43, no. pp. 1345-1363 (1997)
4. Senge, P. M.: The Fifth Discipline: The Art and Practice of the Learning Organization. Doubleday/Currency, New York (1990).
5. Ruhe, G.: Learning Software Organisations. In: Chang, S. (ed.), Handbook of Software Engineering and Knowledge Engineering, vol. 1, pp. 663--678. World Scientific Publishing 2001, (2001)
6. Henninger, S., Lappala, K., Raghavendran, A.: An Organizational Learning Approach to Domain Analysis. In: 17th International Conference on Software Engineering, pp. 95--104. ACM Press, New York (1995)
7. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering. Software, IEEE, vol: 19, no. 3, pp. 26--38 (2002)
8. Bjørnson, F., Dingsøyr, T.: Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts, Findings and Research Methods Used. In: Information and Software Technology, vol. 50 no. 11, pp. 1055--1068 (2008)
9. Sharma, N., Singh, K., Goyal, D.: Can Managing Knowledge and Experience Improve Software Process? - Insights From the Literature. Research Cell: An International Journal of Engineering Sciences, vol: 4, pp. 324--333 (2011)
10. Maier, R.: Knowledge Management Systems: Information and Communication Technologies for Knowledge Management. Springer–Verlag, New York (2004)
11. Oliveira, M., Goldoni, V.: Metrics for Knowledge Management Process. In: IAMOT 2006 15th International Conference on Management of Technology, Beijing (2006)
12. Lee, K. C., Lee, S., Kang, I.W.: KMPI: Measuring Knowledge Management Performance. Information and Management, vol. 42, no. 3, pp. 469--482 (2005)
13. Bose, R.: Knowledge Management Metrics. Industrial Management & Data Systems, vol. 104, no. 6, 457--468 (2004)
14. Agresti, W.: Knowledge Management. Advances in Computers, vol. 53, no. 1, pp.171--283 (2000)
15. Chouseinoglou, O., Bilgen, S.: A Model for Assessing Organizational Learning in Software Organizations and a Case Study. Technical Report. METU/II-TR-2012-01, Department of Information Systems. Middle East Technical University, (2012) accessible at http://www.baskent.edu.tr/~umuth/METU-II-TR-2012-01.pdf
16. Haas, M.R., Hansen, M.T.: Different Knowledge, Different Benefits: Toward a Productivity Perspective on Knowledge Sharing in Organizations. Strategic Management Journal, vol. 28, pp. 1133--1153 (2007)
17. Ahmed, P.K., Lim, K.K., Zairi, M.: Measurement Practice for Knowledge Management. Journal of Workspace Learning: Employee Counseling Today, vol. 11, no. 8, pp. 304--311 (1999)
18. Solingen, R. v., Berghout, E.: The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. McGraw-Hill Publishing, London, (1999)
19. Boehm, B., Lane, J.: Using the Incremental Commitment Model to Integrate Systems Acquisition, Systems Engineering, and Software Engineering, Cross Talk, pp. 4--9 (2007)
20. Thomas, E.L., Robinson, H.A.: Improving Reading in Every Classroom: A Source Book for Teachers. Allyn & Bacon, Boston (1982)
21. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research in Software Engineering, Empirical Software Engineering, vol. 14, pp. 131--164. (2009)