An Exploration of the Relationship between Organizational Learning and Software Development Process Maturity

CHUN-HUI WU[1,2]
[1]Department of Information Management
National Yunlin University of Science & Technology
[2]Department of Information Management
National Formosa University
64, Wen-Hwa Rd, Huwei, Yunlin, 632
TAIWAN

Abstract: - Even after significant resources are poured into software development process, software development process maturity and productivity are still low. Organizational learning has long been realized as a key strategy for improving performance. In our research, we argued that organizational learning could affect the software development process maturity. Data collected via a survey of two groups that is randomly selected from Information Service Industry Association (CISA) and Information Management Association (IMA) members. The results indicate that organization learning have a positive impact on the software development process maturity. These conclusions should help managers revisit their priorities in terms of the relative efforts in software development processes.

Key-Words: - Organizational Learning, Software Development Process Maturity, Capability Maturity Model

1 Introduction

In recent decades, software investment has grown rapidly worldwide and software project continue to grow more critical to the organizations that employ them [11]. As information technologies are evolving so quickly and information system applications grow in size, complexity, and critically, a solution for improving software development processes has become more and more imperative. Many tools, technologies and management methods have been undertaken to guide the management of the development processes over the years [2,13]. However, even after significant resources are poured into software development processes, software development process maturity and productivity are still low [3,20, 22]. In large software development projects, more than 80 percept are excessively late and over budget, and the benefits of better software development methods and tools cannot be realized in “runaway,” undisciplined project development environments [10,14]. This situation has put in evidence the inability of these organizations to manage their software development processes.

Faced with the challenge, managers in many organizations are pursuing software process improvements, and their goals are meeting tight schedules and budgets, and business objectives [6]. Organizational learning is an important approach that advocates that organizations must learn from their experiences in order to survive [21]. Today, it has become one of the most significant trends affecting project management. The capacity for change and improvement has increasingly become associated with organizational learning. Nelson and Cooprider [16] stated that research in IS area has shown that organizational learning is important in improving software project performance.

Past research indicated that Taiwanese business companies rarely gathered error statistics, including code errors, test errors, and design errors. Therefore, they did not have the ability to make error prediction and prevent the errors from happening, not to mention learning how to reduce human errors in design, code, and test activities. Unless organizational learning environment are built in project management practices, past mistakes will be repeated.

Plenty of literatures discussed with the software development process maturity. But there is little empirical evidence to show the impact of the organizational learning on software development process maturity. It is here that valid and reliable empirical evidence is needed. The purpose of this study is trying to explore the relationship between organizational learning and software development process maturity. Specifically, the following question is addressed in this study:

Will increase in organizational learning directly associate with increases in software development process maturity?

2 Related Study

2.1 Organizational learning

The first publications on organizational learning appeared in 1965 [5]. The subject of organizational learning has gained considerable attention since 1980s. The notion of organizational learning has become very prominent in the near past. Managers see organizational learning as a powerful tool to improve the performance of an organization. Thus, it is not only the scholars of organization studies who are interested in the phenomenon of organizational learning but also the practitioners who have to deal with the subject of organizational learning.

Organizational learning is increasingly becoming popular among organizations that are interested in increasing competitive advantage, innovativeness, and effectiveness. Argyris and Schon [1], two of the early researchers in this field, defined organizational learning as a process of detecting and correcting errors so that organizations are able to function and realize their goals and objectives. Fiol and Lyles [9] define learning as "the process of improving actions through better knowledge and understanding". Dodgson [8] describes organizational learning as "the way firms build, supplement, and organize knowledge and routines around their activities and within their cultures and adapt and develop organizational efficiency by improving the use of the broad skills of their workforces". Nadler et al. [15] indicated that organization's ability to change or redesign themselves continuously may be necessary for survival.

2.2 Software development process maturity

To better manage the software development process and increase the chances of software success, the Software Engineering Institute (SEI), for the US Department of Defense (DOD), recommends a number of key software process improvement areas to enhance software development capability. These are formalized into an evaluative framework called the Capability Maturity Model (CMM) [18]. The major objective of CMM is to achieve defined, managed, and optimized processes and also yield a higher maturity of the software development projects. This model is based on knowledge acquired from software processes assessments and extensive feedback from both industry and government.

The CMM is a framework for appraising and improving key practices used as a standard and a guide for evaluating the current state of the organization's software project to provide reliable software development and maintenance capabilities. In fact, the CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process. The CMM covers practices for planning, engineering and managing software development and maintenance. When followed, these key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality. Today, the CMM has already achieved wide interest and acceptance and is now used by thousands of organizations around the globe.

2.3 Research Hypothesis

Calantone et al. [4] found that learning orientation has a direct influence and facilitate the generation of resources and skills essential on firm performance. Snyder and Cummings [19] proposed organizational learning processes affect organization knowledge which, in turn, influences organization performance such productivity and customer satisfaction. Thus, organizational learning is an important factor that will affect the software performance and maturity. Based upon the findings in the literature, we propose:

H1. Increases in organizational learning will be directly associated with increases in software development process maturity

3 Research methodology

A survey methodology was used to test the hypothesis. This section describes the design of the survey instrument and the methods used for developing and administering.

3.1 Sample

Questionnaires were mailed to two separate groups: 1) 400 randomly selected Information Service Industry Association (CISA) members and 2) 250 randomly selected Information Management Association (IMA) members. CISA currently has more than 800 members including government supported IT organizations, and domestic software companies. IMA has over 500 members including IS managers and IS professionals. These samples were chosen because members of CISA and IMA represent a cross section of managerial positions extensively involved with project management and have been widely used in past software development management research in Taiwan. Postage-paid envelopes for each questionnaire were enclosed. All the respondents were assured that their responses would be kept confidential.

Of the 650 initial surveys mailed, a total of 127 responses were received. In order to increase the sample size, a second mailing was conducted. The response from both samples totaled 209, for an overall response rate of 32.5%. Thirteen questionnaires were eliminated due to missing data, leaving a final sample of 196 used in the data analysis.

The respondents consisted of software managers (29%), project leaders (13%), software professionals (42%), and software users (12%). About 72% worked in companies that had 100 employees or more. These samples included firms from a variety of industries including health, manufacturing, communication, information service, electronic, transportation, automotive, banking, steel machine, and education. The combined industries represented 118 service firms (60.2 percent), 68 manufacturing firms (34.7 percent), and others. Demographic features of the sample population appear in Table 1.

Table1 Demographics Information

Gender	
Male	74.5%
Female	22.4%
No response	3.1%
Age	
30 and under	25.0%
31 – 40	49.5%
41 – 50	16.8%
51 and over	0.5%
No response	8.2%
Position	
Software Manager	29.1%
	13.3%
Project Leader	42.3%
	11.7%
Software Professional	3.6%
Software User	
No response	
Industry category	
Service	60.2%
Manufacturing	34.7%
Education	1.5%
No response	3.6%
Number of employees	
Less than 100	24.0%
	19.9%
101-300 employees	12.8%
300-500 employees	38.3%
501 or more employees	5.1%
No response	
Size of IS project teams	
7 and under	49.5%
8 – 15	30.1%
16 – 25	7.1%
26 and over	8.2%
No response	5.1%

3.2 Construct

The instrument consisting of 38 items used to measure the software development process maturity level was adopted from Dekleva and Drehmer [7]. Dekleva and Drehmer presented an empirical scaling of software engineering practices derived from the software process maturity model developed by the SEI. The full text of the 38 survey items is shown in Appendix I. All items of this instrument were key items representing the SEI repeatable (level 2), defined (level 3) and managed (level 4) maturity thresholds. Items 1-12 were SEI level 2 items, items 13-26 were level 3, and items 27-38 were level 4. Since no organization was believed to achieve the optimizing level, none of these items was used. The respondents are asked to evaluate the overall extent of each structure and procedure implemented in their organizations' IS projects. Each scale will be scored using a five-point scale ranging from "not at all" (1) to "extremely" (5) and averaged across all relevant items.

The organizational learning measure is a subset of items identified by Nidumolu [17]. The questionnaire asks respondents to identify the extent of knowledge learning in their organization according to their personal knowledge and experience. The specific items are shown in Table 2. Each item was scored using a five-point scale ranging from never (1) to always (5). All items were presented so that the greater score represented the greater learning.

4 Data Analysis

The preliminary analysis of the responses addressed the external and construct validities, and the reliability of the study. External validity of the study was assessed and factor analysis was conducted to examine the structural validity of the measure.

.

4.1 External Validity

Non-response bias occurs when the opinions and perceptions of the survey respondents do not accurately represent the overall sample to whom the survey was sent. One test for non-response bias is to compare the answers of early versus late respondents to the survey. The idea is that late respondents are more likely to answer the questionnaire like non-respondents than are early respondents.

T-tests were computed on the means of key demographics (work experience, gender, recent project duration, and team sizes) to examine whether significant differences existed between early and lately respondents. No significant difference was found at the 95% confidence level, indicating the absence of the bias. Therefore, these two rounds of respondents were combined for further analysis.

4.2 Measures for constructs

In this study, we attempted to use valid and reliable measures. First, items were adopted from previously validated instruments in literature. Second, the reliability and validity of the measures were reassessed with the current sample.

Measure of sampling adequacy (MSA) is examined to assess whether or not a set of variables are appropriate for factor analysis. Kaiser and Rice [12] provided a calibration of the MSA measure with regard to the degree of appropriateness of using factor analysis on a set of measurement item. The measure can be interpreted with the following guidelines: a value of 0.80 or above, meritorious; values between 0.60 and 0.80, mediocre; and below 0.50, unacceptable. The MSA values of two constructs in this study were computed to be 0.95 and 0.80 and suggested that the items were appropriate for factor analysis.

Cronbach α is calculated to test measurement reliability. Alpha value will be high if the various items that constitute the construct are strongly correlated with one another. Chonbach $\alpha \geq 0.70$ are judged to be high in internal consistency. The reliabilities of the scales are shown in Table 2. All alphas are greater than the recommended 0.70 level, thereby implying an adequate level of internal consistency. Thus, these items have acceptable reliability and validity for testing the research questions, and the sample is adequate.

Table 2 Factor Analysis Results

Construct	Cronbach α
Organizational learning	.90
Software development process maturity	
Level II	.91
Level III	.94
Level IV	.94

4.3 Results

To further examine the relationship between constructs, regression analysis was conducted to test the hypothesis. Table 3 provides results of the regression analysis used to test whether there is a significant relationship between constructs. The *P*-value (<.001) indicates that there ia a significant relationship between organizational learning and software development process maturity levels. Therefore, we conclude that organizational learning influence software development process maturity levels within software development processes. This result gives support for hypothesis H1.

Table 3 Results of Regression Analysis

Dependent Variable	Independent Variable	Coefficient	*P*-value
Organizational learning	Software Development Process Maturity	0.44	<.0001*

5 Conclusions

The concept of organizational learning is receiving an increasing amount of attention in the research and practice due to its potential for affecting performance. However, the improvement of organization learning process in software development field is little mentioned in the past literature. Based on organizational learning theory, we examined the relationship among organization learning and software development process maturity. The analysis of results demonstrates that increases in organizational learning directly associate with increases in software development process maturity. The organizational learning context plays a critical role in affecting software development process maturity. The confirmation of the relationship suggests future failure avoidance solutions consider organization learning. Moreover, the strong effects of organizational learning provide significant support for the software development that expect to effectively control and monitor the project with respect to cost, schedule, quality and user needs.

References:

[1] C. Argyris, and D. Schon, Organizational Learning: A Theory of Action Perspective, *Addison-Wesley*, Reading, MA, 1978.

[2] S. Bandinelli, and A. Fuggetta, Modeling and Improving an Industrial Software Process, *IEEE Transaction on Software Engineering*, Vol.21, No.5, 1995, pp. 440-454.

[3] H. Barki, S. Rivard, and J. Talbot, Toward an Assessment of Software Development Risk, *Journal of Management Information Systems*, Vol.10, No.2, 1993, pp. 203-223.

[4] R. J. Calantone, S. T. Cavusgil, and Y. Zhao, Learning Orientation, Firm Innovation Capability, and Firm Performance. *Industrial Marketing Management*, Vol.31, 2002. pp. 515-524.

[5] V. Cangelosi, and W. Dill, Organizational Learning: Observations toward a Theory, *Administrative Science Quarterly*, Vol.10, 1965, pp. 175-203.

[6] C. Deephouse, T. Mukhopadhyay, R. Goldenson, and M. Kellner, Software Processes and Project Performance, *Journal of Management Information Systems*, Vol.12, 1995, pp. 187-205.

[7] S. Delkleva, and D. Drehmer, Measuring Software Engineering Evolution: A Rash Calibration, Information Systems Research, Vol.8, 1997, pp. 95-104.

[8] M. Dodgson, Organizational Learning: A Review of Some Literatures, *Organization Studies*, Vol.14, 1993, pp. 375-394.

[9] C. Fiol, and M. Lyles, Organizational Learning, *Academy of Management Review*, Vol.10, 1985, pp. 803-813.

[10] R.L. Glass, Software Runways, *Prentice-Hall*, New York, NY, 1998.

[11] J. J. Jiang, G. Klein, H. G. Hwang, J. Huang, and S. Hung, An Exploration of the Relationship between Software Development Process Maturity and Project Performance, *Information & Management*, Vol.41, 2004, pp. 279-288.

[12] H. Kaiser, A Second Generation Little Jiffy, *Psychometrical*, Vol.35, 1970, pp. 402-415.

[13] J.P. Kuilboer, and N. Ashrafi, Software Process and Product Improvement: An Empirical Assessment, *Information and Software Technology*, Vol.42, 2000, pp. 27-34.

[14] K. Linberg, Software Developer Perceptions about Software Project Failure: A Case Study, *The Journal of Systems and Software*, Vol.49, 1999, pp. 177-192.

[15] D. Nadler, R. Shaw, and A.E. Walton, Discontinuous Change: Leading Organizational Transformation, *Jossey-Bass*, San Francisco, 1994.

[16] K. Nelson, and J. Cooprider, The Contribution of Shared Knowledge to IS Group Performance, *MIS Quarterly*, 1996, pp. 409-432.

[17] S. Nidumolu, The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as An Intervening Variable, *Information Systems Research*, Vol.6, 1995, pp. 191-219.

[18] M. Paulk, B. Curtis, M. Chrissis, and C. Weber, Capability Maturity Model for Software, version 1.1, *IEEE Software*, 1993, pp. 18-27.

[19] W.M. Snyder, and T.G. Cumming, Organization Learning Disorders: Conceptual Model and Intervention Hypotheses, *Human Relation*s, Vol.51, No.7, 1998, pp. 873-895.

[20] H.E. Thomson, and P. Mayhew, The Software Process: a Perspective on Improvement, *The Computer Journal,* Vol.37, No.8, 1994, pp. 683-690.

[21] E. Turban, R. Rainer, and R. Potter, Introduction to Information Technology, *John Wiley & Sons*, NY, 2003.

[22] P. Weill, and M. Broadbent, Leveraging the New Infrastructure – How Market Leaders Capitalize on Information Technology, *Harvard Business School Press*, Boston, MA, 1998.