

Walking the Line? The Enactment of the Social/Technical Binary in Software Engineering

Vivian Anette Lagesen,

Department of Interdisciplinary Studies of Culture, Norwegian University of Science and Technology, 7491 Trondheim, Norway, email: vivian.lagesen@ntnu.no, phone: +4791172760

Knut H. Sørensen (corresponding author),

Department of Interdisciplinary Studies of Culture, Norwegian University of Science and Technology, 7491 Trondheim, Norway, email: knut.sorensen@ntnu.no, phone: +4791897365

Published in: *Engineering Studies*, 1(2), 129-149, 2009. DOI: <http://dx.doi.org/10.1080/19378620902911584>. Check with the journal for a final version of the text.

Abstract

This paper shows how professional communication practices with customers are accounted for in software engineering. It looks at how communication and related activities are enacted and placed in relation to the so-called social/technical binary while also critically engaging with analysing how this dualism is performed. Empirically, the paper investigates how software engineering and communication with customers are framed in two settings: at one Norwegian university and in three Norwegian software companies. At the university, an effort was made to reframe software engineering as a communication-oriented rather than technically-focused activity. However, faculty as well as students reproduced a technically-focused framing of software engineering that externalizes communication. The framing observed in the companies was different, with less outspoken distinction between 'technical' and 'social' aspects. Rather, communication with customers was described as based and dependent on technical knowledge. However, a closer reading shows how the social/technical binary is maintained by a consistent reference to the technical in professional terms while communication is described in lay terms. Implications of this are discussed in the conclusion.

Introduction

In this paper, we analyze how professional communication practices with customers (including users) are accounted for in software engineering.¹ We focus on the way communication and related activities are enacted and placed in relation to the

¹ There are several labels one could apply to describe the admittedly diverse group of professionals we analyze in this paper. We have chosen to use 'software engineering', which is employed in an inclusive way to designate professionals working with computers, systems and software.

social/technical binary, so commonly operating in engineers' accounts of their work.² However, this dualism will also be critically investigated, given the unclear meaning of 'social' as well as 'technical'. As a point of departure, we nevertheless start from observations that versions of this dualism operate in such disparate fields as technical communication, software engineering research and feminist studies of women in information and communication technology (ICT). What is at stake here?

Technical communication aims to identify and teach the skills needed for engineers to be good communicators. At the same time, we find in this literature arguments for the importance of communication specialists.³ Thus, we observe a controversy of whether technical communication should be an integrated part of engineering work or made into a task of communication experts. Moreover, critics have noted that the teaching of technical communication tends to be hyperpragmatic, meaning that it is distinguished by "a hegemonic ideology and set of practices that privileges utilitarian efficiency and effectiveness, including rhetorical effectiveness, at the expense of sustained reflection, critique or ethical action".⁴ This also points to the difference between a 'technical' and a more 'social' emphasis.

A similar ambiguity is found in the literature about skills and methods in software engineering research. One popular proposal, here voiced by Dahlbom and Mathiassen, is that software engineers should be hybrids, integrating technical and social skills: "(I)n addition to the traditional technical competence of an engineer, a computer professional today needs more skills of a social, organizational, communicational nature. A vision of a new engineer with a dual competence is presented, in which what is needed in addition to the core engineering competence may vary: both engineer and manager, both engineer and salesperson, both engineer and organizational expert, an ethically, socially responsible engineer, and so on".⁵ Many have long argued for the need for hybrid skills in the software/information systems industry,⁶ arguably stemming from a widening of the tasks expected to be performed by software engineers.⁷ Thus, we find a long-standing effort to create a more inclusive image of what software engineers actually should be doing, that their practice actually should involve a combination of 'technical' and 'non-technical elements'.⁸ This effort has also resulted in calls for educational reform to improve the 'non-technical' skills of software engineers, for example, with respect to communication and the ability to analyze social effects of technological change.⁹

However, in these contributions, the emerging understanding of the social/technical binary remains complex and opaque. This may be due to the

² Faulkner, "Dualisms, Hierarchies and Gender in Engineering", 2000.

³ Rosenstein et al., *Engineering Communications*, 1964; Roze and Maxwell, *Technical Communication in the Age of the Internet*, 2002.; Sales, *Professional Communication in Engineering*, 2006; Johnson-Eilola and Selber, *Central Works*, 2004.

⁴ Scott et al., "Introduction", 2006, p. 9.

⁵ Dahlbom and Mathiassen, "The Future of Our Profession", 1997, p. 85.

⁶ See, for example, Friedman and Greenbaum, "Wanted", 1984.

⁷ Friedman with Cornford, *Computer Systems*, 1989.

⁸ Goles et al., "Information Technology Workforce Skills", 2008; Lee, "Analysis of Skill Requirements for Systems Analysts in Fortune 500 Organizations", 2005; Litecky et al., "The Paradox of Soft Skills versus Technical Skills in IS Hiring", 2004; Turley and Bieman, "Competencies of Exceptional and Nonexceptional Software Engineers", 1995.

⁹ Brookshire et al., "An End-User Information System for the 21st century", 2007; Dahlbom and Mathiassen, "The Future of Our Profession", 1997.

unresolved tension between an education usually defined as technological or techno-scientific and the emphasis on skills so commonly considered part of the social sciences and the humanities by universities across the world. In contrast, other contributions emphasize the need for interdisciplinarity in design and implementation of software and systems,¹⁰ the need for ethnographic approaches,¹¹ or more complex approaches involving designers, other experts, and users.¹² Thus, there are different views of how to organize communication with customers. It is a shared belief among the authors cited above that design and implementation of software and computer systems are hybrid socio-technical activities. The disagreement among them concerns how such hybrid practices should be managed—by extending the core skills of software engineers as argued by Dahlbom and Mathiassen,¹³ or by specialization and interdisciplinary teamwork as suggested by Faola and Scandurra et al.¹⁴

Also, feminist studies of women in software engineering/computing have been concerned with the issue of the distribution of communication skills. This is based on the expectation, particularly among cyberfeminists, that women will acquire an increasingly prominent role in relation to information and communication technologies due to their alleged superior communication skills, compared to men.¹⁵ Moreover, some research suggests that women software engineers see themselves and are seen by management as having better communication skills than men co-workers.¹⁶ Ruth Woodfield found in the software company she studied a widespread understanding that managers preferred people with hybrid skills and that the managers also believed women were the best hybrids.¹⁷

However, the same study discovered that women employees did not benefit from this assumption. Maybe their assumed talent in communicating with customers and users was seen as a ‘natural’ ability and thus as less valuable compared to men’s acquired technical skills?¹⁸ At least, Woodfield found that technical skills were seen as ultimately more important because they were perceived as the basis for selling products and services.¹⁹ Her findings indicate that there may be ambiguous or conflicting perceptions of what communication actually means in software engineering practices.

In this paper, we analyze how communication with customers and users is typically placed with respect to two ways of managing the social/technical binary, as either hybrid competence in individuals or specialization. Moreover, inspired by the feminist literature, we see gender as an accounting resource that may be invoked to reflect on the character of communication and other people skills in software

¹⁰ E.g., Faiola, “The Design Enterprise”, 2007; Scandurra et al., “From user needs to system specifications”, 2008.

¹¹ Hartman et al., “Implementing information systems”, 2008.

¹² Suchman, “Practice-Based Design of Information Systems”, 2002.

¹³ E.g., Dahlbom and Mathiassen, “The Future”, 1997.

¹⁴ E.g., Faola, “The Design Enterprise”, 2007; I. Scandurra et al., “From user needs to system specifications”, 2007. See also Suchman, “Practice-Based Design of Information Systems”, 2002.

¹⁵ See, for example, Spender, *Nattering on the Net*, 1995; Hawthorne and Klein, *Cyberfeminism*, 1999; Kirkup et al., *The Gendered Cyborg*, 2000.

¹⁶ Ruiz Ben, “Defining Expertise in Software Development While Doing Gender”, 2007; Kelan, “Emotions in a rational profession”, 2007; Moore et al., “Gendered Futures?”, 2008.

¹⁷ Woodfield, *Women, Work and Computing*, 2000.

¹⁸ Moore et al. “Gendered Futures?”, 2008, p. 537.

¹⁹ Woodfield, *Women, Work and Computing*, 2000, pp. 186-197.

engineering. This is an issue we will pursue in general terms, although we do not specifically analyze differences between men and women.

The Social/Technical Binary in Engineering Studies: How to Situate Communication

When Wendy Faulkner argues that engineers commonly invoke social/technical binaries when they account for their work and competence, she also notes how such binaries tend to be turned into hierarchies privileging what is seen as technical practices over what is characterized as social or people-related activities.²⁰ In the field of science and technology studies, it has for a long time been a truism that social/technical dualisms should be deconstructed and transformed into seamless webs of sociotechnical constructions.²¹ From this perspective, employing descriptions like ‘social’ and ‘technical’ should indicate a particular focus rather than referring to pure instances of a-social technology or a-technological sociality. Still, in the discourses about software engineering, including feminist contributions, there is a clear tendency to distinguish ‘technical’ and ‘social’ matters in an unproblematic way, like programming and user communication. How come?

The field of engineering studies suggests two main sets of answers. One emanates from the historical effort of engineers to define their professional knowledge as a mix of science and technical competence, obscuring the importance of sociotechnical processes.²² Or, as Lois Bucciarelli observes from engineering education: “The student must learn to perceive the world of mechanics and machinery as embodying mathematical and physical principles alone”.²³ This strand of research suggests that the social/technical binary is reproduced as an accounting device because it allows engineers to relate to ‘people issues’ while retaining the important status of being ‘scientific’. The other and larger body of research has been mainly concerned with the history and sociology of the engineering profession, analyzing autonomy,²⁴ identity,²⁵ professional organization and status,²⁶ class and occupational role,²⁷ and national differences.²⁸ From these studies, we learn that the social/technical binary and the implied hierarchy, which privileges the ‘technical’ over the ‘social’, are productive because they legitimize the more powerful and esteemed social position of engineers, compared to many other employees and professionals.

Faulkner succinctly summarizes one important effect of the social/technical binary by observing that the two sides are deemed mutually exclusive, that engineers cannot be into both technology and people at the same time.²⁹ With respect to software engineers and communication, this would support a hypothesis that they would prefer

²⁰ Faulkner, “Dualisms, Hierarchies and Gender in Engineering”, 2000.

²¹ Bijker et al., *The Social Construction of Technological Systems*, 1987.

²² Downey and Lucena, “Engineering Studies”, 1995; Vincenti, *What Engineers Know and How They Know It*, 1990; Sørensen, “The role of Social Science in Engineering”, 2009.

²³ Bucciarelli, *Designing Engineers*, 1994, p. 108.

²⁴ Ritti, *The Engineer in the Industrial Corporation*, 1971.

²⁵ Gouldner, *Patterns of Industrial Bureaucracy*, 1954.

²⁶ Layton, *The Revolt of the Engineers*, 1986; Larson, *The Rise of Professionalism*, 1977.

²⁷ Zussman, *Mechanics of the Middle Class*, 1985; Whalley, *The Social Production of Technical Work*, 1986; Sørensen, “Engineers Transformed”, 1998.

²⁸ Maurice et al., *The Social Foundations of Industrial Power*, 1986; Meiksins and Smith, *Engineering Labour*, 1996.

²⁹ Faulkner, “Dualisms, Hierarchies and Gender in Engineering”, 2000, p. 764.

communication to be externalized, so to speak, and made into a separate specialty. However, as we note above, this view is contested within the profession. Thus, we need to explore the content of what software engineers consider to be their technological expertise. In the abstract, we know that they combine ‘technical’ and ‘social’ objects to construct or do technology, but how is communication with customers or users enacted in their practice?

Perhaps the social/technical binary allows for different relations between the ‘social’ and the ‘technical’? The traditional engineer *cum* manager was understood as a person in transition from exercising technical skills to become dependent on social competences.³⁰ In this case, the relationship between ‘technical’ and ‘social’ abilities was dichotomous and vertical, in the sense that engineers’ career trajectories meant decreasing dependence on technical knowledge and increasing employment of social skills. This contrasts with current ideas about transdisciplinary practices, where specialists from several fields work together in teams.³¹ The relationship between the ‘social’ and ‘technical’ remains dualist—engineers are the experts on technology, social scientists on social issues, etc.—but that relationship is also horizontal; the different forms of expertise interact on the same level.

To pursue these ideas, we will employ Erwin Goffman’s concept of *framing*. This concept refers to the way “definitions of a situation are built up in accordance with principles of organization, which govern events—at least social ones—and our subjective involvement in them”.³² It is a concept that helps to clarify the acts of definition involved in the use of concepts and their performance. We shall study how software engineers frame software engineering as well as communication with customers, focusing on what aspects are understood to be characterizing features of software engineering and communication respectively, the consequences of this framing, and the actual definition.

In doing so, we draw on the use of Goffman’s thinking about framing processes found in Michel Callon’s efforts to theorize the construction of economic markets.³³ Here, Callon analyzes the preconditions of economic calculation related to markets by studying how the objects of calculations are framed—what aspects are to be a part of the calculation and what features are left out or externalized? Similarly, we see framing of software engineering and communication as processes where some elements are defined as being on the inside, as accepted qualities of the profession, while others are seen as being on the outside, as external and thus irrelevant. Is communication inside or outside of software engineering? To what extent are ‘technical’ aspects of software engineering externalized with respect to communication? Or ‘social’ dimensions?

Processes of externalization, argues Callon, may create problems, making calculations misleading. For example, if destructive effects of pollution from making a given product are externalized, this may make the product too cheap and the rate of consumption too high. Callon terms such problems as overflows. In our case, for example, if ‘social’ dimensions are externalized from communication by making

³⁰ Sørensen and Levold, “Tacit Networks, Heterogeneous Engineers, and Embodied Technology”, 1992.

³¹ Gibbons et al, *The New Production of Knowledge*, 1994

³² Goffman, *Frame Analysis*, 1974, pp. 10-11.

³³ Callon, “An Essay on Framing and Overflowing”, 1999.

communication into a technologically-focused practice, this may lead to overflows in the form of insufficient or misleading exchanges of information. A framing that externalizes technological knowledge, for example by a singular focus on empathy, may produce a different overflow with similar consequences in terms of failures. Are there such overflows in relation to the framing enacted in software engineering practice?

Thus, our research questions may be rephrased as an interest in two related framing processes. First, how do software engineers place communication skills in relation to their framing of their discipline/profession? Are such skills on the inside, as important to their practice, or not? How should they be acquired, and by whom? Second, how do software engineers frame communication with customers and users? Is it defined as a social skill related to personal qualities or characteristics like gender, something some individuals more than others tend to be good at, or is their core technological competence seen as being inside communication?

We have chosen to pursue these questions by analyzing the education as well as the practice of software engineering. This is because we believe that framing processes may be more transparent in educational settings. Educators tend to try to clarify their pedagogical aims and strategies, and this may be helpful. The analysis of software engineering education is mainly meant as a tool to help understand the framing that takes place in software engineering practices.

Methodology

Our study was done in Norway, a small but wealthy, advanced economy with a fairly large software industry, consisting of small- and medium-sized companies. In terms of our introductory discussion, the country provides an interesting context in two ways. First, there is a long tradition of industrial democracy and an outspoken engagement to involve users in the development of technology, not the least with respect to software engineering.³⁴ Second, gender concerns have a high degree of visibility in many areas. The percentage of women working in software engineering is fairly low, probably around 10-15 percent, which has motivated several campaigns to recruit more women. Thus, we believe that Norway provides a context that could be expected to be conducive to a framing of software engineering that would make communication with customers and users a centerpiece. The widespread emphasis on gender issues and efforts to include women in areas like software engineering also suggest that we may expect to find a framing of communication that privileges women and that forms a potentially attractive part of software engineering to women.

We have collected data from two settings, education and industry, mainly based on qualitative interviews, to some extent supplemented by written sources. The framing of software engineering and of communication in the education of software engineers has been studied at the Norwegian University of Science and Technology (NTNU) in Trondheim. NTNU is the largest institution educating graduate-level engineers in Norway and the second largest university in the country, with around 18,000 students. With respect to our research questions, it is important to note that for many years the University has run a relatively large initiative to recruit more women to

³⁴ See, e.g., Gregory, "Scandinavian Approaches", 2003.

study software engineering. Thus, NTNU offers an interesting opportunity to analyze the role of gender in the framing of software engineering and of communication. In addition to collecting campaign material, we interviewed the University's Prorektor as well as six faculty and seventeen students at NTNU's Department of Computer Engineering and Information Science (IDI) about the initiative, which had a clear focus on the importance of communication with customers and users.

The analysis of framing issues related to software engineering practices is based on interviews in three medium-sized Norwegian software companies. For reasons of anonymity we have named them Aconsult, Programconsult and Webdesign. The first company provided consulting services related to selection and adaptation of ICT systems, including systems architecture and infrastructure. The second company sold tailored software to public utilities, including consulting services and training. The third company provided web technology to public institutions and private industry. The companies employed between forty and 200 people. We interviewed between eight and eleven persons from each of them, for a total of twenty-seven interviews. About half of the interviewees were women.

All interviews were transcribed verbatim. We have translated the Norwegian interview quotes into English, trying to retain their oral quality. The most common job titles were systems developer, consultant, advisor, or senior consultant. Table 1 provides an overview of the educational background of the interviewees. While there is no doubt that the three companies were in software engineering in the inclusive way we have defined it, it was a minority that were formally trained as software engineers, even if nearly all were engineers or scientists. Rosalind Williams argues that technology is no longer the exclusive domain of engineers, that engagement with technology has far outgrown single professional status.³⁵ We believe this to be true for software engineering. Table 1 supports this assertion. Moreover, the interviewees without a degree in software engineering claimed to have appropriated core software engineering skills through practice and/or self-education. In the companies, no clear-cut distinction was made between those with a formal education in software engineering and those who had acquired such competence in other ways. We also believe this element of educational heterogeneity is fairly typical for the profession because of its history of recurring recruitment problems.³⁶

Table 1. Education of interviewees. Absolute numbers.

Education	Number
M.Sc. Software Engineering	6
M.Sc. Computer Science	4
B.Sc. Software Engineering	3
B.Sc. Computer Science	3
M.Sc. Other engineering disciplines	5
B.Sc. Other engineering disciplines	3

³⁵ Williams, *Retooling*, 2002.

³⁶ Friedman with Cornford, *Computer Systems Development*, 1989.

Other	3
Sum	27

In the analysis of the data, we have used grounded theory methodology with open coding.³⁷ We have studied statements explicitly about communication but also about activities related to customers, users, selling, running of projects, etc. As an investigative tool, we have used Atlas software for analysis of qualitative material, which is helpful for testing coding and identifying relevant parts of the interviews. Informants have been given token first names: those from Aconsult beginning with an A, those from Programconsult beginning with a P and those from Webdesign beginning with a W.

In the following section, we start out by analyzing the framing of software engineering and of communication at NTNU, in the recruitment of women initiative and among faculty and students. The campaigns to recruit more women students proposed a reframing of software engineering education from a technical to a social focus. As a result, one of our research questions is: What did that mean for the framing among faculty and students and in terms of curricular reform? We then move on to look at the framing processes as accounted for by the interviewees from the three companies. Here, we consider a second research question: Do we find a similar reframing as the one observed in the NTNU campaign? How do they frame software engineering and communication?

Recruiting More Women Students: An Occasion to Reframe Software Engineering Education?

In 1996, the Norwegian University of Science and Technology (NTNU) initiated a comprehensive initiative to recruit more women to study computer science/software engineering. The initiative commissioned several advertising campaigns that have presented a gender-focused re-interpretation of software engineering, mainly emphasizing the importance of communicative skills and social dimensions in this line of work.³⁸ In the 1997/98 campaign, women were presented as better computer experts than men because of their alleged superior communication and people skills: “Women listen. Women talk with each other. Our experience is that women ask ‘why’-questions, like: What is the point with this button? Who will benefit from this function? How will the user understand how to begin? ‘Listen!’ women say. ‘This is too difficult! We must make the system easier!’ (...) This is why we want women. This is why the proficient staff of the [computer and software engineering] department is ready to welcome you. Because it seeks what you’ve got: femininity.” Thus, the campaign booklet framed communication as predominantly about empathy, the ability to understand and identify with other people’s perceptions and emotions.

In fact, this campaign booklet offered two representations of software engineering that were juxtaposed; as a mainly people-oriented, socially focused profession versus an occupation focused on narrowly technical problem-solving. Thus, the rhetorical strategy of the campaign was to claim that the first representation was

³⁷ Strauss and Corbin, *Basics of qualitative research*, 1990.

³⁸ For a more detailed analysis, see Lagesen, “Advertising computer science to women”, 2003.

the correct and sensible one, and one that would make software engineering more attractive to women. Compared to the arguments to reform the education of software engineers by making it more hybrid (reviewed in the introduction), the NTNU campaign was mainly based on a different strategy, to improve the non-technical skills of the profession by recruiting students—women—who were assumed to have attained such skills already before they became students. The play on stereotypical images of women as a way to cater to the need for communication skills made it less necessary to change the curriculum. Rather, the main aim was getting more women students. Implicitly, it was assumed that the communicative skills needed would develop through interaction and collaboration among the students.

Thus, while the NTNU campaign reframed software engineering to include greater emphasis on communicative skills in relation to customers and users, potential contributions from the social or human sciences remained externalized. The subtle change in the topology of the social/technical binary presented by the campaign emphasized communication competence as critical to the design of good technology, even if the attainment of such expertise was not really reflected in the curriculum. Several of the NTNU faculty involved in the campaign mentioned that the software industry had complained that their graduates were too narrowly focused on technology. This was an indicator that the traditional framing of software engineering had resulted in an overflow: the production of graduates that lacked the skills industry needed. One of the professors expressed it like this when explaining his support of the campaign: “We interact with values, languages, communication and visual design. And all this made us want to include the reservoir of ideas, knowledge and attitudes that women possess into the discipline. It was very ‘deep-rooted’. We wanted more women in order to develop the *discipline*.”

Did the campaign’s effort to reframe software engineering and the observed overflow in terms of lacking non-technical skills result in curricular changes? Mainly, the answer is no. In the end, the only change was to establish a course that introduced the students to the computing professions and the software industry. Still, most of the women students recruited through the campaign made no complaints about this lack of curricular reform. They had chosen to study software engineering because they were interested in science and mathematics and thought the program offered good career opportunities. Only one of the women students interviewed complained that the curriculum turned out, she said, to be mainly ‘technical’ and not at all providing ‘social’ skills. In general, the student interviewees from NTNU confirmed our interpretation of the framing of software engineering as overwhelmingly technically focused.

Thus, the rhetorical effort to reframe software engineering into a profession of skilled communicators found little resonance among students and faculty. They wanted more women students, but “We thought that our program did fit women”, as one professor put it. Again, the goal was not curricular reform but rather to achieve gender balance. The dominant framing of software engineering as technically focused coincided with a framing of communication skills as an opaque competence to be developed outside the educational setting. The campaigns tried to reverse the status hierarchy of the social/technical binary, but the dualism itself was not challenged;

rather, it was strengthened. The reframing efforts were dismissed by the faculty as advertising and, according to students' accounts, overlooked.³⁹

Code and Communication: The Role of Technical Knowledge in Interactions with Customers

The analysis of the framing of software engineering and of communication in education at NTNU showed not only the prevalence of a strict social/technical binary but also that only 'technical' elements were defined to be on the inside of their core competence. Apart from the efforts to recruit more women students, social dimensions of communication were put outside of the frame. As we had expected, gender concerns worked to put people skills more visibly on the agenda, but not in a pervasive and stable manner.

Of course, the externalization of social skills like communication is easy to uphold in a context dominated by fairly abstract, mathematically-oriented courses where practical applications tend to be theoretical. Thus, we could expect that the framing of software engineering and of communication would be different in software engineering practice. To what extent would they draw on a social/technical binary in their accounts, and to what extent would they transgress the dualism?

At Aconsult, a medium-sized company engaged in giving advice about and designing and adapting computer systems, we were told by the junior employees that they were very much engaged in programming work, in order to develop tailor-made systems or adapt generic, off-the-shelf software. Most of their job was done at customers' premises and the customer was an important figure in their accounts. So, when asked about what she needed to know to do her job, Anette—a recently employed consultant in her late 20s—stated, "(T)he obvious thing is, that you need to know how to program. True, that's what I sit and do all the time. But really, it is just as important to relate to the people around you. Because, there is a customer. You know, you're supposed to develop something they need. So, if you have your own opinion, completely independent of what they want, then you have missed out."

Anette described a situation where both technical and communicative skills were in demand. She had to be able to write code—that was what she mostly did—but she also required some understanding of what the customer wanted or needed in order to supply a useful product. Later in the same interview, she emphasized the precariousness of an integration of the two competences: "(L)ike, when you start the project, you need to find out what their needs are. And then you have to sit and discuss ... you make these drawings, you know, of what the system is supposed to do. And then, you're quite dependent on reaching an agreement with the customer, in addition to understand what they want. And then, to recognize how this may be done in the simplest possible way. And, not everybody is suited to do this. I haven't tried it myself either, I have only observed it. Because it is a rather time-consuming process. You know, often, like in this case [referring to an ongoing project], there are many sections [in the customer company] that make demands, and to get it all together so that everyone are happy, that is incredibly difficult!"

In this manner, Anette first framed software engineering as an integration of technical and communication skills, and second, framed communication as being, in

³⁹ See also Lagesen, "Strength of numbers", 2007.

an important way, about technological options and limitations. However, there was a tension in her argument between the emphasis on technical knowledge as a key to interact productively with customers and her observation about the skills that were needed to help negotiate a diversity of articulated needs in the same company. Clearly, she was aware of the potential overflow if software engineering and communication were framed in such a way that people skills were externalized. Nevertheless, she avoided invoking a clear-cut social/technical binary in her account. Rather, she blurred the relationship between technical knowledge and communication competence. How can we make sense of this?

To begin with, we want to emphasize that all our interviewees agreed that programming was the basic competence of software engineers. This was a core skill that everybody needed. Anton—a senior consultant in his early 30s—explained this by claiming that: “[Programming] is important, you know. It is such a very, it is a fundamental thing. At least within what we’re doing, so is it a very important ... how should I put it? A foundational pillar, really, in what you do.” However, he went on to emphasize that programming in itself was not a sufficient skill, and it was not necessary to have a comprehensive knowledge about it: “But just as important is what you learn around it. (...) I use to say that programming is something you can learn on your own (...) during a project, for example.”

Accordingly, it varied in what ways programming was considered important. At Aconsult, they practiced a division of labor where junior employees like Anette did most of the actual writing and testing of code. Since the juniors had graduated recently, they were supposed to be up-to-date with respect to programming languages and related technologies. To the more senior employees, programming skills played a different role. Arthur, another senior consultant in his early 30s, told us that he did not enjoy programming in particular. He described himself as an architect, but qualified his position by adding that: “But, you see, I do know how to program, so I’m able to read code and understand what’s there.” In this manner, he claimed to be on the inside of software engineering. He then went on to elaborate what knowledge of programming did for him:

If you take a standard project, there is like a design period, then you’ve got an implementation period where the programming takes place, and then you’ve got a test period. Typically, the architect will actively take part in the analysis, will manage a great deal of the design part, will be available during implementation for clarifications and then, in a way, be able to talk to the programmer, be able to understand the kind of problems you may run into and make a decision with respect to how this should be resolved. And you will be there in a test stage as well, available that is. To see if this thing actually does what one has promised. But the architect’s knowledge of programming, that is not really important in itself in the projects. It’s more about having an understanding of it, to understand possibilities and limitations and to be able to communicate well with those who have the most prominent roles in the various stages of the project.

Thus, when Arthur accepted the status of programming as the core skill, it was not because he found it important to write code. Rather, he needed this knowledge to understand and communicate. This was a fairly widespread way of thinking, also in the two other companies. Walter, a system developer in his early thirties working at

Webdesign, a company that provided web technology and design to a diverse group of customers, including local and central government, explained it in a different way. He collaborated with people originally trained in graphic design and noted that “Like, they have been working within the profession [software engineering] for many years, all of them. However, often, you would wish they knew more technical stuff (...). Frequently, they come with ideas that are very difficult to carry through, afterwards.” The challenges could be substantial, since these ideas often already had been sold to customers.

Many also pointed to the need for more comprehensive technical knowledge than just programming. Petra helped us to get this clarified. She was in her late thirties and worked as a consultant at Programconsult, a company that provided tailored software to utility companies. Petra found much pride and satisfaction in her ability to communicate with customers and the challenges she encountered with respect to identify what they wanted. “Because, often, it is quite difficult when we arrive at a customer’s and they say that ‘this is what we want’. ‘OK, but why do you want precisely this?’ Well, no, they aren’t exactly so sure about that. Rather one has to be with the customer and talk to them, and ask and dig out and find out what’s behind their inquiry. Like, what kind of processes our product is expected to support.” Petra claimed that one could not do this kind of work without an intimate, technical knowledge about Programconsult’s products, but she also saw her own people skills as vital. They included qualities we interpret as empathy and sensitivity. “You know, communication is just as much what is not said as what is actually spoken. And then it is about nonverbal things as well as being able to improve the mood and things like that”.

Compared to the framing of software engineering that we observed at NTNU, our interviewees in the three companies gave a markedly different account. First, we did not find a clear-cut social/technical dualism. While there was no doubt that software engineering was framed as having a technical core, communication with customers loomed large in importance for all interviewees. Moreover, communication was generally framed as a hybrid, sociotechnical activity. With a couple of exceptions, the interviewees did not use concepts like ‘social’ to describe any of their tasks. Rather, their accounts suggested that they might join in with Dahlbom and Mathiassen to portray software engineering as an extended technical practice.⁴⁰ However, it is more tempting to characterize their framing of both software engineering and communication with customers as diffuse and indistinct, with unclear boundaries. We wonder if such an interpretation of their framing is adequate.

What Software Engineers Need to Know – and Why

When Wanja, a developer in her early 40s, was asked about what kind of knowledge she needed in her job, she responded enthusiastically that

What I find to be great fun is that I feel I get to use very different kinds of knowledge. It is so dependent on the project. Something I found really great, it was nearly frustrating enjoyable, was to participate in a so-called requirement study. It was a really large project, so we were engaged in requirement analysis

⁴⁰ Dahlbom and Mathiassen, “The Future of Our Profession”, 1997.

for half a year. About what kind of system we were to design. Heaps and heaps of communication with users, and all the users say different things, and then it is on to finding what they actually need. And some times, it is not the users who have the right answer, you know. It was great fun!

However, when she was prompted, it became clear that ‘the very different kinds of knowledge’ mainly were technical, even if she maintained the need for knowledge about people.

Wera, who was in her early 30s and worked as senior interaction consultant, described the issue in a pragmatic way by noting that “This is what provides us with a living (...). We sell consultants, we sell programming services”. Pia, a product manager and in her early thirties, put it in a similar way: “It is the customer we live by, but the focus is no doubt most strongly on how the product is designed and what kind of technical niceties it offers.” The core business of all three companies was technology. There was no doubt about that. So what was the role of communication skills?

When we examined the accounts more closely, it became evident that communication competence was described in a different fashion than technical knowledge. When interviewees talked about the latter, they referred to concrete methods, particular programming languages, etc. Wenche, an interaction designer in her mid-30s, began jokingly by saying that: “I’m no good at communication. I’m good at doing monologues.” She then went on in a more serious mode, observing that “Just the realization that there is a human being on the other end, there is someone who shall see this thing and use it and understand it. That helps a lot. It is not a computer system ... it is something a person is going to use. I think that helps. People have feelings, even.”

The language employed was generally without professional reference. Wera provided another good example when responding to the question about what kind of knowledge she needed in her job:

Really, I have to use a lot of psychology ... if, for example, you’re going to make a bid, you need to know how express yourself, and you have to consider the customer a bit, and we have to try to identify the person who’s going to read it (...). If it is a technical person who will read the bid, perhaps you should formulate it a bit different, more structured and matter-of-fact like than if it is ... an information type person (...). Then you should use big words. Tell them that this is going to be really great ... no, you have to use a lot of psychology.

Evidently, Wera used ‘psychology’ in its popular meaning as informal knowledge about how people think and behave.

Walter was one of the few who made a passing reference to social qualities as something that might be useful in the interaction with customers. However, such skills were not really important. What they had to tackle, was that “often, the customer has limited knowledge ... you say something to the customer, who sits there and nods and agrees, and then we come up with a solution, but no, it wasn’t this we thought about.” On this basis, Walter proposed an iterative strategy, founded on what he and his colleagues could come up with on a professional basis and take it from there. Communication with the customer would then be a series of proposed solutions, mainly grounded in software engineering knowledge.

This means that, in particular, programming was a backstage competence, a kind of linguistic resource that was needed in order to develop and communicate solutions and challenges. While this meant that technical and communicative skills were seamless and difficult if not outright impossible to distinguish, these skills were not on the same par. This was confirmed by Ann, a senior advisor in her fifties, when she described the importance of technical skills as a prerequisite to communication: “It is important to be able to pose questions. To be able to help ... or, in a way, to be a link between developers and customers, for example. In a situation where you not the least understand the language that is used. And, from an overarching point of view, be able to see and read, to draw some solutions. So, you go a bit into the fundamentals.” And the fundamentals, as we see, were always ‘technical’.

Thus, the framing of software engineering and of communication was not really that much without boundaries. There was a social/technical binary but it was expressed indirectly by using a professional, knowing voice with respect to ‘technical’ issues and a more casual, lay tone when reflecting about communication competence. People skills were not externalized, neither from software engineering nor communication with customers, but they were integrated on different terms, compared to technological knowledge. Did this resonate with the organization of software engineering work?

Coding or Communication?: Tracing a Division of Labor

In the introduction, we saw how scholars in such diverse fields as technical communication, software engineering research, and feminist studies of women in computing suggested a division of labor between technical experts and specialists focused on communication and similar people skills. Moreover, our reading of the engineering studies literature suggests that a social/technical binary might be supported by engineers seeking status by making links with ‘hard’ science rather than ‘soft’ types of knowledge. Do we find such distinctions in our three companies, and if so, what does that tell us about the way software engineering and communication is framed?

As a start, we would like to stress that our interviewees were unanimous in their declaration that customers were of prime importance to their work. Pauline, a junior system developer in her mid 20s, expressed this vividly: “[T]o see how they [users] use it and how it may be done ... and further development and getting the software even better, *that* part is very rewarding, like, you see the point about what you’re doing also. And then, you get inspired to continue working when you see that there is really someone who use it and are dependent upon what you have made.” Interaction with customers was not only rewarding, it was also considered to be a vital input to their job. Paul, a consulting engineer in his late 30s, went as far as saying that “The day the customers say nothing, then we have an enormous problem. Where should we then go and what should we make now ... they are the ones providing inspiration ... it is very, very important to have that kind of dialogue with the customers.”

Thus, contrary to expectations, communication with customers had a high status in all the companies, higher than, for example, advanced programming skills. This was clearly expressed by those who lacked this possibility. The assumption is also supported by the fact that most of the employees who interacted substantially with customers, tended to be senior and higher in rank than those who did not, although the

pattern is not pervasive. The three companies all provide some division of labor between those who mainly do technological development and write code and those who do, more or less, communication with customers.⁴¹

As already noted, in Aconsult, the main distinction was between juniors and seniors. It was the latter group that had the main responsibility for interacting with customers. Programconsult had organized it in a more strict and formal way. They had one department that was responsible for development work, where the employees were without access to customers. This access was taken care of partly by marketing people who initiated sales and partly by consultants who finalize contracts and maintain the contact with customers throughout the project. Finally, in Webdesign, there was a tendency that the graphic designers who were responsible for the appearance of web pages, also took care of the interaction with customers. However, the division of labor with respect to the communication with customers was less pronounced than in the other two companies.

Thus, overall, there was some division of labor with respect to communication with customers, but it was not strict. Also, the underlying framing of software engineering and communication was unclear. Were those who were given the main responsibility for interacting with customers also perceived as having a particular competence due to some form of training? Or was it a matter of personal qualities? Or was experience the decisive feature?

In the Name of Experience

We argue above that skills related to communication were described in the interviews in a more casual and “lay” way, compared to a more assertive and professional tone with regard to technical knowledge. This suggests that communication competence has been acquired neither through education nor through work-related training. This was confirmed through all interviews. No one had attained such competence as part of their curriculum, and none of the interviewees had attended professional training for communication. Some believed that there might have been such opportunities, but they mainly talked about the courses as some form of general management or marketing education.

In particular the interviewees from Aconsult agreed in their assessment of what gave communication competence: experience. Experience, learning by doing, was a central feature of their work, partly to find out how to transform theory to practice, but not the least to extend their knowledge about how to understand and solve customers’ problems. This also implied that experience was a key to their particular form of career; a movement out of technically focused work, above all programming, to become more full-fledged sociotechnical operators, transforming options and problems to strategies and solutions. Compared to junior employees, senior consultants and advisors performed a wider set of tasks in relation to a much more diverse set of customers. Mainly, their careers did not lead to managerial positions but to roles as experts accomplished in diagnosing the problems of a diverse group of organizations and in seeing how solutions could be designed and orchestrated. In addition, they had

⁴¹ However, as shown by Levold, “Kundekommunikasjon”, 2007, communication with customers is also something that has to be restricted and controlled. It is important but it should not demand too much resources.

status within the company as having broad knowledge and experience with different kinds of organizations and business operations.

The nature of this expertise was rendered complex. Agnes, who was an advisor in her mid-40s, told that her learning usually was based on a multitude of sources, even if she emphasized that working with people who knew things she did not was the most efficient way to learn: “You know, working together with other people who have a different competence than yourself, that’s what I think you learn the most from.” She went on to add that:

[P]articularly what I’ve learnt at school [university] and such, it becomes like a kind of bedrock basis. That I may draw upon, under various circumstances. I mean, one has a solid understanding covering technology from the core to the more advanced. (...) Clearly, most of the learning happens with the customers, that’s where you do the daily work. But, it’s a little bit both-and. Or, it’s there you learn every day. Because you encounter new problems every day and things like that. But, you know, it’s very important to be professionally updated. If not, you will be left behind ... and in particular with information technology, where things happen very fast.

Note that Agnes did not in particular address communication skills; truly, she was mostly concerned with her ‘technical’ skills. For her, learning from being with customers seemed to be more related to ‘technical’ issues than to communication and other people skills. Actually, this was common. Experience was important to learning, but in a general sense to enhance one’s skills in formulating and solving relevant problems. Thus, when experience was important for determining who would be given access to interact with customers—in particular in Aconsult but also in the two other companies—it was more in recognition of the person’s technical competence than communication skills.

There was no explicit argument about personal qualities either, even if it were hinted at in some interviews. The expectation that gender would elicit some reflections of that sort, in line of the feminist studies of women in computing reviewed in the introduction, was not met, either. Certainly, Agnes made a political statement about women’s qualities:

I believe technology has many soft aspects, really. Because ... among other things, you are supposed to make systems that shall work for people (...). And good at talking to users, skilled at understanding what users really need and able to transform this into requirements for a system or a model, make it systematic. That’s what I think. And then, women are very clever project managers.

The gender politics of this account resonates nicely with the main drift of the NTNU campaign discussed previously. However, Agnes was really the only one among our interviewees to express such sentiments. All of the other stated bluntly that they did not think there were gender differences in terms of communication skills or other abilities. Gender was definitely externalized in their framing of software engineering as well as communication. Opportunities to interact with customers were described to us as a consequence either of experience—meaning the accomplishment of broad experience in transforming software technology into usable products—or as a result of your position in the company. This does not mean that personal qualities were

irrelevant, but rather that they seldom were articulated specifically as important by the interviewees.

In the section about education we learnt that faculty had observed an overflow from the way they had framed the education; that their candidates were too technically focused. Were there similar overflows from the framings performed by the interviewees in the three companies? Or did they think that their communication skills and the way they communicated with customers were satisfactory? Was it ok to learn communication mainly through experience?

To begin with, most interviewees told us that they experienced some kind of overflow, in the meaning that it often proved difficult to finish projects on time in a satisfactory way. Alexander told us plainly that “I still haven’t finished an IT-project that went well. Or, rather, that was brought to an end on time.” He was a bit more outspoken than the rest, but there was a general agreement that it was difficult to finish project without extra, unplanned efforts and often with some delays. The most common explanation offered was lack of resources, indicating that the project had been calculated in a too optimistic way or had not been planned properly. However, lack of proper planning may be an euphemism. Palma tried to explain that “Always, something unexpected surfaces. And then there are issues that, apparently, at the outset didn’t matter at all, which suddenly kick in and have huge effects [...], So, things may not have been defined in sufficiently clear way and specified.” Pia provided a similar account: “But often, problems emerge because we haven’t been able to specify requirements or what we are supposed to make well enough, and this relates to communication with the customer, project manager and developer.” Such problems could cause substantial increases in cost.

We cannot claim that the way the social/technical binary was managed through the framing of software engineering and communication, was producing the above-mentioned overflows. However, what we can say is that the practice of software engineering in the three companies caused overflows and that these overflows somehow were related to the way requirements were understood and calculated in the projects. Thus, there were definitely problems with respect to the management of the supplier-customer relationship.

Conclusion: Towards a Changed Relationship between the Social and the Technical?

The NTNU campaign to recruit more women software engineering students made an effort to change the representation of software engineering from being about working with machines to working with people. This should make the profession more attractive to women, but it also reflected a point of view shared by many stakeholders; the need for better social skills in software engineering.⁴² While the NTNU campaign could be criticized for taking the argument too far, there seemed to be a widespread opinion that software engineering had been framed in a way that was not satisfactory to industry, with programming dominating the inside and people skills placed on the outside. Nevertheless, no curricular reform was deemed necessary and faculty and students seemed to agree foregrounding a technical focus, while externalizing ‘social’ competences like communicating with customers.

⁴² Lagesen, “Strength of Numbers”, 2007, elaborates this observation.

In contrast, our interviewees from the three software companies presented communication with customers as vital. However, rather than framing communication as a personal skill, external to the technical practice of software engineering, like faculty and students, they emphasized how communication with customers above all required technical knowledge. When our interviewees talked about their work, learning, and engagement with communication, they did not directly invoke the traditional social/technical binary like Faulkner observed.⁴³ Instead, they insisted that their technical knowledge formed the basis of their interaction with customers and their ability to find good solutions. Thus, they saw programming as well as other software engineering knowledge as a linguistic resource that helped them to communicate adequately about the options. Moreover, we interpret their accounts to suggest that they do not start out by diagnosing the situation of their customers in terms of organizational challenges, using this as a basis to negotiate a shared understanding that forms the basis of technical requirements. Rather, they provide a specific diagnosis based on what they think available technologies may do for their customers. What they try to communicate to customers is a technology-based solution to what the customer may think of as a social or organizational problem. Thus, they enact a ‘technification’ of customer practices by introducing new software, and perhaps also new hardware.

The resulting framing of software engineering and of communication with customers had a subtle but definite focus on technical knowledge. Our interviewees did not openly externalize issues related to communication with customers. However, as is hopefully clear from our analysis, while they did not use the social/technical dualism in an explicit way, they talked about ‘technical’ and ‘social’ competences in a different way. The first was the object of a professional discourse, the latter was not. ‘Social’ aspects were not externalized as such, but a professional approach to such issues—for example based on social science knowledge--was.⁴⁴ This is also evident from the absence of professional training with respect to enhancing communication skills as well as competence in the analysis of organizations. In addition, we have observed that gender did not work in the expected way as a resource to elicit reflections around these issues. Only one of the interviewees did that, while the others refused.

Thus, what we have tried to show is that the exercise of software engineering expertise is understood by insiders as being clearly within the boundaries of the technically focused profession due to the central role attributed to appropriate technical knowledge. This knowledge is seen to allow them to solve problems, for example by writing code or selecting the appropriate software, but also to provide a language to analyze and transform the challenges offered by customers into what Joan Fujimura terms doable problems,⁴⁵ meaning problems people with software engineering expertise can take care of. From this point of view, communication between software engineers and customers is, professionally speaking, mainly technical talk supported by lay skills. It is a co-production of doable problems and negotiations about the relevance of proposed solutions.

⁴³ Faulkner, “Dualisms, Hierarchies and Gender in Engineering”, 2000.

⁴⁴ See also Sørensen, “The Role of Social Science in Engineering”, 2009.

⁴⁵ Fujimura, *Crafting Science*, 1988.

This is not to say that current practices of software engineers necessarily are the best possible or that their communication skills are adequate. Actually, we do not believe this to be the case, not the least because of the reported overflows in terms of project overruns. We started out by observing an ambiguity in the thinking about the performance of communication in engineering generally or in software engineering more particularly. This ambiguity centered around two different strategies to deal with the issue, either by making communication into a specialty or as integrated in the core of the profession.

Our findings suggest a different understanding. While we have observed practices that give some (e.g. senior personnel) more responsibility for communicating with customers, there was no clear-cut specialization going on. This claim is supported by the absence of particular training to enhance relevant communication skills. Rather, communication was framed by our interviewees as the exercise of technological knowledge in combination with competences that most normal people would have, albeit to a variable degree. In this way, communication with customers was integrated into software engineering as a combined professional/lay skill.

As noted earlier, it is a basic insight from science and technology studies that to claim the existence of social/technical binaries are misleading. In fact, our interviewees did not provide an upfront social/technical binary either. Seemingly, they did not walk this line. However, as we have seen, a more thorough analysis tells otherwise. Our interviewees did distinguish between ‘social’ and ‘technical’ knowledge, but they did so in a subtle manner by accounting for knowledge about ‘social’ features in lay terms, often as personal skills, while they referred to technological competence in a professional, knowing manner. This shows how complex the enactment of the social/technical binary in engineering may be.

To some extent, this finding resolves the paradox observed in some of the feminist literature between the appreciation of hybrid skills and of women as particularly good at this, and the lack of real rewards given to women employees.⁴⁶ It also emphasizes the challenges we observed from the field of technological communication and from debates about the training of software engineers. To choose between educating engineers to become better at communication and making communication with customers and users into a specialty is clearly not straightforward since knowledge of technology seems so important to the content of the communication process.

It is tempting for social scientists to argue that the social/technical binary produced by our interviewees should be replaced by interdisciplinary collaboration between engineers and humanists/social scientists with relevant competence. The kind of practice described by our interviewees suggests a problematic asymmetry between technology, which is to be developed in a professional way, and social skills that are seen as personal and thus not an object of research-based improvements. Consequently, the asymmetry makes the potential for progress one-sided. However, as Lucy Suchman has argued, interdisciplinary collaboration is no straightforward solution.⁴⁷ To what extent are humanists or social scientists able to describe customer

⁴⁶ Woodfield, *Women, Work and Computing*, 2000.

⁴⁷ Suchman, “Practice-Based Design of Information Systems”, 2002.

requirements in a way that engineers may appreciate and make use of? This is not an easy task.

However, observing that the dualism of the social and technical rather should be understood as a complex distribution of skills and concerns may provide a more fruitful point of departure to develop alternative approaches. This raises several challenges. With respect to the field of technical communication, one may need to be more concerned to emphasize the non-technical, non-instrumental aspects of such tasks, but above all to focus more explicitly and extensively on communication with non-engineering professions. This point is also relevant to software engineering education. Here, one may also consider the realism of such proposals as the one forwarded by Dahlbom and Mathiassen,⁴⁸ demanding very comprehensive training. In addition, there is also a challenge to human and social sciences to provide (some of) their graduates with the skills needed to interact constructively with engineers.⁴⁹

Acknowledgements

The research underlying this paper has been supported by Research Council of Norway and Norwegian University of Science and Technology. We thank Nora Levold and Birgit Nestvold Singstad for collaborating in collection and analysis of data. In addition, we are grateful for very useful comments from the editors and four anonymous referees.

References

- Bijker, Wiebe E., Thomas P. Hughes and Trevor J. Pinch, eds. *The Social Construction of Technological Systems*. Cambridge: The MIT Press, 1989.
- Brookshire, Robert G., C. Steven Hunt, L. Roger Yin and Tena B. Crews. "An End-User Information System for the 21st century." *The Journal of Computer Information Systems* 47,3 (2007): 81-88.
- Buciarelli, Lois L. *Designing Engineers*. Cambridge: The MIT Press, 1996.
- Callon, Michel. "An Essay on Framing and Overflowing." In *The Laws of the Markets*, edited by Michel Callon, 244-269. Oxford: Blackwell, 1999.
- Dahlbom, Bo and Lars Mathiassen. "The Future of Our Profession." *Comm. of the ACM* 40, 6 (1997):80-89.
- Downey, Gary L. and Juan C. Lucena. "Engineering Studies." In *Handbook of Science and Technology Studies*, edited by Sheila Jasanoff et al., 167-188. Thousand Oaks: Sage, 1995.
- Faiola, Anthony. "The Design Enterprise: Rethinking the HCI Education Paradigm." *Design Issues* 23,3 (2007): 30-45.
- Faulkner, Wendy. "Dualisms, Hierarchies and Gender in Engineering." *Social Studies of Science* 30 (2000): 759-92.
- Friedman, Andrew L. with Dominic S. Cornford. *Computer System Development. History, Organization and Implementation*. Chichester: John Wiley & Sons, 1989.
- Friedman, Andrew L. and Joan Greenbaum. "Wanted: Renaissance People." *Datamation* 1 (1984): 134-144.

⁴⁸ Dahlbom and Mathiassen, "The Future", 1997

⁴⁹ See also Sørensen, "The Role of Social Science in Engineering", 2009.

- Fujimura, Joan. *Crafting Science*. Cambridge: Harvard University Press, 1996.
- Gibbons, Michael, Camille Limoges, Helga Nowotny, Simon Schwartzman, Peter Scott, and Martin Trow. *The New Production of Knowledge. The Dynamics of Science and Research in Contemporary Societies*. London: Sage, 1994.
- Goffman, Erwin. *Frame Analysis*. Boston: Northeastern University Press, 1974.
- Goles, Tim, Stephen Hawk and Kate M. Kaiser. "Information Technology Workforce Skills: The Software and IT Services Provider Perspective." *Information System Frontiers* 10 (2008):179-194.
- Gouldner, Alvin W. *Patterns of Industrial Bureaucracy*. New York: The Free Press, 1954.
- Gregory, J. "Scandinavian Approaches to Participatory Design", *International Journal of Engineering Education*, 19 (2003):62-74.
- Hartmann, Timo, Martin Fischer and John Haymaker. "Implementing information systems with project teams using ethnographic-action research." *Advanced Engineering Informatics* 23 (2009):57-67.
- Hawthorne, Susan and Renate Klein, eds. *Cyberfeminism. Connectivity, Critique and Creativity*. Melbourne: Spinifex, 1999.
- Iivari, Juhani, Rudy Hirschheim and Heinz K. Klein. "Towards a Distinctive Body of Knowledge for Information Systems Experts: Coding ISD Process Knowledge in Two IS Journals." *Information Systems Journal* 14 (2002): 313-342.
- Johnson-Eilola, J. and S.A. Selber, eds. *Central Works in Technical Communication*, New York: Oxford University Press
- Kelan, Elisabeth K. "Emotions in a Rational Profession: The Gendering of Skills in ICT Work." *Gender, Work and Organization* 15 (2007): 49-70.
- Kirkup, Gill, Linda Janes, Kath Woodward and Fiona Hovenden, eds. *The Gendered Cyborg: A Reader*. London: Routledge, 2000.
- Lagesen, Vivian A. "Advertising computer science to women (or was it the other way around)?" In *He, She and IT Revisited. New Perspectives on Gender in the Information Society*, edited by Merete Lie, 69-103. Oslo: Gyldendal Akademisk, 2003.
- Lagesen, Vivian A. "The Strength of Numbers. Strategies to Include Women into Computer Science." *Social Studies of Science*, 37(2007):67-92.
- Larson, Magali Sarfatti. *The Rise of Professionalism: A Sociological Analysis*. Berkeley: University of California Press, 1977.
- Layton, Edwin F. *The Revolt of the Engineers: Social Responsibility and the American Engineering Profession*. Baltimore: Johns Hopkins University Press, 1986.
- Lee, Chong Kwon. "Analysis of Skill Requirements for Systems Analysts in Fortune 500 Organizations." *The Journal of Computer Information Systems* 45,4 (2005):84-92.
- Levold, Nora. "Kundekommunikasjon i databedrifter: Mellom kontroll og empati. In *Kommunikasjonssamfunnet – Moral, Praksis og Digital Teknologi*, edited by Nora Levold and Hendrik Spilker, 142-160, Oslo: Universitetsforlaget, 2007.
- Litecky, Charles R., Kirk P. Arnett and Bipin Prabhakar. "The Paradox of Soft Skills versus Technical Skills in IS Hiring." *The Journal of Computer Information Systems* 45,1 (2004): 69-76.

- Maurice, Marc, François Sellier, and Jean-Jaques Sylvestre. *The Social Foundations of Industrial Power*. Cambridge: The MIT press, 1986.
- Meiksins, Peter and Chris Smith. eds. *Engineering Labour. Technical Workers in Comparative Perspective*. London: Verso, 1996.
- Moore, Krenza, Marie Griffiths, Helen Richardson and Alison Adam. "Gendered Futures? Women, the ICT Workplace and Stories of the Future." *Gender, Work and Organization* 15 (2008):523-542.
- Ritti, R.Richard. *The Engineer in the Industrial Corporation*. New York: Columbia University Press, 1971.
- Rosenstein, Allen B., Robert R. Rathbone and William F, Scheer. *Engineering Communications*. Englewood Cliffs: Prentice-Hall, 1964.
- Roze, Maris and Simon Maxwell. *Technical Communication in the Age of the Internet*. Upper Saddle River: Prentice-Hall, 2002 (4 ed.).
- Ruiz Ben, Esther. "Defining Expertise in Software Development While Doing Gender." *Gender, Work and Organization* 14, 4 (2007): 312-332.
- Sales, H.E. *Professional Communication in Engineering*. Hampshire: Palgrave, 2006.
- Scott, J. Blake, Bernadette Longo and Katherine V. Wills. "Introduction: Why Cultural Studies? Expanding Technical Communication's Critical Toolbox". In *Critical Power Tools: Technical Communication and Cultural Studies*, edited by J. Blake Scott, Bernadette Longo and Katherine v. Wills, 1-24. Albany: SUNY Press, 2006.
- Scandurra, I., M. H. Hägglund and S. Koch. "From user needs to system specifications: Multi-disciplinary thematic seminars as a collaborative design method for development of health information systems." *Journal of Biomedical Informatics* 41 (2008): 557-569.
- Spender, Dale. *Nattering on the Net: Women, Power and Cyberspace*. Melbourne: Spinifex, 1995.
- Strauss, Anselm M. and Juliet L. Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park: Sage, 1990.
- Suchman, Lucy A. "Practice-Based Design of Information Systems: Notes from the Hyperdeveloped World." *The Information Society* 18 (2002):139-144.
- Sørensen, Knut H. "Engineers Transformed: From Managers of Technology to Technological Consultants." In *The Spectre of Participation. Technology and Work in a Welfare State*, edited by Knut H. Sørensen, 139-160. Oslo: Scandinavian University Press, 1998.
- Sørensen, Knut H. "The Role of Social Science in Engineering". In *Handbook of the Philosophy of Science. Volume 9: Philosophy of Technology and Engineering Sciences*, edited by Anthonie Mejers, 89-111. Amsterdam: Elsevier, 2009.
- Sørensen, Knut H. and Nora Levold. "Tacit Networks, Heterogeneous Engineers, and Embodied Technology. *Science, Technology & Human Values* 17,1 (1992):13-35.
- Turley, Richard T. and James B. Bieman. "Competencies of Exceptional and Nonexceptional Software Engineers." *Journal of Systems and Software* 28 (1995): 19-38.
- Vincenti, Walter G. *What Engineers Know and How They Know It. Analytical Studies from Aeronautical History*. Baltimore: Johns Hopkins University Press, 1990.

- Woodfield, Ruth. *Women, Work and Computing*. Cambridge: Cambridge University Press, 2000.
- Whalley, Peter. *The Social Production of Technical Work: The Case of British Engineers*. Hampshire: Macmillan, 1986.
- Williams, Rosalind. *Retooling. A Historian Confronts Technological Change*. Cambridge: The MIT Press, 2002.
- Zussman, Robert. *Mechanics of the Middle Class: Work and Politics among American Engineers*. Berkeley: University of California Press, 1985.

Author Notes

Vivian Anette Lagesen is Professor of Science and Technology Studies at the Department of Interdisciplinary Studies of Culture, Norwegian University of Science and Technology (NTNU). She was trained as a sociologist. In her research, she has worked at the intersection of science and technology studies and gender studies. She has published extensively on women in higher software engineering education in different cultural contexts, including inclusion strategies and policy work to recruit women to ICT in a cross-national perspective. Presently, Professor Lagesen is engaged in a study of software engineering professionals in California, Malaysia, and Norway and in a study of how consulting engineers approach environmental problems.

Knut H. Sørensen is Professor of Science and Technology Studies at the Department of Interdisciplinary Studies of Culture, Norwegian University of Science and Technology (NTNU). He has published extensively on a wide variety of topics in science and technology studies, including engineering studies. His research interests include gender and technology, the practice of interdisciplinarity, sustainable energy and climate issues, and ICT professions.