

Research on Explicit and Tacit Knowledge Interaction in Software Process Improvement Project*

Jiangping Wan^{1,2}, Dan Wan¹, Weiping Luo¹, Xiaoyao Wan¹

¹Faculty of Business Administration, South China University of Technology, Guangzhou, China; ²Institute of Emerging Industrialization Development, South China University of Technology, Guangzhou, China.

Email: {scutwj, dandanwan42}@126.com, luoweiping18@139.com, 294663012@qq.com

Received May 19th, 2011; revised June 5th, 2011; accepted June 13th, 2011.

ABSTRACT

This research develops a knowledge model for Software Process Improvement (SPI) project based on knowledge creation theory and its twenty-four measurement items, and proposes two hypotheses about the interaction of explicit knowledge and tacit knowledge in SPI. Eleven factors are extracted through statistical analysis. Three knowledge-creation practices for capturing tacit knowledge contribute greatly to SPI, which are communication among members, crossover collaboration in practical work and pair programming. Two knowledge-creation practices for capturing explicit knowledge have significant positive impact on SPI, which are integrating project document and on-the-job training. Ultimately, suggestions for improvement are put forward, that is, encouraging communication among staff and integrating documents in real time, and future research is also illustrated.

Keywords: Knowledge Creation, Software Process Improvement, Explicit Knowledge, Tacit Knowledge, Communication, Document

1. Introduction

The theory of knowledge creation [1] is based primarily on Polanyi's [2] categorization of knowledge as explicit and tacit. It prescribes the capture of both explicit and tacit types of knowledge, making it available to the organization in order to generate competitive capabilities. Explicit knowledge is codified knowledge articulated in words, figures, and numbers. It is objective, and relatively easy to share in the form of specifications, standard operating procedures, and data. Tacit knowledge is knowledge that has not been codified and is relatively difficult to codify. It is subjective and based on individual experiences.

The software process is the set of tools, method, and practices we use to produce a software product. The objectives of software process improvement (SPI) are to process produce products according to plan while simultaneously improving the organization's capability to produce better products [3]. The six basic principles of

SPI by Watts S. Humphrey are in the following: 1) Major changes to the software process must start at the top. 2) Ultimately, everyone must be involved. 3) Effective change requires a goal and knowledge of the current process. 4) Change is continuous. 5) Software process changes will not be retained without conscious effort and periodic reinforcement. 6) Software process improvement requires investment [3]. Alfonso Fuggetta argues that the scope of software process improvement methods and models should be widened in order to consider all the different factors affecting software development activities. We should reuse the experiences gained in other business domains and in organizational behavior research. Statistics is not the only source of knowledge. We should also appreciate the value of qualitative observations [4]. J. P. Wan argues that managers should think deeply into their think processes. The following issues in software organization can be resolved with SPI: 1) The processes and their principles for how to inherit and acquire others' knowledge. 2) The processes and their principles for conversion knowledge into their capability [5]. 400 process improvement experiments and presents patterns are

*This research was supported by Key Project of Guangdong Province Education Office (06JDXM63002).

in a repository to help organizations plan their improvement initiatives [6].

We begin by relating knowledge creation to SPI projects in Section 2. In Section 3, we develop our conceptual arguments, and present hypotheses that relate practices for capturing explicit and tacit knowledge to SPI project success. Section 4 describes the development of our survey instrument and the empirical methodology used to test our hypotheses. We present the results of our analyses and discussions in Section 5, the conclusions are in Section 6.

2. Literature Review

2.1. Software Process

Since the computer software is a kind of logical product, its quality improvement is difficult and complex. It is clear that a fully effective software process must consider the relationships of all the required tasks, including the tools and methods used, the skill, training, and motivation of the people involved, SPI is usually implemented by project [3,7].

An economist, Howard Baetjer, commented on the software process as following [8]: because software, like all capital, is concrete knowledge, and because that knowledge is initially dispersed, tacit, latent, and incomplete in large measure, software development is a social learning process. The process is a dialogue in which the knowledge on the software is brought together and embodied in the software. The process enables interaction between users and designers, between users and evolving tools, and between designers and evolving tool (technology). It is an iterative process in which the evolving tool itself serves as the medium for communication, with each new round of the dialogue eliciting more useful knowledge from the people involved. It is obvious that software process is also an organizational knowledge-intensive learning process and needed to be supported with knowledge management.

Software organization is a highly knowledge-intensive enterprise, knowledge transfer is critical for software enterprise. It is obvious that software process is also an organizational knowledge intensive learning process and needed to be supported with knowledge management [9].

2.2. Explicit and Tacit Knowledge Types

Nonaka’s framework [1] provides a rationale for the use of knowledge-creation practices to generate group knowledge by engaging individual team members in process improvement projects. The framework depicts the process of knowledge creation as cycles of conversions between two types of knowledge—explicit, and tacit (**Figure 1**). It is worthwhile to note that this classi-

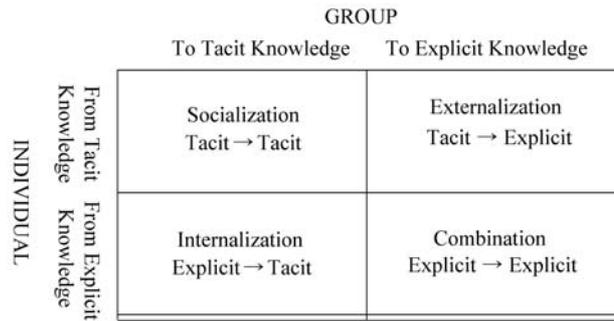


Figure 1. Nonaka’s framework of knowledge-creation mechanisms [10].

fication of knowledge as either explicit or tacit is one of two prominent classifications in the knowledge management literature (**Table 1** provides a brief overview of different classifications of knowledge creation efforts [10]).

Explicit knowledge is codified and documented, and its transfer can take place in impersonal ways—for instance, through written instructions and diagrams. Tacit knowledge is knowledge that is difficult to articulate, especially in terms of cause-effect relationships. It is context-specific, and is transferred mainly through social interactions [2]. Language is an excellent example of tacit knowledge: native speakers of a language are often unable to articulate the grammatical and syntactic rules governing it. Tacit knowledge contributes to the “stickiness” of information required for problem-solving, making it difficult for others to gather, transfer, and utilize. The difficult-to-codify nature of tacit knowledge contributes to difficult-to-imitate capabilities that may provide competitive advantage to the organization. Success of process improvement projects depends on the capture of both explicit and tacit types of knowledge [10,11].

2.3. Knowledge Enabling Software Process Improvement

Knowledge transfer model of SPI and the conceptual

Table 1. Selected classifications of knowledge-creation mechanisms [10].

Author(s)	Year	Knowledge-creation mechanisms
Argyris	1977	Single & double loop learning
Nonaka	1991	Combination, internalization, socialization & externalization
Kogut and Zander	1992	Operational & conceptual
Spender	1996	Capturing individual and organizational knowledge
Nahapiet and Ghoshal	1998	Acquiring intellectual & social capital

framework of influencing factors are established in [12]. The model includes five elements which are knowledge of transfer, sources of knowledge, recipients of knowledge, relationship of transfer parties, and the environment of transfer. The conceptual framework includes ten key factors which are ambiguity, institutionalization, transfer willingness, capacity of impartation, capacity of absorption, incentive mechanism, culture, technical support, trust and knowledge distance.

The knowledge creation effective factors were found in both necessary elements for stimulus of knowledge creation and the key influencing factors of software project success. The research was carried with the specific successful practices of both Microsoft Corporation and William Johnson’s analysis of R&D project knowledge creation in [13]. The knowledge creation effective factors in requirement development project are clarified through deeply interviewing the software enterprises in Guangdong province as well as other corporate information departments. After field survey and literature review, we found that software requirement development (SRD) is a knowledge creation process. Knowledge creation theory of Nonaka is appropriate for analyzing knowledge creating of SRD. The issue of this research is exported: how to improve SRD with knowledge creation theory? And it includes three sub-issues: 1) What factors are impacting SRD in the view of knowledge creation theory? 2) What do enable knowledge flow during SRD? 3) How can we guide SRD by using knowledge creation theory? Case study findings include [14]: 1) It can facilitate the implementation of the project to have the necessary diversity of the project team. 2) The introduction of experts on requirement can achieve the transformation of knowledge effectively, thus helping to carry out the project. 3) Methodology and related technologies are important basis for carrying out the project.

Knowledge creation theory is also applied to analyze

the open source software community with successful application of the typical agile software methods, ten principles of knowledge creation in open source software community are put forward as follows: self-organizing, code sharing, adaptation, usability, sustention, talent, interaction, collaboration, happiness, and democracy [15].

The problems studied are as follows: 1) explaining the different success levels of SPI project; 2) potential benefits of capturing tacit knowledge which is easier to be ignored.

3. Conceptual Development

We focus on the explicit-tacit distinction, and develop hypotheses related to the value of capturing the two types of knowledge in SPI projects (Figure 2).

3.1. Capturing Explicit Knowledge

SPI project teams often deal with cross-functional and cross-divisional issues that warrant the use of integrative knowledge practices. In addition, the teams consist of members from diverse backgrounds who come together only for the duration of the project and, even while involved in a project, work only part-time on the project. Using combination (explicit-explicit) practices, project team leaders can help their teams sift through explicit data, drawing explicit insights about the targeted processes. In addition, internalization (explicit-tacit) practices make it possible for the explicit knowledge that is harnessed to be comprehended and absorbed by team members and people working on the processes. Such recombination of explicit knowledge and its conversion into tacit knowledge is critical for the creation of team knowledge about the working of the processes being targeted for improvement. Thus, our first set of hypotheses: (H1) cover the importance of capturing explicit knowledge for the success of SPI projects.

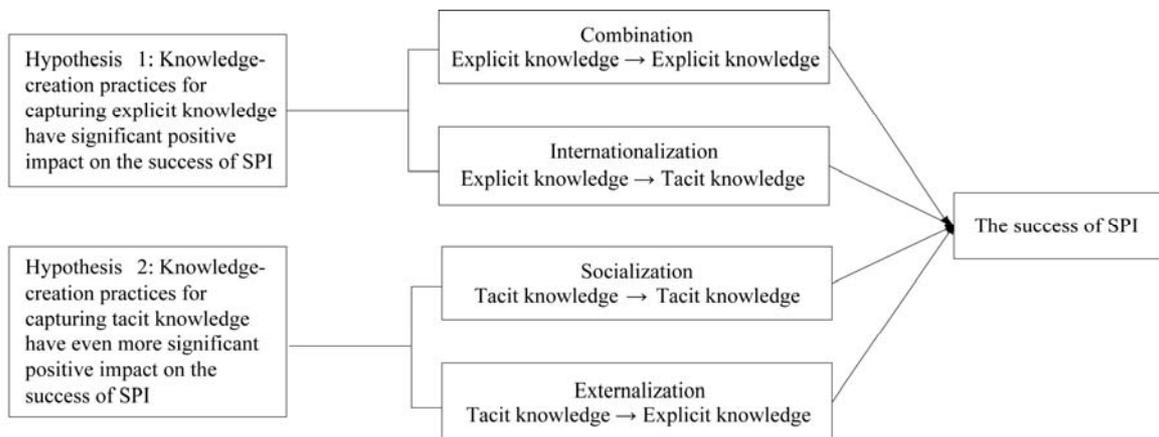


Figure 2. Proposed conceptual model and hypotheses: knowledge-creation practices as predictors of SPI project success.

H1. The following knowledge-creation practices for capturing explicit knowledge contribute significantly and positively to SPI project success:

- 1) combination (explicit-explicit knowledge), and
- 2) internalization (explicit-tacit knowledge).

3.2. Capturing Tacit Knowledge

In SPI projects, externalization (tacit-explicit) practices aid the conversion of difficult-to-codify tacit knowledge into explicit knowledge by providing templates [16], such as cause-and-effect diagrams and failure modes and effects analysis charts. Such templates serve as a common and convenient language for team members, facilitating communication and analysis, and resulting in knowledge that helps to achieve project goals.

Socialization and externalization (tacit-tacit, and tacit-explicit) practices are designed to capture the more-difficult-to capture tacit knowledge from team members that may be crucial for the success of SPI projects. The capture of such knowledge in SPI projects can provide insights that result in higher levels of process improvements than could be achieved solely through explicit-knowledge-capturing practices. Our second set of hypotheses (H2) state that the integration of tacit knowledge through socialization (tacit-tacit knowledge) and externalization (tacit-explicit knowledge) practices adds value over and above that created by concentrating solely on the utilization of explicit knowledge.

H2. The following knowledge-creation practices for capturing tacit knowledge contribute significantly and positively to SPI project success over and above the effects of practices that capture explicit knowledge:

- 1) socialization (tacit-tacit knowledge), and
- 2) externalization (tacit-explicit knowledge).

3.3. Measure Items of SPI with Knowledge Creation Theory

The heterogeneity of tacit knowledge makes a significant direct contribution to SPI success. Explicit knowledge can be transferred and stored in the database with coding, text and diagram and so on, so that it can contribute to SPI project indirectly in the long run.

3.4. Knowledge Creation and Success of SPI Project

This research proposes the following four items to measure the contribution of knowledge-creation mechanism to SPI project (**Figure 3**): 1) Continuous communication and discussion among related project members. 2) Documenting subjective ideas and requirements. 3) Project team members absorbing combination knowledge through training, and converting it to his operation knowledge. 4) To integrate knowledge from existing objective data.

4. Methods

The questionnaire for this research consists of six parts: basic information, socialization for capturing tacit knowledge, externalization for capturing tacit knowledge, internalization for capturing explicit knowledge, combination for capturing explicit knowledge, the relationship between knowledge creation and SPI. A total of 200 questionnaires were distributed, with 104 responses among which two are invalid, so there remained 102 valid ones, resulting in a valid response rate of 51%. The respondents are SPI personnel from domestic (Guangzhou, Shenzhen and Hangzhou) software organizations (companies).

4.1. Data Characteristics

The usable sample distribution characteristics were analyzed in **Table 2**. Some one has one more positions, especial in SME.

4.2. Data Validation

The Cronbach's alpha coefficients were all over 0.6, proving high validation (**Table 3**).

4.3. Factor Analysis

Three factors were extracted through factor analysis with Varimax rotation. The three factors in **Table 4** account for 64.717% of the total variance. The first factor represented items S1, S2 and S3, so it was named communication factor (SF1). The second factor represented items S4, S5 and S6 with a name crossover collaboration factor (SF2). The last factor represented items S7 and S8, and we call it work results factor (SF3).

The three factors in **Table 5** explain 77.639% of the total variance. The first factor represents items E1, E2 and E3, and we call it requirements documentation factor (EF1). The second factor represents items E5 and E6 with a name pair programming factor (EF2). The third factor represents item E4, called recording in real time (EF3).

The three factors in **Table 6** extract 72.806% of the total variance. The first factor represents items I1 and I2, and we call it on-the-job training factor (IF1). The second factor represents items I5 and I6 with a name information utilization factor (IF2). The third factor represents items I3 and I4, called document discussion (IF3).

The two factors in **Table 7** extract 75.734% of the total variance. The first factor represents items C1, C2 and C4, and we call it integrating documents factor (CF1). The second factor represents items C3 with a name procedure standardization factor (CF2).

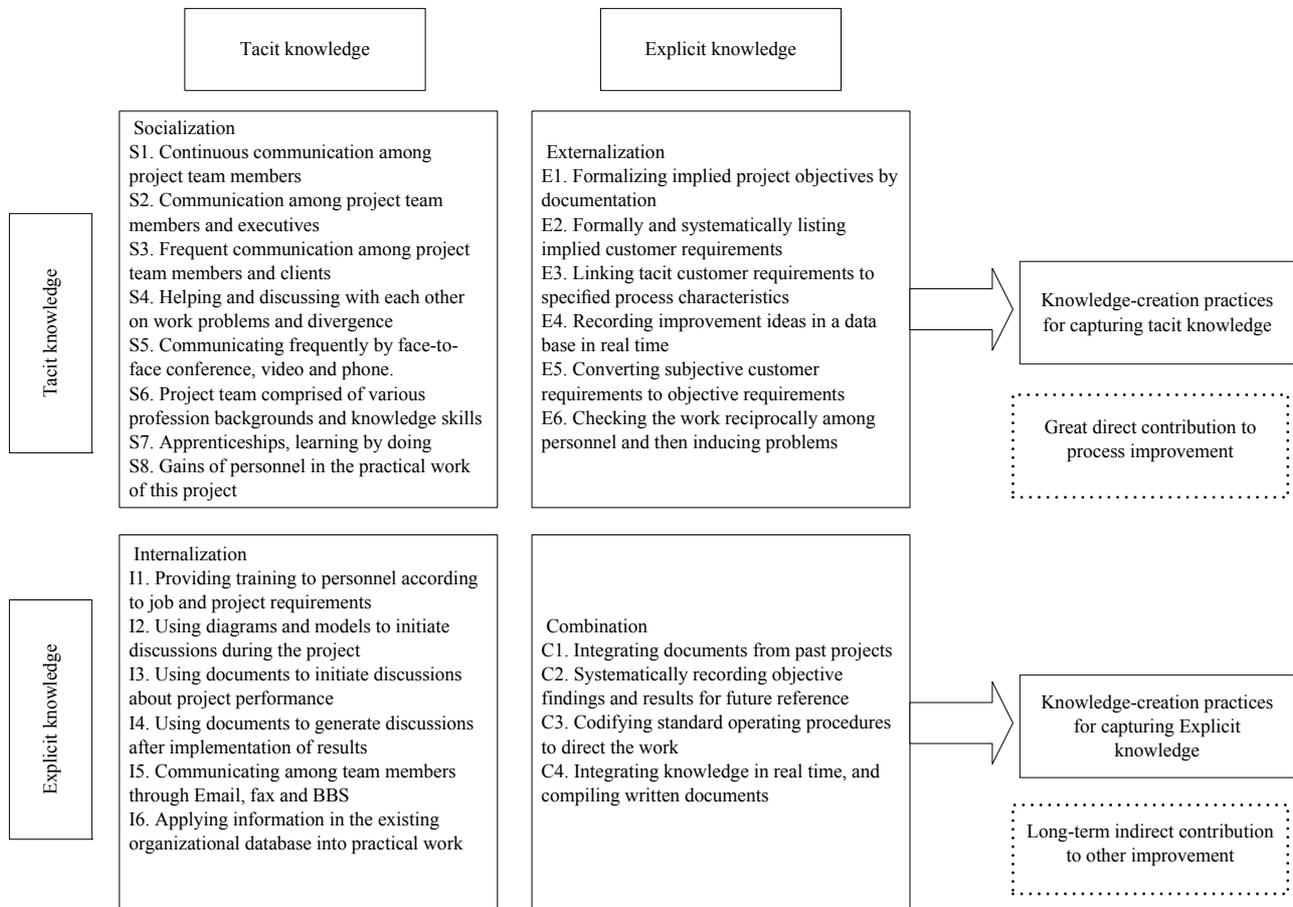


Figure 3. SPI measurement items based on knowledge creation theory.

Table 2. Sample characteristics.

Basic information of respondents		N	%
Position	Project manager	33	32.4
	Requirement personnel	15	14.7
	Designer	19	18.6
	Developer	36	35.3
	Test personnel	16	15.7
Membership of development team	≤10	14	13.7
	11 - 20	24	23.5
	21 - 50	24	23.5
	>50	40	39.2
System developed	Internal requirement	30	29.4
	External client	39	38.2
	Both	33	32.4
Whether do knowledge management	Yes	76	74.5
	No	26	25.5
Whether implement SPI	Yes	82	80.4
	No	20	19.6

Table 3. Basic information analysis.

Indicator	Items	α
Socialization for capturing tacit knowledge	S1、S2、S3、S4、S5、S6、S7	0.729
Externalization for capturing tacit knowledge	E1、E2、E3、E4、E5、E6	0.675
Internalization for capturing explicit knowledge	I1、I2、I3、I4、I5、I6	0.698
Combination for capturing explicit knowledge	C1、C2、C3、C4	0.695
Knowledge creation	Q1、Q2、Q3、Q4	0.614

Table 4. Rotated factor loading matrix of socialization practices.

Factors	Items	Loadings		
Communication	S3	0.784		
	S2	0.723	--	
	S1	0.646		
Crossover collaboration	S4		0.812	
	S6	--	0.756	
	S5		0.691	
Work results	S7			0.874
	S8			0.761
Accumulated variance explained		23.287%	46.471%	64.717%

Table 5. Rotated factor loading matrix of externalization practices.

Factors	Items	Loading		
Requirement documentation	E3	0.841		
	E2	0.785		
	E1	0.766		
Pair programming	E6		0.901	
	E5		0.724	
Recording in real time	E4			0.964
Accumulated variance explained		36.576%	59.048%	77.639%

Table 6. Rotated factor loading matrix of internalization practices.

Factors	Items	Loadings		
On-the-job training	I1	0.849		
	I2	0.849		
Information utilization	I5		0.868	
	I6		0.770	
Document discussion	I3			0.781
	I4			0.780
Accumulated variance explained		25.584%	50.032%	72.806%

Table 7. Rotated factor loading matrix of combination practices.

Factors	Items	Loadings	
Integrating documents	C1	0.872	
	C2	0.812	
	C4	0.584	
Procedure standardization	C3		0.947
Accumulated variance explained		44.095%	75.734%

Table 8. Rotated factor loading matrix of combination practices.

Factors	Items	Loadings	
Capturing explicit knowledge	Q3	0.916	
	Q4	0.736	
Capturing tacit knowledge	Q2		0.783
	Q1		0.723
Accumulated variance explained		35.414%	67.862%

The two factors in **Table 8** extract 67.862% of the total variance. The first factor represents items Q3 and Q4, and we call it capturing explicit knowledge factor (QF1). The second factor represents items Q1 and Q2 with a name capturing tacit knowledge factor (QF2). This is completely corresponding to the hypothesis noted previously—knowledge creation practices contribute to SPI success in both of capturing explicit knowledge and capturing tacit knowledge.

4.4. Regression Analysis

We take socialization practices and externalization practices as the independent variables in discussing the causality of capturing tacit knowledge factor. And when discussing the causality of capturing explicit knowledge factor, the independent variables are internalization practices and combination practices. Stepwise regression is used, including: 1) the regression analyses of practices and capturing tacit knowledge (**Tables 9-11**); 2) the regression analyses of practices and capturing explicit knowledge (**Tables 12-14**).

As to the significance level mentioned in this paper: $p^* > 0.05$, $p^{**} > 0.01$.

It can be illustrated in **Table 9** that communication (SF1) and capturing tacit knowledge (QF2) are significantly positively correlated ($p = 0.01$). Crossover collaboration (SF2) is also significantly ($p = 0.01$) positively correlated with capturing tacit knowledge, with the coefficient being 0.433. Work results (SF3) is significantly ($p = 0.01$) positively correlated with capturing tacit knowledge, so is pair programming (EF2) and recording in real time (EF3).

Thus, the regression is significant with $F = 8.258$ and $P = 0.005$. Meanwhile, variance inflation factor (VIF) is relatively small, indicating that the multicollinearity is not significant. Adjusted R^2 is 0.309, suggesting that the 3 indicators of knowledge transfer can explain 30.9% of the total variance. All the variables are significant with $P < 0.05$. According to the sequence of independent variables entering the regression equation, SF2 has the biggest influence on dependent variable, with SF3 and SF1 followed (**Tables 10 and 11**).

Table 9. Correlation matrix of socialization and externalization practices and capturing tacit knowledge.

Practices	SF1 Communication	SF2 Crossover collaboration	SF3 Work results	EF1 Requirements documentation	EF2 Pair programming	EF3 Recording in real time
Knowledge creation						
QF2 Capturing tacit knowledge factor	0.238**	0.433**	0.292**	0.178	0.386**	0.257**

Table 10. Effect parameters of stepwise regression.

Model	R	R ²	Adjusted R ²	Standard error	F	Sig.
1	0.423	0.188	0.180	0.906	23.103	0.000
2	0.523	0.273	0.258	0.861	11.629	0.001
3	0.574	0.330	0.309	0.831	8.258	0.005

Table 11. Regression and significance coefficients.

Factors	Regression coefficient	Standard error	Standard regression coefficient	t	Sig.	VIF
SF2	0.433	0.083	0.433	5.238	0.000	1.000
SF3	0.292	0.083	0.292	3.533	0.001	1.000
SF1	0.238	0.083	0.238	2.874	0.005	1.000
Standard regression equation	V = 0.433 * SF2 + 0.292 * SF3 + 0.238 * SF1					

Table 12. Correlation matrix of internalization and combination practices and capturing explicit knowledge.

Practices	IF1 On-the-job training	IF2 Information utilization	IF3 Document discussion	CF1 Integrating documents	CF2 Procedure standardization
Knowledge creation					
QF1 Capturing explicit knowledge	0.338**	0.136	0.286**	0.272**	0.341**

Table 13. Effect parameters of stepwise regression.

Model	R	R ²	Adjusted R ²	Standard error	F	Sig.
1	0.341	0.116	0.107	0.944	13.133	0.000
2	0.436	0.190	0.174	0.909	9.051	0.003
3	0.476	0.226	0.203	0.893	4.604	0.034

Table 14. Regression and significance coefficients.

Factors	Regression coefficient	Standard error	Standard regression coefficient	t	Sig.	VIF
CF2	0.289	0.092	0.289	3.135	0.002	1.074
CF1	0.215	0.093	0.215	2.317	0.023	1.090
IF3	0.206	0.096	0.206	2.146	0.034	1.164
Standard regression equation	V = 0.289 * CF2 + 0.215 * CF1 + 0.206 * IF3					

On-the-job training (IF1) is significantly ($p = 0.01$) positively correlated with Capturing explicit knowledge, so is information utilization (IF2) and Integrating documents (CF1). Procedure standardization (CF2) is significantly ($p = 0.01$) positively correlated with capturing explicit knowledge, with the correlation coefficient reaching 0.341 (Table 12). $F = 4.604$ and $P = 0.034$ suggest significant regression effect. And the variance inflation fac-

tor (VIF) is relatively small, indicating that the multicollinearity is not significant.

Adjusted R^2 is 0.203, suggesting that the 3 indicators of knowledge transfer can explain 20.3% of the total variance. All the variables are significant with $P < 0.05$. According to the sequence of independent variables entering the regression equation, CF2 has the biggest influence on dependent variable, with CF1 and IF3 fol-

lowed (Tables 13 and 14).

5. Analyses and Results

Whether factors will be in regression equation depends on whether it is significant in descriptive statistics and whether it correlated with corresponding variables. It can be found that communication factor (SF1) is significant in descriptive statistics, correlation analysis and regression analysis, and communication among project team members and clients has significant effect on SPI. Crossover collaboration factor (SF2) is significant in descrip-

tive statistics, correlation analysis and regression analysis, and helping and discussing with each other on work problems and divergence has significant influence. Integrating documents (CF1) is significant in descriptive statistics, correlation analysis and regression analysis, and integrating documents from past projects to analyze current projects and integrating knowledge in real time and compiling written documents contribute significantly to SPI (Table 15). The results can be illustrated in Figure 4.

The full line in the figure represents the biggest contribution, with dashed line the second biggest and the dot-

Table 15. Test and statistic results of related hypotheses.

Factors	Items	Descriptive statistics	Correlation analysis	Regression analysis
Communication (SF1)	S1	✓		
	S2		◇	☆
	S3	✓		
Crossover collaboration (SF2)	S4	✓		
	S5		◇◇	☆
	S6			
Work results (SF3)	S7		◇	☆
	S8			
Requirements documentation (EF1)	E1	✓		
	E2	✓		
	E3			
Pair programming (EF2)	E5		◇◇	
	E6			
Recording in real time (EF3)	E4		◇	
On-the-job training (IF1)	I1	✓		
	I2	✓	□□	
Information utilization (IF2)	I5			
	I6			
Document discussion (IF3)	I3			
	I4		□	☆
Integrating documents (CF1)	C1	✓		
	C2	✓	□	☆
	C4			
Procedure standardization (CF2)	C3		□□	☆
Capturing explicit knowledge (QF1)	Q3	✓		
	Q4.			
Capturing tacit knowledge (QF2)	Q1	✓		
	Q2			

Standard regression equation:
 $V = 0.289 * CF2 + 0.215 * CF1 + 0.206 * IF3$

Standard regression equation:
 $V = 0.433 * SF2 + 0.292 * SF3 + 0.238 * SF1$

✓ indicating that descriptive statistic results are significant; ◇ suggests that this factor is correlated with capturing tacit knowledge, and ◇◇ indicates stronger correlation; □ suggests this factor is correlated with capturing explicit knowledge, □□ indicates stronger correlation; ☆ suggests that this factor is in the regression equation.

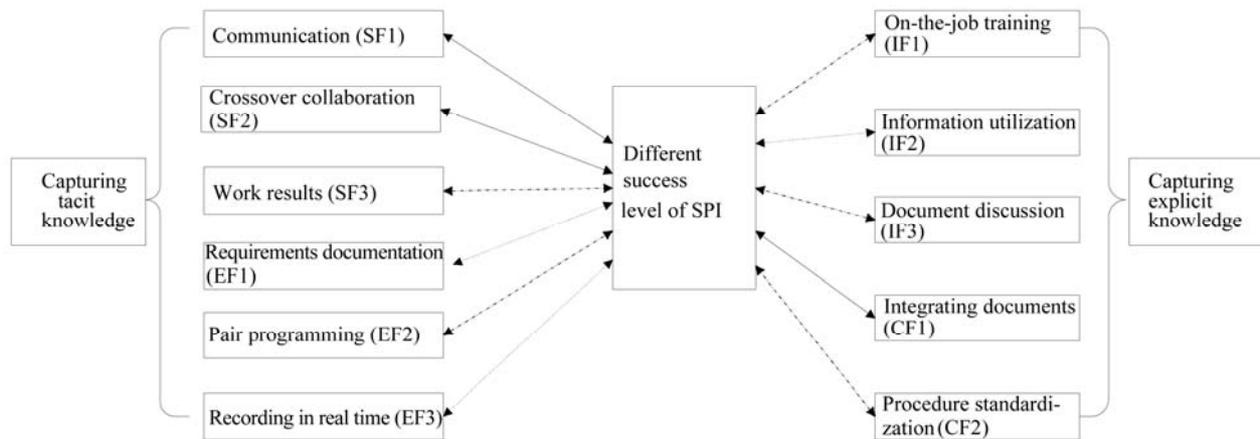


Figure 4. Impact of each factor on the success level of SPI.

ted line the least. IF2 is not in the equation, whose insignificant influence is represented with dotted line. Members can directly and face to face perceive the intention of peers, especially the tacit knowledge which is difficult to express, in communication and collaboration, and this can seldom be achieved with other approaches. Then, acquire tacit knowledge that is not codified in text and files through practical work. Pair programming among members can facilitate the disclosure of invisible tacit knowledge in mutual check, and then members absorb the experience. Making the invisible tacit knowledge visible has great effect on the success of SPI.

6. Conclusions

The two conclusions can provide guidance to decision-making in software organizations. Communication among members, frequent crossover collaboration during the practical work and integrating related documents have great contribution to the success of SPI. And the following suggestions are proposed: 1) Encouraging personnel to communicate. Motivate personnel to communicate by all means, and staff in different levels should commit to crossover collaboration. Some physical or mental compensation should be offered to drive experienced personnel to share their knowledge and their knowledge achievements should be respected, because exchange and assistance would take time and energy. 2) Integrating documents in real time. Integrate the information on the project in real time and generate new logical knowledge which can be reused and taken for reference. This will be useful in guiding and enhancing efficiency.

Our main finding is that practices used in SPI projects to extract team-member knowledge can be quite valuable for SPI project success. Exploration of additional nuances of relationships between knowledge-creation approaches and SPI requires further research. Future investigation should better understand the state of evolution, or matur-

ity, of a firm's SPI initiative and the resultant impact on knowledge-creation practice selection and effectiveness. In addition, the scales we created for capturing explicit and tacit knowledge represent a contribution, and can be used to address questions on the types of projects for which it would be more beneficial to focus on one type of knowledge capture over another. The approach illustrates that process improvement in general, and SPI in particular, benefit from perspective and analysis at the project level of observation. The insights from this study on the role of knowledge creation in process improvement provide practical guidance for SPI project leaders and other managers of SPI projects and initiatives, especially in regards to the importance of capturing tacit knowledge.

7. Acknowledgements

Thanks for the helpful discussion with Mr. Jianzhang Li and Mr. Chuanbo Zhang, and my student Qingjing Liu hard working.

REFERENCES

- [1] I. Nonaka, "The Knowledge-Creating Company," *Harvard Business Review*, Vol. 69, No. 6, 1991, pp. 96-104.
- [2] M. Polanyi, "The Tacit Dimension," Doubleday, Garden City, New York, 1966.
- [3] W. S. Humphrey, "Managing the Software Process," Addison-Wesley, Reading, 1989.
- [4] A. Fuggetta, "Software Process: A Roadmap," *Proceedings of the Conference on the Future of Software Engineering*, Limerick, 4-11 June 2000, pp. 25-34.
- [5] J. P. Wan and J. M. Yang, "Knowledge Management in Software Process Improvement," *Application Research of Computer*, Vol. 19, No. 5, 2002, pp. 1-3.
- [6] M. Blanco, P. Gutiérrez and G. Satriani, "SPI Patterns: Learning from Experience," *IEEE Software*, Vol. 18, No. 3, 2001, pp. 28-35. [doi:10.1109/52.922722](https://doi.org/10.1109/52.922722)

- [7] J. D. Herbsleb and D. R. Goldenson, "A Systematic Survey of CMM Experience and Results," *Proceedings of the 18th International Conference on Software Engineering*, Berlin, March 1996, pp. 323-330.
- [8] R. S. Pressman, "Software Engineering: A Practitioner's Approach," 5th Edition, McGraw-Hill Companies, Inc., New York, 2001, p. 19.
- [9] J. P. Wan, "Research on Software Product Support Structure," *Journal of Software Engineering and Applications*, Vol. 2, No. 3, 2009, pp. 174-194. [doi:10.4236/jsea.2009.23025](https://doi.org/10.4236/jsea.2009.23025)
- [10] G. Anand, P. T. Ward and M. V. Tatikonda, "Role of Explicit and Tacit Knowledge in Six Sigma Projects: An Empirical Examination of Differential Project Success," *Journal of Operations Management*, Vol. 28, No. 4, 2010, pp. 303-315. [doi:10.1016/j.jom.2009.10.003](https://doi.org/10.1016/j.jom.2009.10.003)
- [11] I. Nonaka and G. V. Krogh, "Perspective—Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory," *Organization Science*, Vol. 20, No. 3, 2009, pp. 635-652. [doi:10.1287/orsc.1080.0412](https://doi.org/10.1287/orsc.1080.0412)
- [12] J. P. Wan, Q. J. Liu, D. J. Li and H. B. Xu, "Research on Knowledge Transfer Influencing Factors in Software Process Improvement," *Journal of Software Engineering and Applications*, Vol. 3, No. 2, 2010, pp. 134-140. [doi:10.4236/jsea.2010.32017](https://doi.org/10.4236/jsea.2010.32017)
- [13] J. P. Wan and R. Wang, "The Exploratory Analysis on Knowledge Creation Effective Factors in Software Requirement Development," *Journal of Software Engineering and Applications*, Vol. 3, No. 6, 2010, pp. 580-587.
- [14] J. P. Wan, H. Zhang, D. Wan and D. Y. Huang, "Research on Knowledge Creation in Software Requirement Development," *Journal of Software Engineering and Applications*, Vol. 3, No. 5, 2010, pp. 487-494. [doi:10.4236/jsea.2010.35055](https://doi.org/10.4236/jsea.2010.35055)
- [15] J. P. Wan and R. Wang, "Empirical Research on Critical Success Factors of Agile Software Process Improvement," *Journal of Software Engineering and Applications*, Vol. 3, No. 12, 2010, pp. 1131-1140. [doi:10.4236/jsea.2010.312132](https://doi.org/10.4236/jsea.2010.312132)
- [16] R. J. Jensen and G. Szulanski, "Template Use and the Effectiveness of Knowledge Transfer," *Management Science*, Vol. 53, No. 11, 2007, pp. 1716-1730. [doi:10.1287/mnsc.1070.0740](https://doi.org/10.1287/mnsc.1070.0740)