Programmers' Receptive Attitude toward Code Sharing and their Use of Computer-Mediated Communication Systems

Chatpong Tangmanee
Faculty of Commerce and Accountancy, Chulalongkorn University

ABSTRACT - One way for programmers to improve their work is to share code with peers. Computer-mediated communication (CMC) systems, such as e-mail, or the World Wide Web (WWW), may facilitate the process of code sharing provided that programmers are receptive toward the process. Upon a mail survey with 730 professional programmers, the study discovered three items of what programmers attain through CMC systems: (1) task-related, (2) socio-emotional and (3) exploring accomplishments. In addition, only among the programmers whose work involves actual code reviews was their receptivity toward code sharing correlated with their use of CMC systems for task-related purposes. The correlation with other accomplishments was not statistically significant. Discussions on the results' conceptual and practical contributions are addressed in the final section.

Keywords - computer-mediated communication, programmers, receptive attitude, code sharing

บทคัดย่อ - วิธีหนึ่งที่โปรแกรมเมอร์สามารถปรับปรุงคุณภาพของซอฟแวร์ที่พัฒนาขึ้น คือการแชร์หรือแลกเปลี่ยนความคิดเห็นเกี่ยวกับโปรแกรมที่พัฒนาขึ้นกับเพื่อนหรือผู้ร่วมงาน การสื่อสารหรือแชร์ความคิดเห็นที่ว่า อาจกระทำผ่านสื่อคอมพิวเตอร์ เช่นการใช้ไปรษณีย์อิเล็กทรอนิกส์หรืออินเตอร์เนต แต่กระนั้นการให้และรับฟังความเห็นผ่านสื่อคอมพิวเตอร์ในลักษณะนี้จะสำเร็จได้จริง ก็ขึ้นอยู่กับความใส่ใจของโปรแกรมเมอร์ต่อการแลกเปลี่ยนข้อเสนอแนะใดๆ กับเพื่อนร่วมงาน จากการเก็บข้อมูลด้วยแบบสอบถามทางไปรษณีย์ จากผู้ประกอบอาชีพโปรแกรมเมอร์จำนวน 730 คน พบว่า การสื่อสารผ่านสื่อคอมพิวเตอร์ของโปรแกรมเมอร์ มีขึ้นเพื่อสนองวัตถุ-ประสงค์ (1) ทางการงาน (2) ส่วนบุคคลและสังคม และ (3) เพื่อเรียนรู้ข้อมูลใหม่ ทั้งนี้เมื่อพิจารณาเฉพาะในกลุ่มของโปรแกรมเมอร์ที่เคยมีประสบการณ์จริงในการรีวิวโค้ด จะพบว่าความใส่ใจของโปรแกรมเมอร์ต่อการแลกเปลี่ยนความคิดเห็นกับเพื่อนร่วมงานมีความสัมพันธ์ทางบวกอย่างมีนัยสำคัญทางสถิติกับการสื่อสารผ่านสื่อคอมพิวเตอร์เพื่อการงานเท่านั้น โดยที่การสื่อสารเพื่อวัตถุประสงค์อื่นไม่มีความสัมพันธ์ทางสถิติกับความใส่ใจต่อเรื่องดังกล่าว การใช้ผลวิจัยทั้งทางทฤษฎี และทางปฏิบัติได้นำเสนอในบทสรุปของงานนี้

คำสำคัญ - การสื่อสารผ่านสื่อคอมพิวเตอร์, โปรแกรมเมอร์, ทัศนคติต่อการพัฒนาระบบสารสนเทศ, การแชร์โปรแกรม

1. Introduction

The goal of this research is to bridge a gap between programmers' communication behavior and their attitude toward a programming practice. Given the wide applications of communication technology to software development processes, it is still largely unknown whether programmers' use of such technology is related to their programming perception. The current study is an attempt to explore this void. In particular, it seeks to understand a connection between the purposes for which programmers are engaged in computer-mediated communication and their receptivity toward code sharing.

2. Computer-mediated communication systems and receptivity toward code sharing

Computer-mediated communication (CMC) systems are defined in this study as programmers' perceptions of a collection of tools that primarily facilitate human communication via computers and communication networks. In addition, neither voice mail nor facsimiles are included because a large number of users do not perceive the communication via these two channels as computer-mediated [10]. Thus, the instances of CMC systems investigated in this research include computer conferencing systems (e.g., e-mail or visual conferences), Internet-based communication (e.g., Internet relay chat (IRC), Internet phone, ICQ or the WWW), group support systems, groupware (e.g., Lotus Notes) and the systems that simply support communication via e-mail, newsgroups and/or listserves.

Code sharing is one of the significant programming practices [13]. Software experts have suggested that programmers share and review their code with peers [6,14]. Reaction from peers could improve the code quality because an overlooked flaw may be detected during the review [8]. The benefits of code sharing is still unrecognized, however, unless programmers are attentive and willing to practice it. Still, to practice code sharing appears to rely on how the programmers interact with peers. This is the role in which

CMC systems could help promote their receptivity and thereby provoke them to engage in code sharing. The current research thus defines the receptivity toward code sharing is the extent to which a programmer is willing to share programming code with peers.

3. Scope of the study

As said earlier, the current study attempts to shed light on programmers' computer-mediated communication behavior and their willingness to share code. To achieve the goal, it seeks to answer the following questions:

What do programmers accomplish through the use of CMC systems?

To what extent do the accomplishments relate to receptivity of programmers toward code sharing?

3.1 Accomplishments through CMC systems

The study's first focus is to examine what programmers accomplish through the use of CMC systems. Researchers have acknowledged the use of CMC systems so as to achieve progress or completion in work-related tasks. Sawyer and Guinan [11] examined how one CMC system would affect intra-group conflict-solving process among teams of software developers. "By focusing [on] the work product, and not [on] each other, the product becomes less attached to any one person: it is shared by the team" [11].

In addition to task-related benefits, the use of CMC systems can accommodate the social and emotional needs of programmers. This type of use has been ascertained in various investigations, although only a few have been conducted in the programmer context. For example, workers used e-mail not only to handle work assignments, but also to stay in touch with friends or family, or to meet people with the same interests [10]. With a secured distributed technology, programmers can complete any financial transaction (i.e., purchase a plane ticket or buy stock) over the Internet. They can also be entertained by a variety of information, or download it for their own pleasure [9].

Besides the two major purposes (i.e., task-related and socio-emotional), evidence from the literature suggest that programmers may use CMC systems for other accomplishments. For instance, Rice and Steinfield [10] identified the surveillance purposes of electronic communication. Other writers comment that organizational members may use communication to explore innovative ideas from the environment and, perhaps, share the knowledge with colleagues [5].

3.2 Computer-mediated communication and code sharing practices

Software researchers have suggested that programmers share and discuss their programming code with peers [7, 14]. An overlooked flaw is often uncovered during the discussion [8]. Consequently, the programming code that passes a thorough review will likely be bug-free [7, 13]. The idea of a programmer creating such bug-free software by having a review session with peers is widely accepted in many leading computer companies [3].

In a code review session, a programmer presents code to colleagues and responds to their feedback. At the end of the review, the programmer would learn about overlooked programming bugs and possible solutions.

Kraut and Streeter [15] investigate the extent to which programmers in one organization coordinated their work via various means of communication. E-mail and electronic bulletin boards were reported as critical channels through which the programmers could share their work with colleagues, elicit software requirements from clients, or negotiate with hardware vendors [15]. Another longitudinal study of listserves (one instance of CMC systems) used among program designers exhibits a connection between the design task and different roles of the designers [16].

Empirical evidence seems to confirm that a practice of code sharing involves programmers' communication skills. The use of CMC systems could thus be an alternative. There is no published formal method for programmers to share code via CMC systems. They may, however, utilize several features of CMC systems for the purposes of code-sharing. Examples of the features include (1) posting programming work on a corporate network and making it accessible to only involved members, (2) encouraging members to share feedback through an electronic bulletin board or sometimes to give comments in a company chat room, or (3) attaching a piece of code to e-mail sent to involved parties for subsequent examination. Sawyer and Guinan [11] remark that projecting the code on the wall appears to direct the attention of the members to the code on the screen, not to the programmer who wrote it. This therefore may raise programmer's willingness to share the code.

Despite the promising benefits of sharing code and the potential utility of CMC systems in assisting programmers to do so, the benefit is still unrealized unless the programmers are willing to do it. Johnson [8] reports the results of an informal survey in which 80% of 90 participants admitted that they practiced code reviews irregularly or not at all, although most of the subjects agreed that there are benefits from reviewing code with peers. Consequently, the ways in which the programmers use CMC systems (i.e., what they accomplish through CMC systems) may be associated with the extent to which they are receptive to code sharing.

4. Methodology

4.1 Questionnaire development

The first draft of a questionnaire was developed based upon two strategies. First, the literature on CMC and software engineering was reviewed. It helped the researcher to locate most of the questionnaire scales associated with various accomplishments one may have via the use of CMC systems. Only a few were constructed exclusively for this research because the scales were not found during the literature review. Second, the researcher conducted several interview sessions with actual programmers. This subsequent interview was to ensure that all items in the first draft were clear and understandable to programmers.

A group of scholars pretested the first draft. For the feedback from pretest participants, the questionnaire scales were modified. After the pretest, fifty other actual programmers participated in the questionnaire pilot test, of which the result is to assess the scale reliability and validity. The pilot test's findings ascertained an acceptable level of the questionnaire quality. Due to space constraints, a copy of the questionnaire is excluded from this manuscript but available upon request to the researcher.

4.2 Questionnaire administration

730 professional programmers who are members of the Association of Computer Machinery (ACM) received mail questionnaires. According to Babbie [1], this number of sample size is acceptable to provide a statistically significant finding. For a major drawback of a low response rate in using a mail survey, the researcher made every effort to follow recommendations from survey experts [1, 4] in order to draw a high volume of responses. After a three-month data collection, 438 programmers returned usable questionnaires, amounting to a 60% response rate. An examination of the non-respondents (i.e., the remaining 40%) using the trend projection method [12] detected no bias between the respondents and the non-respondents.

5. Results

5.1 Respondent demographics

Table 1 presents demographic variables of participating programmers. The highlights of these variables are as follows:

Table 1. Shows Characteristics of Participating Programmers

Major Characteristics	Respondents
	N (%)
Age (N=434)	
20-29 yrs.	18 (4)
30-39	134 (31)
40-49	166 (38)
50+	116 (27)
Gender (N=431)	
Male	381 (88)
Female	50 (12)
Highest Education (N=434)	
Some college	13 (3)
College degrees	105 (24)
Masters or some graduate work	273 (63)
Doctoral or higher	43 (10)
Major (N=429)	
Computer science	239 (56)
Mathematics	55 (13)
Engineering	49 (11)
Management or Business Administration	27 (6)
Information science	16 (4)
Physics	13 (3)
Others (e.g., Education, etc.)	30 (7)
Whether a code review is practiced at work (N=430)	
Yes	119 (28)
No	311 (72)
Work responsibility (N=432)	
Developing in-house systems	181 (42)
Developing packaged software	149 (35)
Installing packaged software in-house	17 (4)
Combination of the above three	23 (5)
Others (e.g., educational software)	62 (14)

- Considering that the sample was selected from regular members of the ACM who described themselves as programmers, it is not surprising that more than half of the respondents (63%) hold master degrees and about the same percentage (65%) are forty years old or higher. It seems that young programmers or those fresh from college are not ACM members as they may not yet realize the benefits of the memberships.
- The majority (88%) of participants are men. About half (56%) of the respondents hold their highest degree in computer science while other individuals are from adjacent fields.
- This research collected data from actual programmers, instead of from computer-related students. Furthermore, responses to the questionnaire's "work responsibility" item seem to confirm that the participants are in charge of various types of programming projects, ranging from developing in-house software systems (42%) and building packaged software products (35%), to installing packaged software (4%). These findings

thus ensure that the survey participants encompass professional programmers holding various actual programming responsibilities, not student programmers working on class assignments.

- Since the focus of this research is on a programmer's willingness to code sharing, whether a survey participant has actually been involved in code reviews may affect their perception. The questionnaire had therefore included one item asking respondents about their involvement. The result shows that about a quarter of the participating programmers (28%) reported that code reviews are practiced at their work.

5.2 What programmers accomplish through CMC systems

Thirty-five items reflecting various activities in which one may be engaged through CMC systems were included in the questionnaire [15]. To uncover the key purposes underlying what programmers accomplish through the use of CMC systems, the thirty-five items were factor-analyzed. Prior to the analysis, however, the items with marginal variance were excluded as they would not serve to differentiate among emerging factors. An objective criterion of a standard deviation of less than one is used to determine which items should be dropped from the analysis. As a result, four of the 35 items were excluded, leaving 31 items for the subsequent analysis.

Based on factor analysis with principle axis extraction and oblique rotation, three meaningful factors that underscore the major accomplishments programmers gain through CMC systems emerged. Table 3 displays the three purpose factors and the items that reflect on each purpose. Also included are weights of the items on the three factors. The three factors together explained about 42% of the variance among the purpose items. According to Table 3, Factor I accounted for 29.1% of the variance. Highest weights of the nine items on the first factor seem to reflect the "task-related" accomplishment programmers attain using CMC systems. Factor II explained 8.3% of the variance. Four items loaded highest on this factor, indicating that programmers use CMC systems for "socio-emotional" benefits. The final factor, Factor III, accounted for 4.5% of the variance. Highest weights of the other four purpose items tend to suggest the "exploring" purpose for using CMC systems. Assessments of the factor structure suggest that the discovery of these three accomplishments programmers gain through CMC systems is conceptually parsimonious and methodologically sound.

5.3 Receptive attitude toward code sharing and accomplishments through CMC systems

The study measured the survey participants' attitude toward code sharing using five-item Likert scales (see the scales in Appendix A). A mean of 4.19 with a standard deviation of 1.1 may indicate that the participating programmers appear to be receptive to code sharing. The receptive attitude becomes statistically different (see Table 2) among those who are involved in code reviews and those who are not. That is, the programmers who have actual experience with code reviews seem more willing to share code than those who do not. It may therefore suggest that any subsequent examination on this receptivity must take into account the significant difference.

Table 2 Shows Comparison of Receptivity toward Code Sharing between Programmers Whose Work Involves Code Reviews and Those Whose Work Does Not.

	Experience in Code Reviews	
Statistics	Do not have the experience	Have the experience
N	308	118
Mean[1]	4.098	4.456
S.D.	1.15	1.05

Note[1]: t = -2.94, df = 424, p < .003

5.4 Correlation between programmers' accomplishments through CMC systems and receptivity toward code sharing

In the previous section was a measurement description of programmers' receptivity toward code sharing. To explore the links between the receptivity and their use of CMC systems, three indexes characterizing the extent to which programmers use CMC systems for task-related, socio-emotional and exploring purposes were constructed. The indexes are derived from arithmetic means of the items that have highest weights for a given purpose. For instance, the index of a programmer using CMC systems for exploring purposes is computed from a mean of the four items that have the highest weights on this purpose (i.e., Factor III).

Table 3 Shows Factor Analysis Results: Purposes for Programmers to Use CMC Systems

Purpose Items	Factors			
	I	**II**	**III**	
Factor I: "Task-Related"				
Discuss work information with co-workers	**.75**	.06	-.01	
Coordinate work with distant colleagues	**.52**	-.01	-.04	
Schedule meetings	**.65**	-.05	-.04	
Give or receive feedback on work assignments	**.74**	.04	.03	
Send confirmation to colleagues/clients	**.64**	.02	.01	
Discuss work with clients	**.54**	.04	.08	
Resolve work conflicts or disagreements	**.60**	.03	.07	
Transfer files	**.54**	.05	.31	
Keep track of what's happening in a company	**.53**	.10	.06	
Factor II: "Socio-Emotional"				
Fill free time	-.02	**.77**	-.02	
Greet people on social occasions	.16	**.52**	.05	
Be entertained (e.g., electronic humor)	.02	**.68**	.02	
Take a break from work	-.02	**.78**	-.06	
Factor III: "Exploring"				
Check out new services/products	.03	-.04	**-.75**	
Stay up-to-date on computer upgrades	.02	.04	**-.73**	
Seek out alternatives to work problems	.05	.03	**-.62**	
Download information	.00	-.02	**-.83**	
Percent of Variance Explained	*29.1%*	*8.3%*	*4.5%*	*= 41.9%*

Using correlation analysis, the study found a slight positive yet significant correlation between task use and the receptivity of programmers toward sharing code (r = .101, p < .038). However, the correlation of this receptivity with socio-emotional use and exploring use are not statistically significant (see Table 4, a). Nonetheless, the significant difference in receptivity toward sharing code between programmers who have been involved in code reviews and those who have not (see Table 2) suggests further reexamination of the relationship.

An examination of this relationship among the programmers whose work involves code reviews reveals a slightly stronger and significant correlation between the receptivity and task use (r = .112, p < .023). The interaction effect of involvement in actual code reviews is further confirmed as the study found the insignificant correlation between the receptivity and task use among those whose work does not involve code reviews (see Table 4, b).

Table 4 Shows Correlation of Programmers' Accomplishments through CMC Systems and Receptivity toward Sharing Code

(a)

Accomplishments	Correlation Statistics
Task use	r = .101, p < .038, N = 426
Socio-emotional use	r = -.020, p < .674, N = 424
Exploring use	r = .089, p < .069, N = 423

(b)

Accomplishments	Correlation Statistics	
	Those whose work involves code reviews	Those whose work does not involve code reviews
Task use	r = .112, p < .023, N = 117	r = .071, p < .217, N = 305
Socio-emotional use	r = -.101, p < .281, N = 116	r = -.001, p < .988, N = 304
Exploring use	r = .135, p < .145, N = 117	r = .087, p < .131, N = 302

6. Conclusions

6.1 Programmers' accomplishments via CMC systems

Upon the study's results, programmers appear to achieve three purposes of using computer-mediated communication systems. They are task-related, socio-emotional and exploring purposes. While the first two accomplishments are generally recognized in the literature, the third is noted by few scholars [2]. Consistent with common literature, the findings stress the essential combination of "work" and "play" [5].

Also, researchers have acknowledged exploring functions of communication as the transfer of knowledge between an organization and its environment [2]. This piece of information may help an organization to cope with changes, especially when the organization's environment is undergoing dramatic shift. Given the dynamic and various changes in software environment, it is reasonable to argue that programmers conduct an exploration via CMC systems, perhaps, in search for innovative ideas (e.g., ready-to-use programming applets, or details of product upgrades) so as to survive the turbulent condition.

6.2 Programmers' receptivity toward code sharing and accomplishments through CMC systems

Throughout the entire sample, the degree to which programmers used CMC systems for task-related purposes exhibited a significant positive correlation with their receptivity toward code sharing. However, there existed a crucial difference in this receptivity between programmers whose work involves actual code reviews and those whose work does not. The sample was therefore divided into two groups according to whether a respondent had been involved in code reviews, and the correlation was reexamined. The interaction effect of involvement in code reviews was confirmed because the correlation among those having been involved in code reviews was slightly stronger and the correlation in the other group became non-significant.

Two conclusions may be drawn from the results. First, it appears that programmers' use of CMC systems for task-related purposes and their willingness to share code relate to each other in the same direction. Given that code sharing is one of the programming-related tasks, the finding could have been expected. The more programmers use CMC systems for task-related activities which may include code sharing, the more likely they experience the positive outcomes, which in turn, may increase their receptivity toward it. This speculation may also explain why socio-emotional and exploring uses were not significantly related to the receptivity.

The existence of the correlation among those who are involved in actual code reviews suggests the second conclusion. It seems that the involvement in code reviews mediates the relationship between task use and the receptivity. Only among those who have experience with actual code reviews (e.g., presenting their own code or taking part in a review session) was the correlation confirmed. Those who learn about code reviews and benefits of code sharing but have never been involved in an actual review may have different attitudes, regardless of how they use CMC systems.

7. Implications

The study's results lead to two implications. First, they extend theoretical concepts on CMC and on software engineering. Regarding communication functions, the current study has confirmed the need to incorporate all three major purposes (i.e., task-related, socio-emotional and exploring) into research on electronic communication. Regarding software engineering issues, the results disclose the interaction effect of programmer' involvement in actual code reviews on their willingness to code sharing. It is thus speculated that programmers' greater involvement in code review may result in more receptivity toward code sharing. Alternatively, without participating in real review sessions, programmers may unlikely conceive the benefits of code sharing and thereby their willingness to practice it remain unchanged.

Second, the study offers practical contributions. The results empirically confirmed three accomplishments programmers achieve through CMC systems. Software practitioners may thus expect to witness activities associated with these three types of use from their programming staff. Positive correlation between programmers' task use of CMC systems and their receptivity toward code sharing may imply that encouraging programmers to communicate via computer media for task-related purposes could increase their willingness to share code. The interaction effect of involvement in actual code reviews may remind practitioners that, unless programmers have real experience with the reviews, their task use of CMC systems may remain unrelated to their receptivity toward code sharing.

8. Limitations

Inferences from this research are limited by two major factors. First, the demographics of the participants appear to temper the study's generalizability of results to the programmer population in general. The programmers who participated in this study are dominantly male, between 30-49 years of age and with at least a college degree. This consequently limits the generalization of the study, despite the random sample of about 700 members, and the high percentage of survey returns.

The second limitation is more methodological. This is a cross-sectional study. Further, the phenomenon under study (i.e., CMC system usage) is dynamically changing due to rapid development of computer technology. Hence, the analyses and the conclusions present only a snapshot of how programmers use CMC systems. Additionally, prior to this study, very little was known about correlation between programmers' computer-mediated communication behaviors and their programming perception. The study's execution was therefore made based upon an exploratory approach. This is the main reason that the investigated relationships were neither hypothesized nor tested. Nevertheless, the study has ascertained the link between programmers' accomplishments via CMC systems and their receptivity toward code sharing.

References

[1] Babbie, E. R. *The practices of social research (6th edition)*. CA: Wadsworth publications, 1992.

[2] Choo, C. W. (1998). *The knowing organization: How organizations use information to construct meaning, create knowledge, and make decisions*. NY: Oxford University Press.

[3] Cusumano, M. A., & Selby, R. W. (1995). *Microsoft secrets*. NY: The Free Press.

[4] Dillman, D. (1978). *Mail and telephone surveys: The total design method*. NY: Wiley.

[5] Heath, R. L. (1994). *Management of corporate communication*. NJ: Lawrance Erlbaum Associate.

[6] Humphrey, W. S. (1995). *A discipline for software engineering*. MA: Addison-Wesley publishing.

[7] Humphrey, W. S. (1997). *Managing technical people*. MA: Addison-Wesley.

[8] Johnson, P. M. (1998). Reengineering inspection. *Communications of the ACM*, *41*(2), 49-52.

[9] Mehta, M. D., & Plaza, D. E. (1997). Pornography in cyberspace: An exploration of what's in USENET. In S. Kiesler (Ed.), *Culture of the Internet* (pp. 53-67). NJ: Lawrance Erlbaum Associates.

[10] Rice, R. E., & Steinfield, C. W. (1994). Experiences with new forms of organizational communication via electronic mail and voice messaging. In J. H. Andriessen & R. A. Roe (Eds.), *Telematics and work* (pp. 106-137). East Sussex, UK.: Lawrence Erlbuam Associates Ltd.

[11] Sawyer, S., & Guinan, P. J. (1994). Team-based software development using an electronic meeting system: The quality pay-off. In M. Lee, B. Barta & P. Jaliff (Eds.), *Software quality and productivity: Theory, practice, education and training* (pp. 78-83). New York, NY: Chapman Hall.

[12] Smith, J. (1997). Nonresponse bias in organizational surveys: Evidence from a survey of groups and organizations working for peace. *Nonprofit and voluntary sector quarterly*, *26*(3), 359-368.

[13] Weinberg, G. M. (1998). *The psychology of computer programming (the silver anniversary edition)*. NY: Dorset House Inc.

[14] Yourdon, E. (1999). *Death march*. NY: Prentice Hall, Inc.

[15] Kraut, R. E., & Streeter, L. (1995). Coordination in software development. *Communications of the ACM*, *38*(3), 69-81.

[16] Ahuja, M. K., & Carley, K. M. (1998). Network structure in virtual organizations. *Journal of computer-mediated communication*, *3*(4), Available at http://www.ascusc.org/jcmc/vol3/issue4/ ahuja.htm.

Appendix A

Sharing Code with Peers

To better understand your use of CMC systems, we would like to ask some questions about your perception toward sharing code with peers. Please indicate, by circling the appropriate number, to what extent you agree or disagree with these statements.

If you have not had experience with each of the following statements, please indicate your assessment based on your opinion, otherwise based on your own experience.

	Strongly disagree			**Neutral**		**Strongly agree**	
Reviewing code with peers is not so useful as it may sound….	-3	-2	-1	0	1	2	3
I feel nervous if I have to present my code to peers…………	-3	-2	-1	0	1	2	3
Showing peers my code makes me uncomfortable..…………	-3	-2	-1	0	1	2	3
I like to get comments on my code from peers.......……….	-3	-2	-1	0	1	2	3
It is fearful to share code with peers....................………….	-3	-2	-1	0	1	2	3

Chatpong Tangmanee holds a Ph.D. degree in Information Transfer from Syracuse University in New York. He is a full time faculty member in the field of Information Technology at the Faculty of Commerce and Accountancy's Department of Statistics in Chulalongkorn University. His research interests span across multiple areas including human-computer interaction, computer-mediated communication and impacts of interaction technology on individuals, organizations and society. He is also responsive to methodological issues in the topics of information technology and computer-mediated communication. A former subscriber to IEEE, he is currently a member of the Association of Computing Machinery (ACM) and the Institute for Operations Research and the Management Sciences (INFORMS).