Computers in Human Behavior 25 (2009) 429–438

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Creative self-efficacy and its factors: An empirical study of information system analysts and programmers

Heng-Li Yang *, Hsiu-Hua Cheng

Department of MIS, National Cheng-Chi University, 64, Sec. 2, Chihnan Rd., Mucha Dist., Taipei, 116, Taiwan

ARTICLE INFO

Article history:
Available online 17 November 2008

Keywords:
Information systems developers
Creative self-efficacy
Social network
Information technology skills

ABSTRACT

Based on a survey of 94 information systems developers, this study explored how personal factors (i.e. computer self-efficacy and domain-specific information technology skills), contextual factors (i.e. strength of ties and degree centrality) and creative self-efficacy are related. Regression analysis results demonstrate that system analysts and programmers differ in terms of influencing factors on creative self-efficacy. Domain-specific skills were the main influence in the system analyst model, followed by degree centrality. In comparison, degree centrality was the only influence in the programmer model. Degree centrality exerted a negative influence in both groups. Additionally, among system analysts, the strength of ties slightly influenced creative self-efficacy, while computer self-efficiency and domain-specific information technology skills exerted only small influences on programmers.

1. Introduction

Organizations must innovate to maintain their advantages in the face of environmental stresses (Hanninen & Kauranen, 2007; Lee & Chang, 2007; Morcillo, Rodriguez-Anton, & Rubio, 2007). However, organizational innovation is based on individual creativity (Woodman, Sawyer, & Griffin, 1993). Amabile (1997) indicated that one of the important factors influencing creativity is individual motivation. Ford (1996) further observed strong self-efficacy is an important motivational component in developing creativity. Bandura (1997) also illustrated that self-efficacy belief is a major impetus for creative individual actions. Bandura (1986) defined self-efficacy as "People's judgments of their capabilities to organize and execute courses of action required to attain designated types of performances". Restated, self-efficacy describes individual belief in their capabilities to perform a particular behavior. Creativity is a domain-specific, subjective judgment regarding the novelty and value of an outcome of a specific action (Ford, 1996). People with high self-efficacy have intrinsic driver to develop novel and useful ideas such as creativity. Given the potential link between self-efficacy and creativity, Tierney and Farmer (2002) employed the model of self-efficacy developed by Gist and Mitchell (1992) to present a creative self-efficacy construct, and discussed its potential predictors (personal and contextual resources) and relationship with creative performance. Based on their results, creative self-efficacy appears to provide strong efficacy beliefs for enhancing creative behaviors.

Bandura (1997) further noted that efficacy views can be general or specific. Specificity thus can be increased to transform self-efficacy from a general view into a specific measure that can be applied in a narrow domain such as creativity area. Although Tierney and Farmer noted the significance of personal and contextual resources for forecasting individual creative self-efficacy, issues involving specific domains (e.g. information system) have been comparatively neglected. The creativity of information systems (IS) developers is a source of innovation for software firms. To facilitate innovation, software companies must understand the factors that contribute to the belief of IS developers who can be creative in their work roles. Accordingly, the purpose of this study is to explore the correlations and associations between several possible contextual/personal factors and creative self-efficacy of IS developers.

2. Literature review and hypotheses development

2.1. Creative self-efficacy of information systems developers

2.1.1. Information systems developers

Prior researches indicated that there are different roles of IS staff owing to their tasks. According to Lee, Trauth, and Farwell (1995), among the several types of IS professionals include programmers, technical specialists, business/system analysts, end-user support consultants and computer operators, and data entry clerks. Moreover, Todd, McKeen, and Gallupe (1995) defined IS professionals as including programmers, system analysts and IS managers. This study focuses on examining creative self-efficacy

* Corresponding author. Tel.: +886 2 29387651.
E-mail addresses: yanh@nccu.edu.tw (H.-L. Yang), d92507@mis.nccu.edu.tw (H.-H. Cheng).

0747-5632/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chb.2008.10.005

in software development and concentrates on two main categories of IS professionals, system analysts and programmers. This study thus only considers two major categories of IS professionals, system analysts and programmers. The system analyst is responsible for dividing complex problems into several small and simple tasks, and for evaluating and designing system specifications. Meanwhile, the programmer is responsible for developing a system platform or components, and for coding and testing programs according to system analyst defined specifications.

Moreover, this study only addresses the development of software packages because IS developers, who develop software packages, have more autonomy and more flexibility to develop creativity than other IS developers developing custom-made information systems. Grudin (1991) explained that software package design and information systems development differ in the relationships between developers and users. IS developers need to interact with users directly to understand their requirements when developing information systems. However, most software package developers are not directly connected to end-users, instead indirectly communicating with them through staff who sell products or provide services to customers (Keil & Carmel, 1995). For example, during software package development, system analysts can use their experience and knowledge to plan specifications; programmers have more autonomy to choose appropriate technological trends. However, custom-made IS developers have to follow user's requirement to develop systems. Clearly, to meet market demand, IS developers of software package have more flexibility in incubating ideas or procedures that are new, original, suitable or useful for creativity (Oldham & Cummings, 1996).

2.1.2. Creative self-efficacy of IS developers of software packages

Bandura (1986) proposed self-efficacy through the frame of social cognitive theory, and defined self-efficacy as follow:

> Self-efficacy as people's judgments of their capabilities to organize and execute courses of action required to attain designated types of performances. It is concerned not with the skills one has but with judgments of what one can do with whatever skills one possesses.

However, self-efficacy reflects the *general* beliefs of individuals in their abilities across domains (Chen, Gully, & Eden, 2001). To study creativity performance, which implies novel and valuable outcomes, Tierney and Farmer (2002) developed *specific* self-efficacy, namely creativity self-efficacy, to judge capacities in a narrow domain. Creative self-efficacy differs from creativity. Creativity indicates the generation of domain-specific, novel, and useful outcomes. In contrast, creative self-efficacy denotes the belief that one has the ability to produce creative outcomes for the jobs. Self-efficacy is not concerned with past actions, but rather with the judgments regarding what could be done in the future (Compeau & Higgins, 1995).

Although creativity and creative self-efficacy naturally differ, they are related. Oldham and Cummings (1996) defined the results of creativity to include outcomes, ideas or procedures that are new, original, suitable or valuable. Amabile (1988) defined creativity as novel and useful results. Creative self-efficacy describes the belief of an individual in his ability to generate creativity. Tierney and Farmer (2002) indicated creative self-efficacy as key influence on creativity. Creative self-efficacy is a form of self-evaluation that influences decisions regarding creative behaviors to undertake, the amount of effort and the persistence level when encountering challenges (Bandura, 1977).

Different tasks of IS developers might require different types of creativity and thus have different creative self-efficacy. This study thus defines the creative self-efficacy of IS developers as follows.

1. The creative self-efficacy of a system analyst refers to the belief of an individual in his ability to develop novel and useful ideas regarding system flows or system specifications when planning and analyzing software packages.
2. The creative self-efficacy of a programmer refers to the belief that one has the ability to develop new and useful ideas or outcomes of platforms, components or programs during software package development.

2.2. Possible factors to creative self-efficacy of IS developers

Tierney and Farmer (2002) stated that individuals assess their *personal and contextual resources* to form personal efficacy judgments. Therefore, while proposing the model for exploring the creative self-efficacy of IS developers, this study addresses not only personal resources (computer self-efficacy and domain-specific IT skills), but also contextual resources (the strength of ties and network positions).

2.2.1. Computer self-efficacy and creative self-efficacy

Computer self-efficacy indicates a judgment regarding the ability of an individual to use a computer (Compeau & Higgins, 1995). Gist, Schwoerer, and Rosen (1989) expressed that computer self-efficacy is positively correlated with computer-related performance. Shin (2006) also demonstrated that computer self-efficacy could significantly, positively and directly influence the system usage. In the computing field, an individual with higher computer self-efficacy generally has greater confidence in using advanced software (Fagan, Neill, & Wooldridge, 2004). Amabile (1988) indicated that performance capability is required to achieve creative capability in a domain. An IS developer with higher computer self-efficacy has more confidence in his computer using ability; is unrestricted by existing IT; can quickly learn new skills, and can do his job well. IS workers with high computer self-efficacy may have high creative self-efficacy. Therefore, we posit that a correlation exists between computer self-efficacy and creative self-efficacy in IS developers.

The dissimilarity in the jobs of system analysts and programmers may moderate the relationship between computer self-efficacy and creative self-efficacy. Programmers stress technical skills, while system analysts value systems analysis and design skills (Todd et al., 1995). Although system analysts rarely adopt IT for direct software package development, they still need to understand the degree to which using IT to develop software packages can support business flow. Programmers apply IT for software package development. Programmers with high computer self-efficacy in encountering work difficulties are more confident in learning diverse IT and using different IT to effectively perform their work, thus possibly upgrading their beliefs regarding creativity. Therefore, Hypotheses 1a and 1b are presented with regard to the relationships between computer self-efficacy and creative self-efficacy.

Hypothesis 1a. A correlation exists between computer self-efficacy and creative self-efficacy in system analysts.

Hypothesis 1b. A correlation exists between computer self-efficacy and creative self-efficacy in programmers.

2.2.2. Domain-specific IT skills and creative self-efficacy

Increasing domain knowledge can develop or verify workable ideas, and can also influence creativity (Simonton, 1999). Sternberg and Lubart (1995) and Gardner (1993) observed that domain knowledge influences creativity. Moreover, Gist and Mitchell

(1992) concluded that knowledge drives efficacy judgment. Tierney and Farmer (2002) also demonstrated a correlation between the role of knowledge and creative self-efficacy. IS developers with more IT knowledge and skills are more likely to feel confident that they can be creative in their work roles. Therefore, this study hypothesizes the existence of a correlation between domain-specific IT skills and creative self-efficacy in developers.

The computer skills required by IS developers differ according to the domain in which they are working (Benbasat, Dexter, & Mantha, 1980). Researchers expressed the skills required for system analysts are technical, business and system skills (Lee, 2005) and systems analysis and communication skills (Cheney & Lyons, 1980). System analysts must not only to understand existing IT, but must also to be able to transfer requirements into system specifications. System analysts with highly domain-specific IT skills have more confidence in using their IT knowledge to design novel and effective system specifications. In contrast, Todd et al. (1995) showed that programmers require technical knowledge and skills more than business and system knowledge. Watson, Young, Miranda, Robichaux, and Seerley (1990) concluded that programmers should carefully consider applications of programming languages, as well as databases and operating systems. When confronting development problems, programmers who deeply understand the technology of development software have more confidence in adopting diverse language syntax and logic to solve problems by developing creative concepts.

This study defines the domain-specific IT skills of two IS developers as follows.

1. The domain-specific IT skills of a system analyst include systems analysis and design.
2. The domain-specific IT skills of a programmer include knowledge of programming languages, databases and operating systems.

Additionally, these relationships are studied in relation to the following hypotheses.

Hypothesis 2a. A correlation exists between domain-specific IT skills of systems analysts and their creative self-efficacy.

Hypothesis 2b. A correlation exists between domain-specific IT skills of programmers and their creative self-efficacy.

2.2.3. The strength of ties and creativity

Gist and Mitchell (1992) indicated that people may obtain information to form self-efficacy views based on observation of others. Bandura (1988) showed that "people partly judge their capabilities in comparison with others". Individuals can observe the behaviors of others through their interpersonal interactions. Furthermore, people can acquire diverse information via their interpersonal networks. Abundant and high quality information helps individuals to think broadly and combine existing approaches into something new and unique (Perry-Smith & Shalley, 2003). The social network theory defines human interaction as a tie (Granovetter, 1973; Levin & Cross, 2004). Interaction in a social network not only indicates communication, but also illustrates that a personal network is formed by interaction among individuals (Freeman, 1979). This study thus addresses social network characteristics as critical contextual resources.

The strength of ties is the first proposed contextual resource. Here a tie refers to a connection between two actors, such as individuals, groups or organizations (Wasserman & Faust, 1994). Ties can be categorized as strong or weak. Levin and Cross (2004) adopt closeness of a working relationship, frequency of communication and frequency of interaction to measure the strength of a tie. Granovetter (1973) presented a function of the level of interaction, emotional intensity, and reciprocity that occurs between two individuals to judge tie strength.

Weak and strong ties differ. Weak ties may help people to connect to others with dissimilar views, perspectives, interests and approaches to solve problems (Coser, 1975). People with more weak ties can link more extensive parties and can acquire more non-redundant information (Burt, 1997; Granovetter, 1973). Conversely, people can share similar information with others if they have strong ties among them (Ibarra, 1992). Strong ties and trust have been considered significant, since people wish to share complex and private information (Hansen, 1999). Weak ties provide positive benefits in situations involving explicit information. However, the importance of strong ties increases in situations involving uncertain or unclear information (Smith, Collins, & Clark, 2005). Hansen showed the importance of strong ties in obtaining complex and tacit knowledge. People with strong ties have greater willingness to exchange information and collaborate to obtain multiple benefits (Krackhardt, 1992).

Owing to the complexity of software development, IS developers frequently consult closely and exchange complex and tacit ideas to perform non-routine tasks. IS staff develop strong ties by continually interacting with each other. During this interaction, they can acquire more private, fuzzy and complex information that is useful in generating creative self-efficacy. Additionally, IS workers can also appraise their own creative self-efficacy after observing the creative achievements of their colleagues. This study thus conjectures that a correlation exists between IS developers' strength of ties and their creative self-efficacy.

The communicative subjects and contexts differ between system analysts and programmers. System analysts must consult closely with one another to establish a common view of systems design and to link system flows. Although programmers can understand the requirements of software packages based on system specifications, they must still interact frequently to clarify system flows, and to understand system frameworks designed by programmers cooperating with one another. Programmers communicate with others to obtain program syntax and logic when coding components, programming, or debugging. Different communicative subjects and communicative contexts lead to different tie strengths, which help shape perceptions of creative beliefs. Accordingly, Hypotheses 3a and 3b are presented.

Hypothesis 3a. A correlation exists between the strength of ties among system analysts and their creative self-efficacy.

Hypothesis 3b. A correlation exists between the strength of ties among programmers and their creative self-efficacy.

2.2.4. Degree centrality of social networks and creativity

Consistent with the view that people gather useful information for shaping judgments of efficacy in their work environment (Gist & Mitchell, 1992), this study recommends the degree centrality of a social network as the second contextual resource for creative self-efficacy. Degree centrality indicates the degree to which an actor interacts directly with other actors (Burt, 1982). Wasserman and Faust (1994) characterized the function of degree centrality as follows.

$$C_D(n_i) = \left(\sum_j x_{ij}\right) \Big/ (g-1)$$

$C_D\ (n_i)$ as degree centrality index.
$\sum_j x_{ij}$ as the amount of direct ties that actor *i has.*
"*g*" as group size.

Two types of ties exist in social networks, direct and indirect. Direct ties refer to the situation where two actors connect directly with each other, while indirect ties refer to the situation where two actors link with one another via brokers. The properties of direct ties are explained as follows. Direct ties can enhance knowledge sharing (Berg, Duncan, & Friedman, 1982). Individuals with numerous direct ties can rapidly obtain unique and varied information, enabling them to integrate and exchange knowledge (Burt, 1992; Nahapiet & Ghoshal, 1998). Furthermore, direct ties can facilitate the rapid obtaining of private information (Uzzi, 1996). Transferring tacit knowledge requires relying on direct ties (Hansen, 1999). However, direct ties are less important in situations involving clear and easily acquired information. While direct ties possess many advantages, they are costly to maintain (Hansen, 1999).

The features of indirect ties are expressed as follows. Indirect ties can provide access to rich information. For instance, a company can obtain knowledge or information from the partners of their partners (Gulati & Garguilo, 1999). People facing uncertainty can obtain most information through indirect ties (Shane & Cable, 2002). Indirect ties can help people to acquire information from unconnected people to access rich information (Burt, 1997). People spend relatively little much time maintaining indirect ties (Nahapiet & Ghoshal, 1998). Although indirect ties offer many benefits, they can also distort information (March & Simon, 1958). People often misunderstand information, forget details or filter information during information transfer, causing information distortion.

Direct and indirect ties carry different information and can have different effects on creative self-efficacy. People can judge their creative capabilities via information obtained through direct and indirect ties. However, most of IS development tasks are complex and non-routine. IS developers often depend on direct ties to obtain private ideas and information that are frequently complex and tacit from their colleagues for undertaking tasks. Thus, IS developers with more direct ties thus can quickly access more complementary and tacit information, which is useful in shaping their creative self-efficacy. Accordingly, we conjecture that a correlation exists between degree centrality and creative self-efficacy of IS developers.

System analysts and programmers communicate in different contexts. The primary tasks of system analysts are to analyze business flows and produce requirement specifications. Systems analysts frequently discuss with other analysts during the system analysis stage, owing to the system functions for which they are responsible being related. For example, system analysts consult with other system analysts to exchange ideas regarding the best methods of passing and receiving system data, and integrate flows to plan consistent system flows and system data. Programmers who face development problems frequently ask others who have previously experienced problems for help in developing a solution. The tasks of system analysts and programmers differ, so the subjects and contents of their interactions also differ. Consequently, the relationships between the degree centrality and creative self-efficacy differ between the two classes of IS developers. Thus, Hypotheses 4a and 4b are proposed:

Hypothesis 4a. A correlation exists between degree centrality and creative self-efficacy in system analysts.

Hypothesis 4b. A correlation exists between degree centrality and creative self-efficacy in programmers.

Based on the above discussions, this study develops a research model, as illustrated in Fig. 1 to address the possible correlations between personal and contextual resources, and the creative self-efficacy of IS system analysts and programmers developing software packages.

3. Methods

3.1. Subjects

This study obtained the cooperation of the top executives of Taiwanese software firm α and conducted an investigation to explore the creative self-efficacy of IS developers during 2006. Company α was established in 1982, and currently has capital of approximately US$63 million. The company has around 1000 employees in Taiwan, and 700 employees in China. The tasks of the company are to provide customers with software packages. The customers are involved in various manufacturing, banking, and service sector industries.

Participants in this study included 94 developers whose work involved software package development. All participants were employees of the product departments of company α. The participants included 34 system analysts and 60 programmers. Over 68% had graduated from information-related college departments, and 76% had at least three years experience of related work. Broken down according to gender, the sample was 38% female at 62% male. Participants were asked by company executives to participate in the survey to contribute to academic research and were not compensated. To our knowledge, this was the first survey of social network and creativity conducted in these production departments.

3.2. Investigation procedures

The questionnaire was developed using the following procedure. First, for customizing the survey questionnaire (particularly the section on domain-specific skills), product department managers were interviewed. Semi-structured questions were

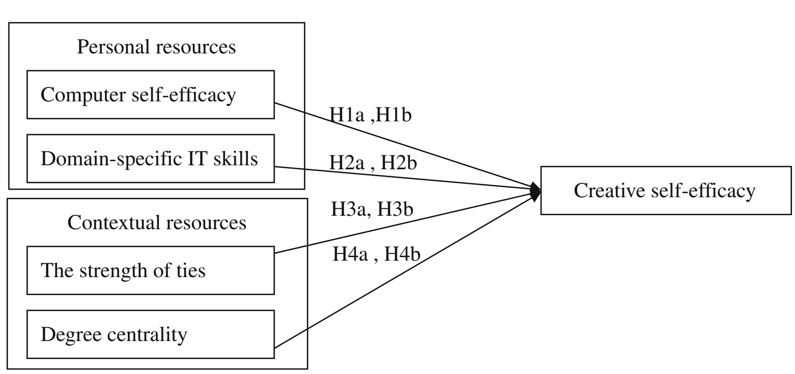

Fig. 1. Research model.

asked to understand how departments operated, how IS developers worked together, the IT skills of IS developers, and the names of the developers. An initial set of domain-specific IT skills items was thus obtained. Second, software industry managers and experts were invited to verify the face and content validity of the questionnaire. Third, the items were adjusted based on the recommendations from managers and experts. A pretest was then performed before the final survey. The pretest involved 26 students from a university Department of Information Systems. Three weeks were spent distributing and collecting the pretest questionnaires. The domain-specific IT skills items were adjusted after calculating the item-to-total correlation. Table 1 lists the source, modification and reliability of the pretest questionnaire.

The questionnaires were sent to four managers, who then distributed them to all IS developers in their teams. Each participant was asked to complete a set of self-reported questionnaires dealing with areas that included computer self-efficacy, domain-specific IT skills, the strength of ties and creative self-efficacy. Then, participants were further required to seal the completed questionnaire before returning it. Because these two types of IS developers possess different IT skills and communication subjects, they used different questionnaire items. System analysts were asked to answer items related to of domain-specific IT skills dealing with systems analysis and design. Programmers were instructed to answer items involving of domain-specific IT skills emphasizing program development. All 94 participants completed and returned the questionnaires, providing a valid response rate of 100%.

3.3. Measures

Table 1 lists the sources of the measures used in this study. Computer self-efficacy indicates the judgment and confidence of IS developers in their ability to accomplish computer-related tasks, and was measured using items proposed by Compeau and Higgins (1995). A sample item was, "I could complete the job using the information technique if there was no one around to tell me what to do as I go". In the study of Compeau and Higgins, the internal consistency reliability coefficient of computer self-efficacy was .95, while the discriminant validity coefficient of computer self-efficacy was .81, exceeding the correlations among computer self-efficacy and other constructs. The results obtained by Compeau and Higgins already indicated good reliability and discriminant validity for computer self-efficacy.

Domain-specific IT skills describe the specific knowledge and technologies required for IS developers to accomplish tasks. This study developed sections of the questionnaire comprising the following dimensions: generic (5 items), systems analysis (6 items), systems design (2 items), basic/advanced programming (12 items), operating systems (2 items) and database skills (3 items). The following is a sample item from the database skills category "I have knowledge of stored procedures and database triggers". The domain-specific IT skills of each respondent were calculated using the IT skill weightings as suggested by the managers.

The strength of ties indicates the strength of interaction between an IS developer with others developing the same software package, and was measured using three items, drawn from Levin and Cross (2004). Levin and Cross reported that in their study the Cronbach α of tie strength items was .90, and their factor analysis results also indicated good validity for the tie strength construct. In this study, every participant was provided a list of names of other IS developers. A sample item was, "How close was my working relationship with each of the others with whom I was working?" Participants answered these items with reference to their interpersonal relationships within the network.

Degree centrality refers to the amount of direct ties of an IS developer. The tie strength matrix was inputted into UCINET software to calculate the degree of centrality for each IS developer.

Creative self-efficacy indicates the extent to which the judgment and confidence of an IS developer assists in developing novel and valuable ideas. The construct of creative self-efficacy was measured using 13 items, as listed in Table 2, adapted from Zhou and

Table 2
Items of creative self-efficacy in the final questionnaire.

Item
1. The belief that I would suggest new ways to achieve goal or objectives
2. The belief that I would come up with new and practical ideas to improve performance
3. The belief that I could search out new technologies, processes, techniques, and/or product ideas
4. The belief that I would suggest new ways to increase quality
5. The belief that I would be a good source of creative ideas
6. The belief that I would be not afraid to take risks
7. The belief that I would promote and champion ideas to others
8. The belief that I would exhibit creativity on the job when given the opportunity to
9. The belief that I would develop adequate plans and schedules for the implementation of new ideas
10. The belief that I would often have new and innovative ideas
11. The belief that I would often come up with creative solutions to problems
12. The belief that I would often have a fresh approach to problems
13. The belief that I would suggest new ways of performing work tasks

Table 1
The sources and reliability of the pretest questionnaire.

Variable	Source	Number of remaining items after pre-testing	Cronbach α of remaining items
Computer self-efficacy	Items were drawn from Compeau and Higgins (1995)	10	.87
Generic skills	Items were developed by this study	5 after deleting few items which 'item to total' was low	.93
Systems analysis skills	Some items were drawn form Green (1989) and others were developed by this study	6 after deleting one item which 'item to total' was low	.87
Systems design skills	Items were developed by this study	2 after deleting one item which 'item to total' was low	.81
Basic programming skills	Items were developed by this study	5 after deleting one item which 'item to total' was low	.78
Advanced programming skills	Items were developed by this study	7	.91
Operating systems skills	Items were developed by this study	2	.69
Database skills	Items were developed by this study	3	.83
Creative self-efficacy	Items were adapted from Zhou and George (2001)	13	.93
The strength of ties	Items were drawn from Levin and Cross (2004)	3	.91

Note: degree centrality is computed by UCINET software.

George (2001). In their study, Zhou and George measured creativity and obtained a Cronbach α of .96. In this study, the wording of the present investigation thus was changed to measure the creative self-efficacy situation. A sample item is "The belief that I would suggest new ways to achieve goals or objectives".

4. Results

4.1. Reliability, validity and descriptive statistics of variables

The Cronbach α levels, as listed in Table 3 all exceeded .60, indicating internal measurement consistency in this survey. The construct validity of the measures for research variables was validated via factor analysis. Factor analyses were separately performed on measuring items belonging to variables, as recommended by Kerlinger (1978). Eigen-values exceeding 1 and factor loadings greater than .50 were adopted to be the criteria (Hair, Anderson, Tatham, & Grablowsky, 1979). The factor loadings, eigen-values and percentages of variances explained were satisfactory, as listed in Table 3.

Table 4 lists the means, standard deviations and correlations of the variables. No correlation exceeded .50. Multi-collinearity was not a problem in this investigation according to the criteria suggested by Kennedy (1985).

Table 3
The reliability and validity of the final questionnaire of this study.

Variable	Sub- dimension	Item	Loading	Cronbach α	Eigen-value	Percentage of variances explained
Computer self-efficacy		Com1	.687	.91	5.69	56.89%
		Com2	.636			
		Com3	.751			
		Com4	.840			
		Com5	.746			
		Com6	.786			
		Com7	.783			
		Com8	.780			
		Com9	.765			
		Com10	.750			
Domain-specific IT skills	Generic skills	Gen1	.804	.81	2.90	57.90%
		Gen2	.803			
		Gen3	.836			
		Gen4	.795			
		Gen5	.522			
	Systems analysis skills	Sa1	.737	.91	4.236	70.60%
		Sa2	.871			
		Sa3	.875			
		Sa4	.911			
		Sa5	.923			
		Sa6	.698			
	Systems design skills	Sd1	.854	.63	1.46	72.99%
		Sd2	.854			
	Basic programming skills	Bp1	.708	.79	2.74	54.73%
		Bp2	.773			
		Bp3	.649			
		Bp4	.797			
		Bp5	.762			
	Advanced programming skills	Ap1	.755	.91	4.55	65.03%
		Ap2	.746			
		Ap3	.690			
		Ap4	.833			
		Ap5	.865			
		Ap6	.876			
		Ap7	.860			
	Operating systems skills	Os1	.863	.76	1.49	74.50%
		Os2	.863			
	Database skills	Db1	.777	.86	2.30	76.73%
		Db2	.907			
		Db3	.936			
Creative self-efficacy		Crea1	.769	.94	7.44	57.24%
		Crea2	.809			
		Crea3	.738			
		Crea4	.825			
		Crea5	.812			
		Crea6	.678			
		Crea7	.749			
		Crea8	.747			
		Crea9	.725			
		Crea10	.763			
		Crea11	.770			
		Crea12	.701			
		Crea13	.734			
The strength of ties		Ties1	.943	.94	2.69	89.57%
		Ties2	.963			
		Ties3	.933			

Table 4
Means, standard deviations, and correlations of all IS developers ($n = 94$).

Variable	Mean	Standard deviation	(1)	(2)	(3)	(4)	(5)
(1) Computer self-efficacy	3.96	0.54	1.00	.411**	−.100	.041	.348**
(2) Domain-specific IT skills	3.68	0.56	–	1.00	.059	.026	.465**
(3) The strength of ties	2.75	0.92	–	–	1.00	.415**	.124
(4) Degree centrality	98.19	4.73	–	–	–	1.00	−.137
(5) Creative self-efficacy	3.71	0.52	–	–	–	–	1.00

** $p < .01$.

4.2. Detecting relationships

This investigation examined the possible correlations between personal and contextual resources, and between the creative self-efficacy of participants including system analysts and programmers (two specific models). Regression analysis was performed to find these relationships. However, before running regression analysis of the two specific models, this study attempted to investigate the relationships in all participants (the wholistic model) as the baseline model for comparing with system analysts and programmers models.

Table 5 lists the results of the regression analysis for the wholistic model, including all participants. Statistical analysis results indicate that the wholistic model was significant for all IS developers, $F(3,89) = 10.103$, $p < .01$, and $R^2 = .312$. As revealed in Table 5, all variables, computer self-efficacy, domain-specific IT skills, the strength of ties, and degree centrality in the wholistic model significantly influenced creative self-efficacy. The variance inflation factor (VIF) of each independent variable was below 2, implying that multicollinearity was not a problem (Neter, Wasserman, & Kutner, 1990). The β coefficients show that the best predicators were domain-specific IT skills. Fig. 2 depicts the wholistic model of all IS developers.

However, system analysts and programmers are two different group of IS professionals. Although the differences in their variable values were not significant (as shown in Table 6), their relationships deserve separate exploration.

4.2.1. The model for system analysts

Table 7 lists the results of the regression analysis for the model of system analysts. Analytical results indicate that the system analyst model was also significant, $F(4,29) = 6.819$, $p < .01$), and $R^2 = .485$. From Table 7, in the system analyst sample, domain-specific IT skills and degree centrality were significant ($p < .05$) to influence creative self-efficacy, but the strength of ties was only marginally significant ($p < .10$). Additionally, the VIF of each

Table 5
Regression results of the wholistic model.

Predicator	Standardized β coefficient	p Value
Computer self-efficacy	0.232	.02*
Domain-specific IT skills	0.362	.00**
The strength of ties	0.230	.02*
Degree centrality	−0.251	.01**

* $p < .05$.
** $p < .01$.

Table 7
Regression results of the system analyst model.

Predicator	Standardized β coefficient	p Value
Computer self-efficacy	0.243	.13
Domain-specific IT skills	0.528	.00**
The strength of ties	0.282	.09
Degree centrality	−0.342	.03*

* $p < .05$.
** $p < .01$.

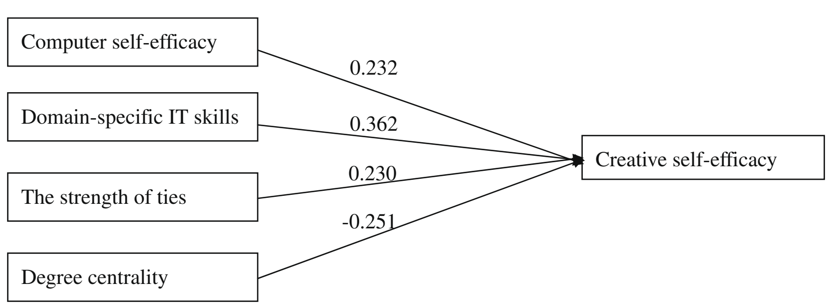

Fig. 2. A wholistic model of creative self-efficacy of all IS developers. Note: Solid line is significant ($p < .05$).

Table 6
Means of all IS developers, SA, and programmers.

Variable	All mean ($n = 94$)	SA mean ($n = 34$)	Programmer mean ($n = 60$)	Difference of SA and programmers
(1) Computer self-efficacy	3.96**	3.92**	3.99**	−0.07
(2) Domain-specific IT skills	3.68**	3.70**	3.67**	0.03
(3) The strength of ties	2.75**	2.93**	2.65	0.28
(4) Degree centrality	98.19	98.71	97.9	0.81
(5) Creative self-efficacy	3.71**	3.79**	3.66**	0.13

In the columns of mean (all mean and SA mean), they are compared with median of range (3 in rows (1), (2), and (5), 2.5 in row (3)).
** $p < .01$.

Table 8
Hypothesis summaries of the system analyst model.

Hypothesis	Support	Relational direction
Hypothesis 1a: A correlation exists between computer self-efficacy and creative self-efficacy in system analysts	Not supported	
Hypothesis 2a: A correlation exists between domain-specific IT skills of systems analysts and their creative self-efficacy	Supported	Positive
Hypothesis 3a: A correlation exists between the strength of ties among system analysts and their creative self-efficacy	Not supported	
Hypothesis 4a: A correlation exists between degree centrality and creative self-efficacy in system analysts	Supported	Negative

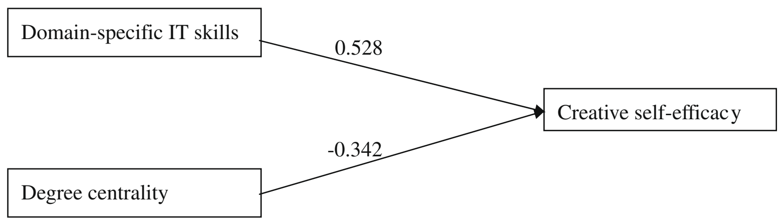

Fig. 3. A model of creative self-efficacy of system analysts. Note: Solid line is significant ($p < .05$).

Table 9
Regression results of the programmer model.

Predicator	Standardized β coefficient	p Value
Computer self-efficacy	0.254	.06
Domain-specific IT skills	0.243	.07
The strength of ties	0.172	.19
Degree centrality	−0.262	.05*

* $p < .05$.

independent variable was below 2, indicating no multicollinearity. The β coefficients show that the best predicators were domain-specific IT skills. Table 8 summarizes the hypotheses. Interestingly, degree centrality was negatively related to creative self-efficacy (Hypothesis 4a). Fig. 3 displays the model of system analysts.

4.2.2. The model for programmers

Table 9 lists the results of the regression analysis for the programmer model. The analytical results indicate that the programmer model was significant, $F(4,55) = 4.20$, $p < .01$, and $R^2 = .234$. Table 9 shows that degree centrality significantly influenced ($p < .05$) creative self-efficacy in the programmer sample, but computer self-efficacy and domain-specific IT skills were only marginally significant ($p < .10$). Furthermore, the VIF of each independent variable was below 2, indicating no multicollinearity problem. Table 10 summarizes these hypotheses. Again notably, degree centrality was negatively related to creative self-efficacy (Hypothesis 4b). Fig. 4 shows the programmer model.

5. Discussions

The findings of this study regarding the *domain-specific IT skills* of system analysts are consistent with the results from the literature, and indicated a positive correlation between knowledge and self-efficacy (Bandura, 1997; Gist & Mitchell, 1992; Tierney & Farmer, 2002). Domain-specific IT skills of system analysts include system analysis and design skills, which can help analysts to filter, judge and choose novel and feasible ideas to shape their creative judgment. In the model of system analysts, the main variable for explaining creative self-efficacy is domain-specific IT skills. However, the relationship between domain-specific IT skills and creative self-efficacy was not significant in the programmer model ($p = .07$). The domain-specific IT skills of programmers include knowledge of programming languages, databases and operating systems, which are versatile, changeable, and require daily updating. Programmers must keep on learning new knowledge. Gist and Mitchell also noted that experience-based familiarity with a task is important to individual assessments of their abilities. Programmers with high domain-specific IT skills may still lack the confidence to produce creative outcomes in the future. In this exploratory study, domain-specific IT skills of programmers exerted only a slight positive influence on creative self-efficacy.

Computer self-efficacy was not significant in the models of system analysts ($p = .13$) or programmers ($p = .06$). As we knew, all IS developers used computers to perform their tasks. The Table 6 indicated that IS developers (both system analysts and programmers) already had high computer self-efficacy. The results of this empirical study indicated that such a basic capability only satisfied the lower threshold for producing creative outcomes. Computer

Table 10
Hypothesis summaries of the programmer model.

Hypothesis	Support	Relational direction
Hypothesis 1b: A correlation exists between computer self-efficacy and creative self-efficacy in programmers	Not supported	
Hypothesis 2b: A correlation exists between domain-specific IT skills of programmers and their creative self-efficacy	Not supported	
Hypothesis 3b: A correlation exists between the strength of ties among programmers and their creative self-efficacy	Not supported	
Hypothesis 4b: A correlation exists between degree centrality and creative self-efficacy in programmers	Supported	Negative

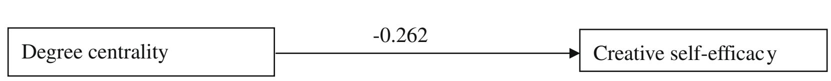

Fig. 4. A model of creative self-efficacy of programmers. Note: Solid line is significant ($p < .05$).

self-efficacy is related to domain-specific IT skills. However, there was still an interesting phenomenon that the correlation between computer self-efficacy and creativity self-efficacy was stronger for programmers (0.449) than system analysts (0.374). In the programmer model, the impact of computer self-efficacy on creative self-efficacy exceeded that for domain-specific IT skills. This phenomenon is explained as follows. System analysts do not require deep technological mastery to accomplish their tasks. In contrast, programmers must apply IT skills to develop, debug or maintain software products. Programmers, who can rapidly understand new technologies, can apply these skills flexibly to program coding. High computer self-efficacy in programmers thus can directly influence their performance capability and enhance their creative self-efficacy. However, this exploratory study found that computer self-efficacy exerted only a minor influence on creativity self-efficacy.

Strength of ties was not significant in both the model of system analysts ($p = .09$) and that of programmers ($p = .19$). Both strong ties and weak ties could help system analysts and programmers. However, differences still exist between system analysts and programmers. The impact of strength of ties on creative self-efficacy is greater for system analysts than programmers. Table 6 also shows a lack of significant differences between the mean and median (2.5) tie strengths of programmers. Our explanations are as follows. Human interactions appeared to improve the creative self-efficacy of system analysts more than that of programmers. Furthermore, system analysts must communicate frequently and closely with other developers to exchange ideas or tacit knowledge. System analysts with strong ties can easily obtain complete and tacit information, which then proves valuable in analyzing or designing system requirement and system process. This exploratory study thus found that tie strength slightly influenced creative self-efficacy. Conversely, discussion of content by programmers includes both explicit knowledge (such as programming syntax), and tacit knowledge (such as programming logic). Programmers facing logic problems frequently discover workable ideas, or exchange ideas via strong ties. However, programmers who encounter syntax problems may seek assistance through many weak ties. To programmers, strong ties can deliver complex and tacit knowledge while weak ties can carry explicit information. Both strong and weak ties provide valuable information in shaping programmer creative beliefs.

Degree centrality and creative self-efficacy were negatively related in the models of both system analysts and programmers. This relationship existed because greater degree centrality implies more direct ties. As mentioned, social comparison is a key factor in self-appraisal of capabilities (Suls & Miller, 1977). However, these comparisons are meaningful only when they involve comparable subjects in similar situations, such as work associates. Higher assumed similarity could lead to more persuasive mutual experiences (Bandura, 1997). Furthermore, many ties produce irrelevant or superficial information, or may even cause information overload, thus reducing time spent developing creative ideas (Perry-Smith & Shalley, 2003). Some IS developers surveyed in this study had numerous direct ties because they played too many roles. For instance, an IS developer, named β, is both a programmer and a team leader, and interacts with others in three ways. First, β is responsible for filtering information and assigning tasks to other programmers. Second, β must communicate with system analysts to clarify the problem specifications. Third, β interacts with other programmers to discuss programming logic and syntax. Not all direct ties help enhance creative self-efficacy among programmers. Only the third type of interaction increases programmer creative self-efficacy. Therefore, a negative relationship was found between degree centrality and creative self-efficacy.

6. Conclusions

This study makes several contributions to the literature. First, the construct of creative self-efficacy is extended to define the creative self-efficacy of IS developers. Second, social network theory is employed to explain contextual resources, and combined with personal resources to develop a model of creative self-efficacy for IS developers. Third, different models for system analysts and programmers are explained by addressing the task characteristics and types of information exchanged among developers.

6.1. Implications

The analytical results presented here have significant implications for IS practitioners. IS developer creativity is fundamental to software firm innovation. Tierney and Farmer (2002) identified creative self-efficacy as a key influence on creativity. This study identified the influences on creative self-efficacy of IS developers. The findings indicate that in the system analyst model, domain-specific skills were the main influence, followed by degree centrality. However, degree centrality was the only influence in the programmer model. The analytical results can help explain some of the differences between system analysts and programmers in software organizations.

These results indicate a strong positive correlation between domain-specific IT skills and creative self-efficacy in the system analyst model. The study findings thus support the conclusion of Gist and Mitchell (1992) that task knowledge is a personal factor employed in assessing self-efficacy. Knowledge is also a key element in developing creativity (Amabile, 1988). System analysts require IT-specific domain knowledge to assess their creative self-efficacy. IS managers thus should provide appropriate training to help system analysts shape their creative beliefs.

The findings of this study suggest that computer self-efficacy is the minimal requirement for IS developers, but does not determine creative self-efficacy. However, computer self-efficacy still positively (though slightly) influences creativity self-efficacy for programmers, but not system analysts. This phenomenon explains why not all IS developers require high computer self-efficacy. The tasks of programmers require the use of computer technology more than do those of system analysts.

Ties can transfer domain knowledge, skills and experiences that modify creative beliefs through comparisons with the creative achievements of others. IS managers should establish a sharing and interactive climate within their networks to facilitate sharing of creative models. Type of information transferred can vary according to whether ties are weak or strong. This exploratory study identified a minor influence of tie strength on creative self-efficacy in the model of system analysts. However, system analysts may require stronger ties than programmers to obtain information regarding creative events.

Furthermore, based on the finding of negative impact of degree centrality on creative self-efficacy, this study concludes that inappropriate job assignment may lead to unnecessary direct ties, and prevent employees from developing creativity self-efficacy. IS managers thus should be aware of those with whom IS developers must interact to accomplish their jobs and avoid unnecessary interactions.

6.2. Limitations and future research direction

This study suffers certain limitations. First, the study subjects belonged to a single software company in Taiwan. Although the characteristics of the software company are typical of Taiwanese software companies, over-generalization still needs to be avoided.

This study examines the correlations between several factors and the concept of creative self-efficacy. The hypotheses should be tested again by performing more general sample surveys, rather than using a single company sample. Second, the research subjects examined in this study are IS developers of software packages. Thus, the results of this investigation might not apply for IS developers of custom-made information systems. Third, this study considered the scope of social network only in terms of a software product. Only interactions among IS developers were observed. However, IS developers may interact with others beyond this product network. Future studies can explore other interactions involving all individuals with whom IS developers can interact. Fourth, observing interactions among IS developers to understand their social networks is time consuming. The data in this investigation were gathered only for a certain time period. Future works can consider longitudinal studies.

References

Amabile, T. M. (1988). *A model of creativity and innovation in organizations*. CT: JAI Press.

Amabile, T. M. (1997). Motivating creativity in organizations: On doing what you love and loving what you do. *California Management Review, 40*(1), 39–58.

Bandura, A. (1977). *Social learning theory*. Englewood Cliffs, NJ: Prentice-Hall Press.

Bandura, A. (1986). *Social foundations of thought and action: A social-cognitive theory*. Englewood Cliffs, NJ: Prentice-Hall Press.

Bandura, A. (1988). Organizational applications of social cognitive theory. *Australian Journal of Management, 13*, 137–164.

Bandura, A. (1997). *Self-efficacy: The exercise control*. New York: W.H. Freeman.

Benbasat, I., Dexter, A. S., & Mantha, R. W. (1980). Impact of organizational maturity on information system skill needs. *MIS Quarterly*, 21–34.

Berg, S., Duncan, J., & Friedman, P. (1982). *Joint venture strategies and corporate innovation*. Cambridge: Oelgeschlager.

Burt, R. S. (1982). *Toward a structural theory of action: Network models of social structure perception and action*. New York: Academic Press.

Burt, R. S. (1992). *Structural holes*. Cambridge: Harvard University Press.

Burt, R. S. (1997). The contingent value of social capital. *Administrative Science Quarterly, 42*(2), 339–365.

Chen, G., Gully, S. M., & Eden, D. (2001). Validation of a new general self-efficacy scale. *Organizational Research Methods, 4*, 62–83.

Cheney, P. H., & Lyons, N. R. (1980). Information systems skill requirements: A survey. *MIS Quarterly, 4*(1), 35–43.

Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly, 19*(2), 189–211.

Coser, R. (1975). The complexity of roles as a seedbed of individual autonomy' in K. Robert. In R. K. Merton & L. Coser (Eds.), *The idea of social structure*. New York: Harcourt Brace.

Fagan, M. H., Neill, S., & Wooldridge, B. R. (2004). An empirical investigation into the relationship between computer self-efficacy, anxiety, experience, support and usage. *Journal of Computer Information Systems, 44*(2), 95–104.

Ford, C. M. (1996). A theory of individual creative action in multiple social domains. *Academy of Management Review, 21*(4), 1112–1142.

Freeman, L. C. (1979). Centrality in social networks: Conceptualizations clarification. *Social network, 1*, 215–239.

Gardner, H. (1993). *Creating minds*. New York: Basicbook.

Gist, M. E., & Mitchell, T. R. (1992). Self-efficacy: A theoretical analysis of its determinants and malleability. *Academy of Management Review, 17*(2), 183–211.

Gist, M. E., Schwoerer, C., & Rosen, B. (1989). Effects of alterative training methods on self-efficacy and performance in computer software training. *Journal of Applied Psychology, 74*(6), 884–891.

Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology, 78*, 1360–1380.

Green, G. I. (1989). Perceived importance of systems analysts job skills, roles, and non-salary incentives. *MIS Quarterly, 13*(2), 115–133.

Grudin, J. (1991). Interactive systems: Bridging the gap between developers and users. *IEEE Computer, 24*(5), 59–69.

Gulati, R., & Garguilo, M. (1999). Where do interorganizational networks come from? *American Journal of Sociology, 104*(5), 1439–1493.

Hair, J. F., Jr., Anderson, R. E., Tatham, R. L., & Grablowsky, B. J. (1979). *Multivariate data analysis*. Tulsa, OK: PPC Books.

Hanninen, S., & Kauranen, I. (2007). Product innovation as micro-strategy. *International Journal of Innovation and Learning, 4*(4), 425–443.

Hansen, M. T. (1999). The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly, 44*(1), 82–111.

Ibarra, H. (1992). Homophily and differential returns: Sex differences in network structure and access in an advertising firm. *Administrative Science Quarterly, 37*(3), 422–447.

Keil, M., & Carmel, E. (1995). Customer-developer links in software development. *Communications of the ACM, 38*(5), 33–44.

Kennedy, P. (1985). *A guide to econometrics*. Cambridge, MA: MIT Press.

Kerlinger, F. N. (1978). *Foundation of behavioral research*. New York: Holt Reinhart Winston.

Krackhardt, D. (1992). The strength of strong ties: The importance of philos in organizations. In N. Nohria & R. C. Eccles (Eds.), *Networks and organizations: Structure, form, and action*. MA: Harvard University Press.

Lee, C. K. (2005). Analysis of skill requirements for systems analysts in fortune 500 organizations. *Journal of Computer Information Systems, 45*(4), 84–92.

Lee, D. M. S., Trauth, E. M., & Farwell, D. (1995). Critical skill and knowledge requirements of IS professionals: A joint academic/industry investigation. *MIS Quarterly, 19*(3), 313–340.

Lee, M.-C., & Chang, T. (2007). Linking knowledge management and innovation management in e-business. *International Journal of Innovation and Learning, 4*(2), 145–159.

Levin, D. Z., & Cross, R. (2004). The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer. *Management Science, 50*(11), 1477–1490.

March, J. G., & Simon, H. A. (1958). *Organizations*. New York: Wiley.

Morcillo, P., Rodriguez-Anton, J. M., & Rubio, L. (2007). Corporate culture and innovation: In search of the perfect relationship. *International Journal of Innovation and Learning, 4*(6), 547–570.

Nahapiet, J., & Ghoshal, S. (1998). Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review, 23*(2), 242–266.

Neter, J., Wasserman, W., & Kutner, H. (1990). *Applied linear regression models*. New York: McGraw-Hill Com.

Oldham, G. R., & Cummings, A. (1996). Employee creativity: Personal and contextual factors at work. *Academy of Management Journal, 39*(3), 607–635.

Perry-Smith, J. E., & Shalley, E. E. (2003). The social side of creativity: A static and dynamic social network perspective. *Academy of Management Review, 28*(1), 89–106.

Shane, S., & Cable, D. (2002). Network ties, reputation, and the financing of new ventures. *Management Science, 48*(3), 364–382.

Shin, Y. Y. (2006). The effect of computer self-efficacy on enterprise resource planning usage. *Behavior and Information Technology, 25*(5), 407–411.

Simonton, D. K. (1999). *Origins of genius*. New York: Oxford University Press.

Smith, K. G., Collins, C. J., & Clark, K. D. (2005). Existing knowledge, knowledge creation capability, and the rate of new product introduction in high-technology firms. *Academy of Management Journal, 48*(2), 346–357.

Sternberg, R. J., & Lubart, T. I. (1995). *Defying the corwd: Cultivating creativity in a culture of conformity*. New York: The Free Press.

Suls, J. M., & Miller, R. L. (1977). *Social comparison processes: Theoretical and empirical perspectives*. Washington: Hemisphere.

Tierney, P. T., & Farmer, S. M. (2002). Creative self-efficacy: Its potential antecedents and relationship to creative performance. *Academy of Management Journal, 45*(6), 1137–1148.

Todd, P. A., McKeen, J. D., & Gallupe, R. B. (1995). The evolution of IS job skills: A content analysis of IS job advertisements from 1970 to 1990. *MIS Quarterly, 19*(1), 1–27.

Uzzi, B. (1996). The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *American Sociological Review, 61*(4), 674–698.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. New York: Cambridge University Press.

Watson, H. J., Young, D., Miranda, S., Robichaux, B., & Seerley, R. (1990). Requisite skills for new MIS hires. *Data Base*, 20–21.

Woodman, R. W., Sawyer, J. E., & Griffin, R. W. (1993). Toward a theory of organizational creativity. *Academy of Management Review, 18*(2), 293–321.

Zhou, J., & George, J. M. (2001). When job dissatisfaction leads to creativity: Encouraging the expression of voice. *Academy of Management Journal, 44*(4), 682–696.