Communication Learning Outcomes from Software Engineering Professionals: A Basis for Teaching Communication in the Engineering Curriculum

Susan Ruff and Michael Carter
ruff@mit.edu, mpc@ncsu.edu

Abstract - **It is widely acknowledged that employers tend to be dissatisfied with the communication skills of new engineering graduates, yet research into employers' expectations has not provided a sufficient foundation for incorporating workplace communication skills into the engineering curriculum. This study seeks to bridge that communication gap. Through focus groups and interviews with software engineers and managers, we have identified over 35 communication skills that comprise communication in the software engineering workplace, including a broad range of formal, interpersonal, professional, and team communication skills. These skills are presented in the form of outcomes that can be used by faculty not only to build a communication-rich program but also to enhance classroom instruction. Although the focus of this study is software engineering, most of the outcomes are applicable to other engineering programs.**

Index Terms – communication, computer science education, curriculum, education, engineering education, oral communication, outcomes, professional communication, software engineering, teamwork, writing

INTRODUCTION

Despite the fact that good communication skills are vitally important to being an effective engineer [1]-[7], the communication skills of recent engineering graduates typically fall short of employers' expectations [8]-[13]. To design a curriculum that bridges this gap, educators need to know which communication skills are expected in the workplace. Although the expectations of engineering professionals and managers have been the subject of many studies, these studies either focus only on technical skills or treat communication skills in terms that are too general for designing a curriculum [1], [6], [8], [11-13]. For example, the participants in Lidtke's [11] research—from academe, business, and industry—were "uniformly concerned about the lack of skills to work in a team environment and to communicate orally and in written form"; yet when these participants described a knowledge base for a computer information science curriculum, the knowledge base contained no knowledge about communication. More information about communication expectations is provided by Pfeiffer's [12] informal survey of exhibitors at OOPSLA in 1998: Pfeiffer lists 6 expectations for communication of software engineers—such as "The ability to sell oneself" and "Strong written and spoken English." Though a start, these expectations are still quite general. Designing a curriculum requires a more specific list of communication skills.

One study of professionals has provided more detail about communication skills. Using interviews conducted with engineers and a review of oral and written materials these engineers had produced on the job, Norback and Hardin [14] generated what they refer to as a master list of oral and written communication skills for a senior design course. Norback's "Criteria for Communication Excellence" [15], based on Norback and Hardin, are guidelines for the process of preparing and presenting engineering discourses. Though these communication criteria provide helpful detail about the skills needed to prepare presentations and written documents, the focus on oral and written materials may be too narrow for training engineers because it neglects less formal but equally important interpersonal communication.

What is needed, then, is a set of workplace communication skills that are both specific enough to serve as a foundation for curriculum building and broad enough to encompass a full range of professional needs. We also believe that in order to be useful, these skills should be presented in a way that is "curriculum-ready," that is, in the form of outcomes. Such industry-derived outcomes can be used by individual engineering programs to generate or revise their own curricular outcomes. Because outcomes delineate what faculty expect students to be able to do upon graduation, they may also be used to guide teaching and learning.

In this paper, we offer a list of communication skills that meet the criteria for specificity, breadth, and curriculum-readiness, with a particular emphasis on software engineering. We used focus groups and interviews of software engineers and managers to generate and analyze a fine-grained list of communication outcomes that can be used to guide software engineering curricula.

IDENTIFICATION OF OUTCOMES

To identify the communication skills that comprise communication in the software engineering workplace, we talked with 29 engineering professionals from 22 different companies from across the United States. About two thirds of the participants were managers, most of whom had been

978-1-4244-4714-5/09/$25.00 ©2009 IEEE

trained as software engineers; about one third of participants were practicing software engineers; one participant was a user interface designer and one was a project manager. One sixth of participants were women. No recent graduates were represented: no one had received a software-related degree within the past three years. Represented companies ranged in size from fewer than 10 employees to more than 50,000 employees. We solicited the participants' views through focus groups, informal interviews, and by e-mail.

Focus groups

Two focus groups were conducted, one face to face with 3 managers and the other by conference call with 14 managers. The first was done in conjunction with an NSF-sponsored Chautauqua in Teaching Communication Skills in the Software Engineering Curriculum. The second group consisted of members of the strategic advisory board of a computer science department at a large land grant university.

For both groups, the facilitator used a set of open-ended questions to stimulate a conversation about the communication skills needed for software engineers. The sessions ran 35-45 minutes. Because these were not structured focus groups and because time was limited, the facilitator did not cover all the questions and did not necessarily take them in order (see Appendix A for list of questions). In both groups the facilitator took notes on the participants' responses and turned assertions of desired attributes of software engineers into outcomes statements. These statements were given to participants within 24 hours of each focus group and participants were asked to make any necessary changes. No change was suggested.

Informal interviews

Six software engineering professionals were interviewed, 4 individually and 2 jointly. The interviewees were either previously known to the researchers or were referred by software engineers known to the researchers.

Open-ended questions were used to elicit interviewees' thoughts about the communication skills needed for software engineers (see Appendix A). Interviews typically lasted 60-90 minutes. At the end of some interviews, participants were shown preliminary research results and were encouraged to indicate the extent to which they agreed or disagreed with those results. This portion of the interview came at the end of the interview to avoid influencing the responses to the central interview questions. The interviewer took notes during the interviews and turned assertions of desirable attributes of software engineers into outcomes statements within 24 hours.

E-mail responses

Three software engineers who were unable to participate in a focus group offered their thoughts by e-mail, in person, and/or on the phone. One of the engineers solicited responses from three of his coworkers as well; these responses were communicated by e-mail and in person. The original e-mail query contained an invitation to a focus group, a link to the Web page of the Software Engineering Chautauqua to provide context, and a statement of the objective of the research.

OUTCOMES

The responses of the participants were used to generate a list of more than 80 communication outcomes; these outcomes were further combined and distilled. Outcomes that were suggested by at least two participants are listed in Table I. A sampling of outcomes that were suggested by only one person (and hence were omitted from Table I) is shown in Appendix B.

As Table I illustrates, software engineers should be able to design communication based on its context, explain clearly, discuss productively, receive communication, communicate professionally, and use common forms and tools of communication. Each of these broad communication categories contains several fine-grained communication outcomes. The outcomes illustrate that software engineers must communicate not only in oral and written forms, but also through less tangible interpersonal interactions; not only within a team, but also across organizational boundaries; not only giving communication, but also professionally eliciting and receiving it.

Scope of results

Although our participants were software engineers and managers and our questions specifically addressed software engineering, few of the identified communication skills are unique to software engineering. These findings suggest that our software engineering outcomes may be appropriate for other engineering curricula. This observation is supported by a comparison of our results with those of studies from other engineering disciplines.

Comparison with other studies

Though the goal of Norback and Hardin [14]-[15]—to help industrial engineering seniors to write and speak more effectively in their capstone course—is more narrowly focused than ours, their study provides partial support of our outcomes. In particular, the outcomes in our categories "Design Communication" and "Explain Clearly" overlap to a large extent with the criteria in Norback's categories "Communication Strategy" and "Preparing the Material," and our outcome "Give effective and engaging presentations" is the focus of the criteria in Norback's category "Delivery of Presentations" [15]. Despite this agreement between our outcomes and Norback and Hardin's work, most of our outcomes related to informal and interpersonal communication were not reported by Norback and Hardin. This difference may be explained by the fact that Norback and Hardin's interviews focused largely on artifacts (technical reports and presentation materials).

The emphasis on informal and interpersonal communication that is evident in Table I agrees with and extends the work of Vest, Long, Thomas, and Palmquist [13] in electrical engineering, Pfeiffer [12] in software

TABLE I COMMUNICATION OUTCOMES FROM SOFTWARE ENGINEERING PROFESSIONALS

Software engineers should be able to:

Design communication

Evaluate communication situations and design communication appropriately for different purposes and contexts.

Frame communication in terms of the knowledge & concerns of the audience. Communicate effectively to a variety of audiences, (e.g. managers, peers, across organizational boundaries, customers, & end users).

Recognize the different communication cultures and norms of different countries, organizations, areas within organizations, ethnic groups, and individuals, and adapt to those differences.

Prioritize communication tasks to use time wisely.

Discern when it is more appropriate/effective to keep silent rather than to speak and to ask questions rather than to assert an opinion.

Explain clearly

Present information in a way that goes beyond the specific details of a project to provide the big picture, a higher level of summary.

Explain code, methods, and design decisions by communicating the intent—what was meant to be achieved—and reasons—why key choices were made.

Achieve an appropriate balance between conciseness and explanation; go directly to the point.

Answer questions clearly by going beyond what the questioner has explicitly asked; anticipate what else the questioner might need to know.

Communicate effectively under stress.

Communicate convincingly.

Use consistent and appropriate terminology.

Discuss productively

Lead a productive group discussion.

Deal constructively with conflict: debate/discuss/negotiate/collaborate productively and respectfully.

Support the transition from debate to the formation of a decision; e.g. summarize issues, propose solutions, &/or back down, as needed.

Hear criticism as a constructive contribution to the outcome of a project (without getting defensive).

Give criticism constructively and respectfully.

Collaborate with others within an integrated project team or from different areas of the organization. Demonstrate an understanding of how software engineering decisions affect others by communicating across organizational boundaries to inform, solicit input, and identify win-win solutions.

Receive communication

Solicit help, advice, or information.

Listen actively; ask clarifying questions.

Read with comprehension and evaluate information to determine what is credible and relevant.

Adjust communication based on (non-verbal) reactions of the audience; solicit feedback about the effectiveness of the communication.

Learn & improve communication skills, especially interpersonal skills.

Communicate professionally

Give opinions with a balance of confidence & humility.

Avoid complaining, by proposing a solution, fixing the problem, or remaining silent.

Be nice to others, through words and tone.

Manage non-verbal communication to avoid sending inappropriate messages.

Make own accomplishments known without arrogance.

Communicate charismatically; be passionate/animated in order to influence people.

Mentor others and help them grow.

Communicate through transparency (make information openly available).

Develop the flexibility to communicate in different roles within an organization.

Inform managers and team members of potential problems before the problems become serious.

Participate in meetings.

Use common forms & tools

Demonstrate a mastery of the kinds of formal and informal communication most often used in the industry (e.g., email, bug reports, meetings, presentations to groups, one-on-one, teleconferences, IM, code comments, documentation, requirements, status reports).

Use digital tools that are beneficial for communication and teamwork (e.g., tools for document control, bitmap and vector illustrations, documentation, web pages, basic video/audio for presentations, intuitive GUI design, and project planning).

Give effective and engaging presentations.

Use email appropriately, demonstrating an understanding of what information should be included and what should not, of when to use "reply all," and of the necessity to read carefully before sending.

engineering, and Davis, Beyerlein, and Davis [16] in engineering in general. Vest et al. identify face-to-face communication and e-mail as accounting for the vast majority of all workplace communication, and Pfeiffer's six communication skills include informal skills such as "ask questions" and "sell oneself." In the engineering profile developed by Davis et al., based primarily on data from teachers of capstone design courses, the authors' "Communicator" engineering role overlaps somewhat with our "Design communication" and "Receive communication" categories, and their "Collaborator" role overlaps to a large extent with the outcomes in our "Discuss productively."

Despite the above similarities, Vest et al. and Pfeiffer mention a few informal communication skills that do not appear in Table I: for example, Vest et al. identify as purposes of communication "develop a sense of community" and "establish stable, broad avenues of communication" [13], and Pfeiffer identifies the skill "create clear metaphors that communicate a system's purpose" [12]. More importantly, many of the outcomes in our category "Communicate professionally" are mentioned in none of the other studies, despite being mentioned by multiple participants in our study. Although it might be tempting to conclude that only software engineers need to communicate professionally, this conclusion seems unlikely. We conclude, rather, that the skills that comprise professionalism are numerous, varied, and poorly defined, so these skills are usually omitted from discussions of communication skills. Thus, our list may provide a needed starting point for addressing professionalism within an engineering curriculum.

APPLICATION OF OUTCOMES

In Table I, the skills that comprise communication in the software engineering workplace are presented in the form of curricular outcomes. These outcomes may be focused to fit the needs of an individual department. For departments accredited by ABET, these outcomes could be used for the communication and teamwork categories or to flesh out current outcomes that may presently be too general to be useful. Departments that have not yet established curricular outcomes could generate their own outcomes by using Table I to spark discussion with alumni in industry and an advisory board.

We presented the list of communication skills in the form of outcomes to make them more useful as guides to teaching and learning. Simply distributing the list of outcomes to educators and students may be sufficient to enable them to direct their own teaching and learning. To assist educators, we briefly provide some suggestions for application.

Provide opportunities to communicate in an engineering context

Because the outcomes describe what graduates should be able to do in the workplace, they point explicitly to what students should be given opportunities to do in a curriculum. Providing practice with each skill in isolation would be quite difficult—a more natural approach is to provide students practice with engineering tasks that are cooperative and hence inherently require a range of the communication skills described in Table I. Several examples of such tasks are described by Felder, Woods, Stice, and Rugarcia [17]. Prince [18] provides quantitative evidence that such cooperative learning can measurably improve both interpersonal skills and academic achievement. Thus, by engaging in cooperative activities within an engineering context, students can improve interpersonal communication skills and they can better learn the engineering subject matter about which they are communicating (see also [19-20]). We therefore recommend that students be given opportunities for cooperative communication in an engineering context.

Perhaps the most important function of these outcomes is to provide a foundation for integrating engineering communication in the engineering curriculum. An integrated curriculum would ideally incorporate some aspects of an engineering apprenticeship. For example, in group engineering projects, students could be asked to negotiate timelines and dependencies, to present status reports, to copy the teacher on minutes of group meetings, etc. In other words, students would learn by participating as professionals as they prepare to become effective members of a community of engineers [21-24]. Integrating engineering communication in the curriculum is supported by Vest et al. who interviewed 6 newly hired electrical engineers and found that "communication [is] a core activity that is not separable from their other tasks" [13]. Vest et al. therefore recommend that engineering courses should reflect the engineering workplace by making communication not tangential but integral to the real work of student engineers (see also Patton [25]).

Provide communication instruction, feedback, & assessment

Some engineering faculty may not feel qualified to provide communication instruction, feedback, and assessment. We do not believe that engineering faculty should try to teach communication as English or communication teachers; rather, engineering faculty can be more helpful to students by responding as engineers. An experienced engineer is likely to be the best judge of the effectiveness of engineering communication: the primary criterion for this effectiveness is the extent to which the communication achieves its purpose for its audience, e.g., other engineers or managers. In our experience, giving students thoughtful feedback about the communication's effectiveness is often sufficient to help them improve. Students with persistent difficulty communicating effectively may be encouraged to seek help from communication specialists.

To focus feedback and evaluation on effective communication, appropriate items from the outcomes may be incorporated in a grading rubric. Teammates may also give each other structured feedback relating to the outcomes of productive discussion and professionalism, using team

feedback strategies similar to those described by Oakley, Felder, Brent, and Elhaji [26].

Conclusion

The value of the outcomes in our list is that they offer concrete descriptions of what engineers should be able to do; thus, they suggest concrete ways for engineering faculty to provide students with the broad range of formal, interpersonal, professional, and team communication skills needed in industry.

APPENDIX A

Open-ended questions like the following were used to guide discussion during focus groups and interviews.

- Imagine a software engineer who is the ideal communicator. What are the traits that make that person such a good communicator?
- What are some of the key problems you find in the communication of software engineers? In what way do they not meet your expectation?
- What are the most important forms of communication for software engineers? the most important purposes?
- What communication skills do you look for in hiring?
- Which communication skills are rewarded (through promotion, bonuses, choice of projects/teams, etc)?
- If you could have learned one communication skill before entering industry, what would you want it to be?
- What communication skills are needed of recent software engineering graduates?

Although the last question is about recent graduates, responses tended to be based on experiences with coworkers. Thus, responses indicated skills needed by software engineers in general, not by graduates in particular.

APPENDIX B

These are examples of outcomes omitted from Table I because they are from comments made by only one person.
Software engineers should be able to

- Develop rapport with others.
- Clearly articulate personal goals to supervisors.
- Professionally and respectfully push back if asked/told to do unreasonable tasks.
- Request help; communicate about things they don't know or understand, without getting upset or defensive.
- Offer help respectfully.
- Write readable code.
- Speak English fluently.
- Respond professionally to own mistakes.
- Communicate accurately, precisely, and honestly.

REFERENCES

[1] Bates, F.E. and Conner, D. A. "Industry survey for the University of Alabama at Birmingham (UAB): 2005 Electrical Engineering Curriculum Study," *Frontiers in Education Conference*, 1994, pp. 242-244.

[2] Evans, D. L., Beakley, G. C., Crouch, P. E., and Yamaguchi, G. T. "Attributes of engineering graduates and their impact on curriculum design." *Journal of Engineering Education*, Vol. 82, No. 4, 1993, pp. 203-211.

[3] Sageev, P. and Romanowski, C. J. "A message from recent engineering graduates in the workplace: Results of a survey on technical communication skills." *Journal of Engineering Education*, Vol. 90, No. 4, 2001, pp. 685-692.

[4] Spretnak, C. M. "A survey of the frequency and importance of technical communication in an engineering career." *Tech. Writing Teacher*, Vol. 9, 1982, pp. 133-136.

[5] D'Aquino, R. "Well prepared?" *Chemical Engineer*, No. 766, 2005, pp. 50-51.

[6] Henderson, K. "Educating electrical and electronic engineers." *Engineering Science and Education Journal*, Vol. 6, No. 3, 1997, pp. 95-98.

[7] Polack-Wahl, J. A. "It's time to stand up and communicate." *Frontiers in Education Conference*, Vol. 1, 2000, pp. FiG/16-F1G/21.

[8] Baldwin, C. J., Cahn, C. R., Forman, J. W., Lehmann, H., and Wischmeyer, C. R. "A model undergraduate engineering curriculum." *IEEE Trans. on Education*, Vol. E22, No. 2, 1979, pp. 63-68.

[9] Lebowitz, J. "Teaching the importance of communication in IT." *IT Professional*, Vol. 6, No. 1, 2004, pp. 38-42.

[10] Brammer, C. and Ervin, N. "Bridging the gap: A case study of engineering students, teachers, and practitioners." *Communication Jazz: Improvising the New International Communication Culture*, 1999, pp. 251-255.

[11] Lidtke, D. K. "Educating the next generation of information specialists, in collaboration with industry." *Frontiers in Education Conference*, 1996, pp. 126-129.

[12] Pfeiffer, P. "What employers want from students: A report from OOPSLA." *SIGCSE Bulletin*, Vol. 31, No. 2, 1999, pp. 69-70.

[13] Vest, D., Long. M., Thomas, L., and Palmquist, M. E. "Relating communication training to workplace requirements: The perspective of new engineers." *IEEE Trans. on Prof. Commun.*, Vol. 38, No. 1, 1995, pp. 11-17.

[14] Norback, J. S. and Hardin, J. R. "Integrating workforce communication into senior design tutorial." *IEEE Trans. on Prof. Commun.*, Vol. 48, No. 4, 2005, pp. 413-426.

[15] Norback, J. S. "Norback criteria for communication excellence." http://www2.isye.gatech.edu/workforcecom/commtools/tool6_commexcel.pdf Visited 6 February 2009.

[16] Davis, D. C., Beyerlein, S. W., and Davis, I. T. "Development and Use of an Engineer Profile." *2005 ASEE Annual Conference & Exposition*, Session 3155, 2005.

[17] Felder, R., Woods, D., Stice, J., and Rugarcia, A. "The future of engineering education: II. Teaching methods that work." *Chemical Engineering Education*, Vol. 34, No. 1, 2000, pp. 26-30.

[18] Prince, M. "Does active learning work? A review of the research." *Journal of Engineering Education*, Vol 93, No. 3, 2004, pp. 223-231.

[19] Timmerman, B, Lingard, R., and Barnes, G. M. "Active learning with upper division computer science students." *Frontiers in Education Conference*, 2001, Vol. 1, pp. T3B/19-T3B/23.

[20] Cordes. D. "Active learning in computer science: Impacting student behavior." *Frontiers in Education Conference*, 2002, vol. 1, pp. T2A/1-T2A/5.

[21] Lave, J. and Wenger, E. Situated learning: Legitimate peripheral participation. Cambridge: Cambridge University Press, 1991.

[22] Brown, J. S., Collins, A., and Duguid, P. "Situated cognition and the culture of learning." *Educational Researcher*, Vol. 18, No. 1, 1989, pp. 32-42.

[23] Carter, M., Ferzli, M., and Wiebe, E. N.. "Writing to learn by learning to write in the disciplines." *Journal of Business and Technical Communication*, Vol. 21,No. 3 2007,, pp. 278-302.

[24] Guzdial, M. and Tew, A. E. "Imagineering inauthentic legitimate peripheral participation: An instructional design approach for motivating computing education." *Proceedings of the 2nd International Computing Education Research Workshop*, 2006, pp, 51-58.

[25] Patton, M. D. "Beyond WI: Building an integrated communication curriculum in one department of civil engineering." *IEEE Trans. on Prof. Commun.*, Vol. 51, No. 3, 2008, pp. 313-327.

[26] Oakley, B., Felder, R. M., Brent, R., and Elhajj, I. "Turning student groups into effective teams." *Journal of Student Centered Learning*, Vol. 2, No. 1, 2004, pp. 9-34.

ACKNOWLEDGMENT

We thank the research participants for their valuable time and NSF award 0722231 for travel support to the Chautauqua that sparked this research.

AUTHOR INFORMATION

Susan Ruff, Writing Across the Curriculum, Massachusetts Institute of Technology, ruff@mit.edu.

Michael Carter, Associate Director, Campus Writing and Speaking Program, North Carolina State University, mpc@ncsu.edu