

A Micro-Ethical View On Computing Practice

Peter Bittner

Humboldt-Universität zu Berlin
Dept. of Informatics
10099 Berlin DE
bittner@informatik.hu-berlin.de

Eva Hornecker

INTERACT Lab
Dept. of Informatics, University of Sussex
Falmer, Brighton BN1 9QH, UK
eva@ehornecker.de

ABSTRACT

Critical computing should include a concern for the everyday work practice of IT professionals. We explain our ‘micro-ethical view’, which shifts focus towards everyday work culture, seemingly mundane action, and describe what is gained thereby and by ‘preventive’ ethics. Four scenarios illustrate the situations relevant for our analysis. Reconstructing the history of ‘responsibility’ uncovers various concepts that provide fields of tension and systematize our analysis. We finish with related research questions relevant for this micro-ethical view that offer a broad field for research, as well as for professional action.

Keywords

Ethics, responsibility, everyday work practice, professional practice, work culture, moral improvisation, self-reflection

INTRODUCTION

‘Critical Computing’ means analyzing our own practices self-critically, asking where we are heading, which values drive us, and how we contribute to societal change. Undeniably there is a need to discuss new trends and fields like ubiquitous or embedded computing and bioinformatics, and to develop socially responsible visions, methods, and concepts. This discussion is at the forefront of our profession, focusing on innovative research in industry and academia. Yet only a minority of our colleagues enjoy the privilege of questioning being part of their work and deciding on general directions of technology development. Despite this they partake in computing culture, shaping the relation between the IT profession, ‘users’ and indirect stakeholders. Thus critical computing should include a serious concern for the mundane practice of computing, that is the everyday work practice of IT professionals.

Talking of a ‘micro-ethical view’ (analogous to micro-sociology or micro-analysis of interaction) we emphasize our focus on the scale of seemingly unremarkable and mundane everyday professional action. The ethical analysis aims at cultivating awareness for work cultures and reflection of thinking patterns and action patterns. This style of research addresses Kling’s [18] call for computer ethics to focus more on the everyday practice of computing and mundane, small design or action decisions. In a similar

vein, Hanseth and Monteiro [13] argue: “Much – arguably too much – attention is devoted to issues such as privacy, liability and copyrights” that are “high-flying in the sense that only a minority of systems developers experience these problems to any extent in their everyday working life”. Instead, research on computer ethics should take account of the messy, practical, day-to-day technical design decisions and their relations with political and power issues [9, 13, 17, 24]. Whistleblowing cases, similarly prominent in discourse on the social responsibility of engineers and computing specialists, are often used to sensitize students to professional responsibility. We argue that this approach conceals important aspects of professional practice (cp. [10, 20]). These cases usually have a long prehistory and form the tip of a larger iceberg. We believe that it is more fruitful to analyze how history led into disaster, and to detect the current the iceberg drifts within. Instead of a crisis ethics, we need ‘preventive’ ethics (cp. [20]).

We will now explain in more detail what can be gained by shifting focus from crisis situations towards everyday work culture and ‘preventive’ ethics. Four scenario sketches illustrate the type of situation relevant for our analysis. Then we delve deeper into the historical development of the term *responsibility*. By reconstructing responsibility and uncovering its many aspects, we harvest a range of concepts useful for the subsequent discussion of everyday professional responsibility. The concepts provide fields of tension, phrased as *In-Betweens*, that systematize our analysis. Here we return to the scenario sketches, using them as illustrations while at the same time analyzing them with our framework. We finish with a selection of research questions that could be pursued following the proposed micro-ethical view on computing.

REFLECTING EVERYDAY WORK PRACTICE

Shifting Focus from Crisis to Preventive Ethics

Lynch and Kline [20] analyzed for engineering how work cultures and seemingly routine decisions in the long run result in negative effects by accumulatively defining a trajectory for future actions. On its own every seemingly irrelevant decision seems rational. While no one can be accused, a culture of risk taking or of departmentalization of issues may be emerging. Work culture shapes categories, values, and patterns of thinking and acting. It defines what is normal and expected, making it difficult to deviate from this trajectory even if its harmfulness is detected. Thus moral problems have a long period of ‘incubation’, during which alternative development is possible. Employees are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AARHUS’05 8/21-8/25/05 Århus, Denmark

© 2005 ACM ISBN 1-59593-203-8/05/008...\$5.00

integrated in work culture, which defines the normal space of action. At the same time they re-enact and fill it with life. This provides options for evolution and change. Members can influence the evolving trajectory in early stages by questioning routine decisions or following different routines. This view is close to Wenger's [26] discussion of communities of practice, which highlights the "transformative potential" of participation, changing both the participant and the community. [2] addresses related issues when seeing the best chances for a cultural change of organizations in re-framing everyday life. This starts from the middle of an organization, subtly influencing it through "re-negotiation of meaning", incrementally, informally, and locally through individual personal interaction, potentially producing further ripples through the organization. Yet he sees this as an opportunity mostly for mid-level managers, requiring stamina, courage and willingness for sacrifice.

Thus it seems useful to provide students with the ability to observe and identify characteristics of work cultures. This would help them to reflect on their experiences, to choose which traits of a culture to adopt, which sub-communities to participate in, and to take an active role in transforming work cultures. For engineering education [20] recommend training the ability to recognize implicit assumptions and everyday ethical issues in badly structured problem areas. Education should furthermore foster the ability to develop creative solutions by identifying support resources (e.g. look for comrades and mentors). [10] also recommends a reflection of everyday work practice in education (for public mediation and facilitation), emphasizing the messiness of real stories as their strength, which reflect the tellers' search for value, depict the grey zones of everyday decision work on who to listen to, which arguments and issues to regard as significant, while forced to act timely.

On another level the proposed shift of focus applies to research on computing ethics. Instead of an ethics of crisis we should focus on 'preventive' ethics that identifies and fosters 'good' work cultures or practices, and creates a supportive environment for professional responsibility. We leave aside (competing) imperative moral norms and observe how actors actually 'juggle' situated moral action.

Shifting Focus to Everyday Work Scenarios

The work situations we refer to differ from whistleblowing scenarios in terms of the scope of effects and their ubiquity in everyday work experience. We give four short examples to illustrate the types of situation we find relevant for our analysis of everyday professional responsibility. These examples originate from experiences of acquaintances working in computing industries, answers to an informal e-mail questionnaire [16], or are abstracted from other reports [1, 23]. Even though they are abstracted, these are in essence true stories, capturing everyday life problems.

Scenario 1

A freshly graduated computing professional has started working for a middle-sized software company. She has the impression that the software that she contributes to is quite complicated and lacks usability. There are no specialists

within the company on these topics and no-one seems to regard usability as a relevant topic. Her education only included some basics of usability, but she feels she has some intuitive understanding of it. Other doubts she develops relate to the quality of a user manual. When she mentions her doubts, her colleagues and superiors tell her: "The users will quickly learn to handle this" and "In practice, it is no problem".

Scenario 2

A computing graduate starts as a programmer for business software. Soon he notices that the team that he is part of is aware of defects in the software, which optimizes purchase orders. In specific situations the software calculates higher purchases than necessary, in effect making the client spend more money than necessary. These defects are concealed, because the team lacks time to improve the software and a defect would have to be repaired without charge. It is implicit policy to do nothing as long as the client does not notice (this is very improbable due to the number-crunching nature of the software). Programmers and team managers argue that code always contains bugs and that the damage produced is only marginal.

Scenario 3

A software development division is mandated to develop a software application according to specification, which was provided by an IT consultancy. The contact at the client side is provided by its IT unit, which knows the formal workflows, but little details of the concrete use context. The client does not want the user department to be contacted, as it is busy with critical work. It is further argued that the consultancy already performed a task and business process analysis informing the specification. Yet the programmers think that the specification is incomplete and ambiguous. There are many open questions and alternative options regarding concrete implementation details, alternatives in workflows, and exception handling. Nevertheless, the end-user department is shielded off. As their company is under pressure to fulfill the contract, the software developers have to proceed according to specifications and to resolve open questions by imagination and replicating solutions from previous projects.

Scenario 4

An IT consultant works for a company that provides outsourcing facilities for computing centers but also consults clients on how to do this. As the company makes most of its profits from the provision of outsourcing, it is the consultants job to convince potential customers to outsource parts of their computing. Yet sometimes he has the impression that the client he is working with would be better off when keeping these resources and skilled people in-house. Sometimes he tries to hint customers at the disadvantages, but he cannot speak openly, needing to hide his 'disloyal' behavior from colleagues or supervisors.

RETHINKING RESPONSIBILITY

Why do we need a new structural approach? Traditional approaches of ethical analysis ignore practical everyday action, focusing on crisis. Furthermore they tend to use

‘responsibility’ in a monolithic way and focus on individual responsibility. If adhering to this viewpoint, individual responsibility often clashes with collective responsibility, creating the dilemma of either ascribing all responsibility to an individual or diffusing it. We found Bayertz’ [4] reconstruction of the historical formation and development of the concept *responsibility* useful as an alternative approach, lending systematic and principled structure to our discussion of everyday professional responsibility. The following presentation of Bayertz’ reconstruction is selective to aspects relevant here and short-cuts arguments primarily relevant for ethical theory (for a more extensive account see [4, 7]).

Uncovering the rich structure and history of the term responsibility, Bayertz [4] states five theses underlying his re-construction: (1) The notion of responsibility evolved as a specific solution within European society for the problem of attribution. (2) Attribution is not self-evident, but a result of social construction. (3) Different social conditions result in different constructions. (4) Structure and range of human action are crucial conditions of responsibility. (5) The ideas of human freedom and autonomy are constitutive for ‘responsibility as a specific type of attribution’.

Reconstructing Responsibility

Usually someone is regarded as being responsible if negative consequences can be attributed to his actions. This ‘classical’ model has several, often forgotten or unquestioned, but necessary prerequisites: *causality*, *individuality* and a *sharp distinction between humankind and nature*. Advanced moral thinking furthermore considers the conditions of action, asking whether attribution is ‘just(ified)’. This takes into account (besides causality) the actor’s *intentions* and her *possibility of foresight into consequences*. In Aristotle’s *Nicomachean Ethics* *voluntariness* is an important factor for evaluation.

The attribution of responsibility always involves a value judgment – attribution itself is only descriptive and without moral significance (note: accountability – in the sense of legible traces – eases attribution). Instead of an ontological view, where the event itself points onto the bearer of responsibility [22], one should therefore strengthen the constructional character of responsibility: (a) Human actions are not naturally given. ‘Actions’ originate primarily from post-hoc interpretation of events, interpreting them as manifestation of a subject responsible for its behavior. (b) Freedom of action (as prerequisite) is not empirically ascertainable, but only (normatively) assumed. (c) The focus on specific actions (with negative consequences) fulfils social means: penalty and directing human behavior into socially accepted tracks. (d) Certain actions or contexts are excluded from this scheme, e.g. cutting out rivals on the marketplace.

Although the essentials of responsibility have undergone a long process of development, the concept and term *responsibility* become relevant in ethical literature and public debates only by the second half of the 18th century. Bayertz [4] states that this should be understood as a

consequence of reflection on the fundamental changes in structure and type of human activity that result from industrialization. Two processes render attribution of negative consequences of actions to a person difficult: the advance of technology and the intensified division of labor. Suddenly damages with a social dimension become a focus of discussions, e.g. who is responsible for the pauperization of the working masses. Mechanisms and *possibilities of self-observation* (communication media, institutions) are an important resource for publicity. Discussion about the government’s duty to prevent such kinds of risks started in these times (e.g. laws concerning steam boilers).

Accidents in the domain of technological activity seem to occur independently of human action and will. But who is liable for damage resulting from *system failure*, if accidents cannot be causally attributed? Resolving the the originator/the-party-responsible-pays principle (causal attribution) with *strict liability* for risks and damages, regains access to (a) responsible person(s) – but we cannot distinguish between intended and unintended consequences any more. Liability is thus dependent on how society decides to deal with social problems resulting from technological risks.

Besides technification, industrialization initiated a progressive *division of labor* (with groups and institutions as subjects of action) inside business units and across regional, national, and international marketplaces. These associations and organizations, while necessary for the division of labor, can fail. Then attribution to an individual cannot be made or is hardly possible. The classical question “Who is responsible for damage?” is supplemented with “Who is obliged to fulfill certain tasks?” [4]. The subject of action is not responsible for negative consequences, but for a positive condition, guaranteeing smooth fulfillment of (allocated) tasks. A role-based type of responsibility gains in importance, which is oriented prospectively and defines care and custody. Yet *care responsibility*, closely connected with ‘duty’, is reflected on markedly less in ethical literature than ‘responsibility for consequences of action’. In any case one can only be held accountable for a certain (positively valued) condition, if one *possesses causal influence* on a given issue (possibility and ability) and is in a specific, normatively relevant relation to it (by higher mandate, self-commitment, or the value of the object) and thus obliged to fulfill the task [4]. This change in the concept of responsibility bears the risk of playing off against each other moral substantial responsibility, and functionally oriented accountability.

Essential Aspects

Using the rich structure and history of the term ‘responsibility’ as a resource, we learn to use essential notions like intention, voluntariness, autonomy, obligation, possibility of foresight, possibility of self-observation, causal influence, liability, (care) responsibility, and (constructed) attribution, and discuss how these relate to everyday computing practice. These concepts provide fields of tension (or forces) that systematize our analysis of IT practice and the everyday work scenarios given earlier.

This analysis is oriented by three questions: (a) What does the notion of insight mean for the process of attribution? (b) How does one move from insight into consequences, to discerning suitable and appropriate actions? (c) How is problem awareness related to public perception?

This approach aims at giving some structure to the messy entanglement of practice, pointing to general problems and issues like: workplace and computing culture, the ability for ethical judgment, the mutual supplementation of global and local actions, and collective and individual choices.

A Micro-View On Responsibility

“Which facts to take as significant, and which rules and responsibilities, goals and obligations, promises and understandings, to fulfill in what ways – these are inescapably moral matters that practitioners must face all the time” [10].

Ethical judgment is inescapable in (messy) practice. Yet awareness of the possibilities of choices is widely missing and not cultivated as part of computing culture. When presenting scenarios like ours to students, they often neither recognize that moral issues are involved nor see any choices. Forester [10] points to the value of everyday practical stories with “moral depth” for discussion and education. Their messiness shows the work of identifying issues and values, judging and improvising morally. Referencing Martha Nussbaum, he highlights the role of ethical judgment and moral improvisation.

Ethical judgment means “to fit action to circumstance, to see general principles in the light of contextual details (and vice versa)” [10]. It is the ability to move between taking the local context serious and applying principles, without putting one above the other. *Moral improvisation* is the ability to act situated while following principles, to recognize the richness of principles involved, and the richness of the situation, and to react to both. Forester emphasizes that in facilitation and public mediation, in order to resolve the underlying issues (not only symptoms) and to achieve consensus and justice, it is necessary to “not just get the facts, but the facts that matter” while “resisting the rush to interpretation”. *Improvisation* highlights the highly situated nature of acting and the need for sensitivity.

Similarly, [20] emphasize the value of everyday stories, telling about the small decisions forming a trajectory of action, about recognizing ethical issues in badly structured problem areas, and the influence of workplace culture. [13] argue “a increased awareness of ethics in systems design is inconceivable (or ... illusory) if not tightly coupled to the practical, day-to-day design decisions. Such decisions are seldom heroic; they tend to be ‘small’ and ‘concrete’.”

THE MESSINESS OF ETHICS IN EVERYDAY COMPUTING PRACTICE

We will now return to the three questions introduced earlier, rephrasing them as *In-Betweens*, which represent fields of tension provided by the concepts uncovered in the reconstruction of responsibility. These concepts are combined with ideas on local work culture and everyday action. Here we also return to the scenarios and illustrate

the *In-Betweens* with them, while at the same time analyzing them with the acquired concepts.

Insight – In-Between Foresight into Consequences of Action and the Problem of Attribution

Insight into consequences of action was previously identified as a major factor in attributing responsibility to a person. Yet complex systems and division of labor limit insight into consequences and the influence of actors. The systems that IT professionals devise are particularly complex and therefore necessitate a division of labor. For example the software developer in scenario 1 has a basic education in usability, so she cannot be sure and is unable to prove her point without involving an expert. Her colleagues probably know nothing about usability and therefore have no insight into the problem’s relevance. Their experience is that users are able to muddle through, but they are not aware of the costs and efforts required.

Standard practices of IT companies add to the limits on insight. Without time resources, limited budgets for analysis of use contexts and site testing, developers have little chance to foresee the effects of software employment. [23] describes how time pressure, non-adequate development processes and educational deficits result in unintended impediments. If separate teams work on different project phases (requirements analysis and coding), this interrupts communication and knowledge flow. Frequently, developers can only ask mediating persons about the usage context. Thus, it is almost impossible for them to develop an adequate conception of the actual use context. Scenario 3 is a typical example for this situation. Here we deal with a systemic problem, which leaves actors little chance of insight into consequences of design actions.

The problem of insight not only relates to IT professionals, but also to clients and users. Often they expect software to be a neutral tool and are not aware of the need for precise adaptation to their individual context. Furthermore they are usually not able to assess usability, security, stability, extensibility etc. of software, lacking the means and the ability to evaluate it. The client in scenario 2 exemplifies this, being unaware of the software miscalculating. Even if noticing it, it will be difficult proving that the software deliverer was grossly negligent. Customers and users also share responsibility by believing marketing and seeing IT-systems as a cost-factor only (cp. [5]).

Action – In-Between Possibilities to Take Action and Their Suitabilities

Insight into consequences of action neither implies the existence of alternative paths of action nor the feasibility and reasonableness of alternatives. Freedom and autonomy are always limited. Economy and competition often prevent or punish responsible action [5, 25]. Division of labor reduces the sphere of influence of individuals. Persons of different status have different scopes of influence and will be granted different degrees of deviation. Organizations often minimize the field of recognized ethical issues, discouraging discussion and considering ethics as

“private”, thus generating *ethical closure* – whoever raises moral issues is seen as a trouble-maker and unreliable [2].

Before attributing responsibility we thus need to ask: Is there an alternative path? Is it feasible or reasonable? What punishment can be expected? Furthermore we should ask what is the balance between the effects of alternative action and the punishment, and how high the chances are of having an effect at all. Nor can we demand everybody to be a hero nor might heroic action be the cleverest means (cp. [20]). In an e-mail inquiry workload and time pressure were mentioned most often as hindrance to put good intentions (usability, privacy) into practice [16]. Various conflicts of interest result, regarding customers and users, the personal environment (colleagues and subordinates), one’s family and the company’s survival (deadlines, fines). Furthermore loyalty to the employer is a contractual obligation. Here we join with the critique by [9] of many professional approaches to ethics that ask the individual developer to “undertake a rather impressive personal responsibility”, without offering organized support in case of conflict.

The software developer in scenario 1 is new in the company. She may decide to trust her co-workers’ experience. If she voices her opinion, consequences depend on the openness to new ideas. But the company seems to have established a culture of indifference to user needs, and even of resistance against suggestions for improvements. The young professional’s doubts are shrugged off without further review, revealing the low standing of users within the work culture of this company. Arguing against such deeply ingrained beliefs is tricky. The young developer’s options may increase once she is recognized as an accomplished worker and has established personal relations. Then she may be able to convince colleagues and change the work culture from below. She may also attempt to convince superiors to invite usability experts or trainers.

The development team in scenario 2 knows very well about the flaws of its product. It only acts once the client becomes aware of these. Within a service relation this is a clear breach (or abuse) of trust. Revealing flaws would create costs and damage the company’s reputation. If the damage is large enough it is justified to break discretion. But assessment of its gravity is difficult, re-raising the question of insight. Breaking discretion means risking one’s job or facing (informal) punishment. The implicitness of the above policy reveals the software team’s acceptance of damage to the client. This hints at ethical closure [2], a culture of unaccountability, even of disloyalty and exploitation of users. A change of company culture seems hard to achieve. Therefore the graduate should ask himself if he wants to become a member of this community at all.

The development team in scenario 3, which cannot contact its end-users, has little chances to produce professional results. This needs to be articulated to the client, as the project may disrupt work practices or fail.

The IT consultant in scenario 4 has insight into the consequences of outsourcing for his client. Caught in a conflict between employers’ and customers’ needs, he is

forced to knowingly act against the customers’ interests. This is more than a micro-ethical problem, rather a company-wide, and probably long-term cultivated contradiction at the highest level of the consultancy’s organization and self-conception. The consultant has little latitude and is caught in the general dual motive of all professions: “to provide service and to use their knowledge for economic gain” [19]. Trying to balance, he gives slight hints, but needs to take care to keep his job.

Attribution as Social Construction – In-Between Problem Awareness and Public Visibility

Whenever problem awareness for certain risks increases, the perceived pressure to act and to accept consequences rises. We see little public debate relating to IT, its risks, and the responsibility of its protagonists. Society seems strangely unwilling to attribute responsibility to IT, while the IT profession itself seems quite happy with being unaccountable. Present law allows IT companies, different from engineering and construction business, to enforce limited liability for product deficiencies or defects – a dominant practice with shrink-wrapped software. The company in scenario 2 might have included such phrases in license agreements. The historical reconstruction of responsibility told us that strict liability was a result of political/social processes of negotiating responsibilities and duties and thus *could be* extended to software.

The limited insight of customers and users into quality of software contributes to limited problem awareness. Users have been almost ‘conditioned’ to accept software flaws and bad usability like the laws of nature, not expecting any better. Furthermore on the client side there is often a short-sighted perspective and no holistic evaluation of costs/benefits that includes follow-up costs related to maintenance, workers’ health etc. [2, 5, 25]. If the clients of the company depicted in scenario 1 were concerned about usability, the company would not succeed with its attitude. This requires knowledgeable customers and competitors meeting such demands. The outsourcing client in scenario 4 should be careful if an employed consultant suggests solutions offered by his company, and consult another opinion. There seems an (intended) discrepancy between customers’ and consultants’ perceptions of the consultancy’s mission and objectives. All of this calls for public education and a more self-reflective stance for the computing profession.

INSPIRATION FOR FUTURE RESEARCH

This section introduces a selection of research questions relevant to our research perspective. These represent an initial set of issues, far from complete, quickly sketched, giving a flavor of the issues that come into the field of view and the approaches to be taken when following our micro-ethical perspective. Many require a combination of empirical studies and conceptual analysis. Any of these topics presents complexity and challenge for one or more research projects. A small study done by students under our supervision illustrates this type of approach. Finally we give an overview of appropriate methodical approaches for a micro-ethical approach with empirical studies.

Research Questions

On Computing Culture, Self-Images, Images of the User and Communication with Users

Part of a reflection of work cultures in informatics could be reflecting the dispositions and attitudes of computing professionals towards clients and users and their style of interacting with them. This refers to several levels of behavior and relations, e.g. types of business relations and strategies in interacting with clients, person-to-person relations, or perceptions and images held of users and clients. These behaviors and strategies are furthermore related to the self-images of computing professionals and the role they see themselves in or choose to take.

We assume that images of the 'user' implicitly affect design decisions and the behavior of software developers. 'Funny' terms, such as the 'worst case user' (in German: "the most stupid user to be assumed") can quickly turn into a habit within a team. The disregard and disdain reflected in this term may shape the general attitude towards end users. The user here is seen as a problem (cp. [3]) that needs to be repaired or silenced. Use of such terms makes it easier to ignore complaints about usability or system failures and less likely to invest in understanding user needs or user participation. It may also lead to a one-sided focus on simple-to-use, non-demanding systems, which neglect experienced users' skills and don't empower them to become power-users by providing paths for learning.

Thus 'images of the user' would be an interesting research topic for interviews with computing professionals. More detailed questions would be what affects these images, e.g. factors in the workplace (workload, interaction structures), and whether they are related to actual experiences. And how are these images handed over to newcomers in a company, division or team? A further issue would inquire how images of the user affect decisions in software design. We might also turn our question upside-down and inquire how clients and users perceive computing professionals and how these images influence the relations they enter.

An analysis of work cultures also includes the relations between computing professionals, their styles of working and interacting, and their work conditions. Research questions might include: What are psycho-social characteristics and structures of IT-departments or teams, and how do these relate to work patterns and relations towards others (cp. [2])? What are the effects of workload and structures of interaction? How do different methods in Software Engineering affect internal communication? We might also inquire how characteristics of a culture, such as a high willingness to take risks (creating a tendency towards hazardous decisions without adequate risk-analysis) evolve. More generally one can ask whether there is a relation between quality criteria and cultures. Furthermore it might be revealing to investigate which quality criteria software professionals adhere to privately and which are part of their workplace culture.

There are some studies to turn to for inspiration on how to conduct this type of research. [14, 15] compared small IT-

companies in Eastern and Western Germany and the US, focusing on (innovation) strategies as a part of company cultures as well as on working styles, using participant observation, semi-structured interviews and document analysis. [6] conducted biographical interviews with IT-professionals, investigating women's careers. O'Neill [21] analyzed user-developer interaction in task analysis and system design through interaction analysis of videos. He focused on how developers handled users' contributions, differentiating between mere user presence and active, effective participation. Only the latter results in contributions taken account of in task models and software design. It was found that most usability problems were a result of ignored user contributions, of blindly following an unstructured, ad-hoc user comment, or of features introduced by programmers without involving users.

On Requirements Engineering as a Field of Ethical Issues

The early phases of software development can be seen as inherently full of ethical aspects. As they decide on the shape of future technologies and are known to be crucial for the success of projects, this topic seems particularly fruitful. Goguen [11] describes the requirements engineering process as inherently social, with some of the most vexing difficulties being social, political, and cultural. Who is supposed to benefit from the proposed system and who will actually use it often has ethical implications [9, 13, 24]. Systems demanded by management often ease administration, but add pressure to users work and increase surveillance. It is not uncommon that the workers interviewed are not to be told that the system will eliminate their jobs. [13, 24] sketch case studies illuminating the intertwining of technical decisions with political/social processes and their influence on power relations. The requirement process often reveals conflicts within clients' organizations and requires a discussion of value systems [11] (cp. [10] describing a need for moral imagination and improvisation). Requirements engineering may furthermore conflict with or be exploited for hidden agendas.

O'Neill [21] points out that participatory design research (PD) focuses on cooperative design and prototyping, and calls for the development of methods on participatory analysis. Yet, as has been noted in PD literature, participatory analysis will not alleviate all of the before mentioned ethical issues, as involved users may have hidden or conflicting agendas and interests [9, 17, 24].

As directions for research we can imagine (following Forester's [10] suggestion) collecting stories of requirements engineering work and analyzing them. Case studies as given by [13, 17, 24] could raise sensibility for the social consequences of technological decisions. Many questions suggested for computing culture, self-images and images of the user could be pursued for this subfield of the computing profession, as we cannot assume different types of activities within computing, taking place within specific conditions, to share the same characteristics.

On Structural Aspects Constraining Responsibility

For the discipline as well as for preparing students for the workplace it would be useful to identify structural problems of common practice in IT-industry, which constrain the individuals' options for insight and available action spaces. In addition to fostering students' ability to develop creative solutions to situations of ethical concern [20], we need to develop their sensitivity for these constraints so they do not run into walls or offend co-workers and superiors. Moreover, identifying these structural constraints might enable the discipline to collectively take influence and to engage at multiple levels [9].

It seems that some constraints result from systematic problems due to the structure of software development. The workload in industry and missing knowledge about usage contexts constrain insight into the effects of design decisions. It is still common to subdivide work into separate tasks and teams for requirements analysis and software development, leaving developers with little information on, or contact to the use context [11, 23]. Often managers both in user and developer organizations are reluctant to allow contact between users and developers [3, 12, 21, 23]. This is sometimes motivated by fear of competitors, but often is simply not regarded important, is intended to keep developers "happy, not to waste their time" [3], or users are high-paid and indispensable at work and hence inaccessible. Division of labor and high workloads limit the action space even if an individual is able to discern negative effects [7, 16]. The requirement to remain loyal to employers in combination with the responsibility for on-time completion of projects (documented in contractual secrecy and loyalty statements) often competes with what's best for the client.

Many organizational obstacles exist for user-centered design and usability [3]: e.g. ignorance, it not being a competitive factor, marketing the interface with customers, prejudices, output-oriented reward systems, short product life cycles, small budgets... Some of these problems are being addressed by attempts to include usability and user-centered design in software development models or by collecting evidence for the business value of usability.

It is obvious that different methods or models of software development (waterfall model, spiral model, extreme programming, iterative design etc.) provide specific structural constraints. E.g. iterative design enables early insight into design consequences and permits changes due to evolving insights. Thus one could systematically analyze the action spaces and options for insight that certain models offer. Besides of software development methods there are specific types of structural relations between clients, users and developers. Grudin [12] analyzes the interaction and interrelation of involved persons (or organizations) and their communication for contract, product, and in-house/custom development, depicting graphically who is known from the outset and which options for interaction exist. These are essential aspects of work structures and conditions. This method of structural analysis would lend itself to case studies, but should be extended with indirect

stakeholders. Furthermore it needs to be adapted and extended to new contractual relations in software development that have evolved since.

On Computing as Profession

Usually our understanding of profession is based on a trait approach. This approach has been criticized both from within informatics and from within the sociology of professions. In [8] we argued how computing practice might be better understood from a professionalization viewpoint. A multidimensional view with sensitivity for the various perspectives (from the sociology of profession tradition) provides new impetus for discourse, raising questions like: Which structures regulate the interaction between computer professionals, clients and society? Which mechanisms of interaction dominate? How do professionals act in conflicts between technical problem-solving and vicarious crisis-handling? What is our approach towards the paradoxical, the fragile and erroneous in professional action? The debate on professionalization in pedagogics provides further impulses. We learn that computer professionals should provide situated arrangements that facilitate (self)empowerment or advance it. The client must be able to deal productively with the (future) system and the cultural changes it will entail - otherwise the computer professional's task as an 'advocate' is not feasible. Acting professionally we structure and accompany the process in which the clients articulate problems and preconditions of their work processes.

Further Suggestions

We'd hope for a collective effort to foster a *culture of storytelling* within computing, sharing stories "of moral depth" [10] with other professionals and with students. Most computing professionals we talked to feel that - not encountering big moral dilemmas - they do not have significant stories to tell, disregarding their everyday struggles as irrelevant. Yet there are also legal issues for sharing stories, as these often counter contractual loyalty and discretion. Thus there is a need for 'safe places' for discussions and for mechanisms of anonymizing stories. Further a shared effort to discern which issues discretion applies to and where it is used by organizations as a rhetoric device to silence employees is needed, and an investigation of what else might discourage storytelling. Places and occasions for shared storytelling and story discussion might act as facilities of imagination, reflection and exploration, which for communities of practice [26] seem to be a major element enabling learning. Shared effort is also needed to help in case of conflicts of responsibilities. What role can professional communities play in these cases (e.g. as mediator) and what can company self-regulation do? Does self-regulation need to be supported by laws?

Education of future computing professionals provides a field for experimenting with didactic and conceptual approaches and evaluating these. Aims should be fostering a sensibility for ethical issues and acquiring skills of ethical judgment and moral imagination.

Some issues discussed in this section point to the need for *public awareness*. If clients feel contact between developers and users to be unnecessary, are unaware of the profound effects of new business software on their work processes, and do not buy software based on usability, there is no reason for software companies to re-think their strategies. Efforts for user-centeredness are simply not rewarded on the market, as clients often choose based on price. Furthermore, end-users tend to accept software bugs like a law of nature. Unlike with other professional products, license agreements can declare the manufacturer to be not accountable for any damage due to the software (faulty calculations by a software for private income tax declarations recently led to several persons being charged). Society thus doesn't attribute responsibility to computing yet. Thus how can we heighten public awareness, make customers aware of software influences on their organization and of long-term effects, and how can we educate users to apply a wider range of quality criteria to software? A question both for ethical theory and law concerns what strict liability for risks means in the field of system development.

An Example Study – Interviews With IT Consultants

In a one-term project supervised by one of the authors, students interviewed IT consultants about their job motivation, self-images and conceptions. The study aimed to find out what ethical criteria IT consultants find relevant for their work, what kinds of ethical conflicts they encounter and about their relations with customers. Students interviewed five consultants working self-employed, in small companies and in large IT corporations [1], recording and transcribing the conversations.

A key result for students was a distinction between independent and 'dependent' consultants. The former are not related (contractually or financially) to the company that produces the 'solutions' and therefore can afford to focus on counseling with the clients' welfare as top priority. 'Dependent' consultants essentially sell the recommended product, are obliged to act in the interest of their employer, and therefore honestly should be termed salespeople. Independent consultants tended to say that there is no such thing as a typical project: "Each client must be viewed individually." Dependent consultants were more inclined to match projects and client demands with existent project patterns or 'norm solutions' to lower efforts and costs, and in this course often modified client requests.

Distinguishing himself from these salespeople, an independent IT consultant describes his approach: "The client pays me. My task is to prevent that someone talks him into buying something he does not need and that is too expensive. (...) But this independence has a price as well" (quotes translated from German). He compares dependent consultants with car dealers and points out that it is the clients fault if they are taken in and do not seek a second opinion, eschewing the cost of independent consultancies. An interviewed dependent consultant may serve as an example here, admitting that her division sometimes includes products where sales need boosting into a project

and that expensive products would be sold purposely if the clients' budget allows it. Here we see a good example of the systemic distribution of responsibility, with clients co-producing the very problem they might later complain about.

Apart from the (obvious) distinction between dependent and independent consultants that results from contractual duties, the interviews uncover differences in consultants' attitudes towards clients and relations with them. These influence their strategies and communication styles. Two of the interviewed consultants exemplify this.

One consultant (IT project manager for hardware and outsourcing) describes a relation based on partnership with clients. He believes an IT-consultant should serve the clients' wishes and provide high quality work. His goal is finding an efficient solution that is feasible with reasonable effort (for his company, which offers 'norm solutions' for adaptation) and solves the clients' requirements. Problems occur if clients have "fancy wishes" and are not aware of the costs. This requires tact, sensitivity and persuasion. It should be noted that independent consultants as well might need to persuade clients to look for affordable alternatives and downgrade expectations. Asked if he stipulates the project direction, he denies this and describes it as a "jointly developed solution". His task is to "tell the client how to realize his ideas and to transform these into concrete goals", repeating that this means the solution is (jointly) developed. The main goal of a salesman being to satisfy clients, one would not try to convince a client to buy a solution inferior to the desired one. He admits that once in a while unneeded small things are part of the sold package. Usually this would happen if in planning or meetings there were misunderstandings or issues the client did not address.

A young and very successful IT consultant working for an international company provides a contrast. Asked if her company always finds adequate solutions, she first answers "in most cases". Otherwise, they would "offer something oriented roughly/approximately by the clients' wishes." If this is impossible, they would "try to change the requirements so we can offer something." Asked whether the client or the consultancy stipulates the direction of projects, she answers: "Usually clients do not know what they want or do only have a very coarse idea. Then my task is to help them and show the options." The students probe further: "Isn't it a problem if you need to tell the client what he wants?" Her answer now reveals her attitude: "If you do this well, the client will believe that he came up with it and thinks it was his idea. Then he is convinced he is getting what he wanted." Her view of the client is very different from the other consultant who emphasized partnership, joint development of solutions and negotiation of project direction. Here the client is viewed as incapable and clueless. Persuasion is not coupled with tact, sensitivity and respect, but employs manipulative tactics to stipulate project direction and change requirements. Later she admits that her division sometimes includes products that need higher sales numbers into a project. Her matter-of-course style of talking along with the reference to standard

practices of product selection give rise to the assumption that her attitude and approach are consistent with the work culture of her division and perhaps of the entire company.

Few of the interviewed report having encountered ethical problems, e.g. being asked to identify 20% of employees for job-cutting, essentially ‘washing’ the client of this unpleasant responsibility, and not being allowed to suggest alternative solutions. The interviews did not explicitly touch competing responsibilities besides mediating between employers and clients’ interests, not referring to other societal responsibilities or end-users.

Even though this were only five interviews of fifteen minutes to at most half an hour length, they reveal various aspects of IT-consultancy, different self-conceptions and evolving relations with clients. We can relate the interviews with many of the issues discussed so-far, e.g. work-culture, the individuals’ scope of influence, and the co-production of problems by users. For a real-sized research project we would recommend (besides more and longer interviews) also interviewing clients on their experiences with IT-consultants and their views on the distribution of responsibility in this specific professional relation.

Methodological approaches

The wide range of research issues sketched here requires a similarly wide range of methodological approaches. As many issues require investigation of actual practice, empirical methods play a large role, combined with conceptual work in ethics, sociology, etc. Such methods include interviews (with specific focus and methodologies coming e.g. from sociology, pedagogy, psychology, political sciences), and ethnographic methods relying on observation. Stressing the role of interaction processes, local work context, social practices [2] (and obviously focusing on kind of workplace communities of practice [26]) similarly recommends ethnographic approaches. Case studies as given in [5, 13, 17, 24] provide material for analysis, connecting theoretic work, structural approaches and empirical studies.

Some issues and questions clearly call for interventionist and action research strategies, e.g. when investigating how to raise public awareness on specific issues and raising student awareness. On a more conceptual level, research e.g. from sociology on professions and on organizational culture [2] could be integrated into the method vendor’s tray and inform the interpretation of studies. The concept of communities of practice (CoP) [26] seems highly related to ‘work cultures’ with software development teams forming a CoP, company, clients and users forming a connected community of CoPs. This might provide a useful analytical vocabulary/method informing analysis and generation of suggestions for a more reflective computing practice. Furthermore ethical theory should be reviewed for theories applicable to our research questions, e.g. for theories on everyday ethics and relational responsibility, and collaboration with philosophers established who are interested in empirical investigation.

CONCLUSIONS

We have argued for the relevance of a micro-ethical view on computing practice, focusing on everyday action and work culture in the IT or computing profession, and shifting towards ‘preventive’ (or situated) ethics. We illustrated the situations and problems that we focus on with four example scenarios, which address e.g. issues of limited awareness of usability, disregard of users, contractual obligations, conflicts of interest, and dysfunctional organization of software development. We then presented Bayertz’ [4] reconstruction of the historical formation of the concept *responsibility*. This reconstruction yields a range of concepts that lend a systematic and principled structure to our discussion of everyday professional responsibility. It furthermore evades playing out individual against collective responsibility or classical against role-related responsibility and liability. Using the rich structure and history of the term ‘responsibility’ as a resource, we learned to use notions like intention, voluntariness, autonomy, obligation, possibility of foresight, possibility of self-observation, causal influence, liability, (care) responsibility and (constructed) attribution. Our analysis was augmented by analysis and concepts from [10, 20], which point to the value of everyday stories and their messiness, to ethical judgment and moral imagination and to the influence of work cultures.

The extracted concepts were then rephrased as In-Betweens or fields of tension, structuring and systematizing our discussion of everyday ethical action within computing practice. These In-Betweens concern insight, availability of alternative action and its suitability, and the influence of public awareness. Here we returned to the scenarios presented earlier. We found standard patterns of computing practice to severely limit insight into consequences of action as well as to limit the space of alternative actions. In addition we feel that there is too little public awareness of ethical issues concerning computing and therefore little attribution of responsibility by society towards computing. Finally we suggest a range of research issues that follow from our micro-ethical approach, and which feel fruitful for increasing the reflectiveness of our profession. We illustrate this type of research with a small study from students supervised by us, who interviewed IT-consultants.

Empirical studies need to feed back into conceptual approaches and theories, yielding those “stories of depth” that [10] praises for their fruitfulness for education and reflective professional practice. The ideal would be inter- and trans-disciplinary cooperation of computer scientists, sociologists, anthropologists and philosophers. International collaboration would be required to access previous work, which is distributed over country-specific intellectual schools (typical e.g. for sociology and pedagogic) and not published in English. How to establish collaboration on these questions with practitioners and industry poses a particularly tricky issue.

A methodological dilemma results from the fact that the required methods are not part of standard computing education. Yet computing should not leave the study of its

own practice to other disciplines and thereby declare this issue to be in its own margins – other disciplines regard self-analysis as an elementary part of their disciplines (e.g. pedagogy, health care). Thus interdisciplinary cooperation is required, along with the appropriation of empirical methods by computer scientists, as already happens in social informatics, HCI and Participatory Design.

Our approach differs from usual approaches to ethics by looking at everyday practical action and the workplace culture which it is shaped by and that it shapes simultaneously. Some of the issues addressed (e.g. how technical decisions affect power relations and are thus ethical issues and inherently political) have been discussed for research projects, mainly in PD (e.g. [5, 17, 24]) – these stories can serve as examples for education and reflection by practitioners – but our concern is broader, towards the overall profession and its ability to serve its clients as well as the society in a respectable way. Similarly to what [9] suggests, our approach demands to intertwine the local and global, as we cannot ask individuals to take on personal responsibility and then leave them without support – needed is a multi-level approach spanning the workplace, the organizational and the legislative level.

ACKNOWLEDGMENTS

We thank Daniel Görlich for discussion of the scenarios and proofreading, Marjo Rauhala for discussions on ‘micro’-ethics, John Halloran for proofreading, and the students [1] who carried out the interview study on such a delicate and tricky topic in such an impressive way.

REFERENCES

1. Aigner, S. Knoll, J., Mauer, A., Rieschl, W., Hornecker, E. Verantwortung in der IT-Beratung: “Wenn man das gut macht, dann glaubt der Kunde, er sei selbst drauf gekommen”. *FIfF-Kommunikation* 22(1), 2005, 12-18.
2. Alvesson, M. *Understanding Organizational Culture*. Sage Publications, London, 2002.
3. Anderson, R. *Addressing Organizational Obstacles to, and Achieving (Greater) Business Benefits from “User-Centered” Design ...* 2002, <http://www.well.com/user/riander/obstacles> (read July 2002).
4. Bayertz, K. Eine kurze Geschichte der Herkunft der Verantwortung. in Bayertz, K. (ed.) *Verantwortung: Prinzip oder Problem?* WBG, Darmstadt, 1995, 3-71.
5. Bälter, O., and Lantz, A. A Case Study where PD Would have Helped – Or Maybe Not? in *Proc. of PDC’00*, CPSR, Palo Alto, 2000, 253-257.
6. Birbaumer, A., and Wagner, I. Women Professionals in Innovative Companies. in *Proceedings of Gender Work and Organization Conference*, Keele, 2003.
7. Bittner, P., and Hornecker, E. Responsibility and the Work of IT-Professionals. in Brunstein, Berleur (eds.) *Human Choice and Computers*. Kluwer, 2002, 171-181.
8. Bittner, P. On Professional Informatical Action. *Int. Review of Information Ethics* 1(2), 2004, e-journal available at http://www.i-r-i-e.net/issue_2.htm.
9. Bjerknes, G., and Bratteteig, T. User Participation and Democracy: A Discussion of Scandinavian Research on System Development. *Scandinavian Journal of Information Systems* 7(1), 1998, 73-98.
10. Forester, J.F. *The Deliberative Practitioner. Encouraging Participatory Planning Processes*. MIT Press, Cambridge/London, 1999.
11. Goguen, J.A. Social Issues in Requirements Engineering. in *Proc. of Requirements Engineering’93*, IEEE, 1993, 194-195.
12. Grudin, J. Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer* 24(4), 1991, 59-69.
13. Hanseth, O., and Monteiro E. Ethics versus Politics in System Development. in *Proc. of IFIP WG 9.1 Workshop Ethics and Systems Design*, 1994, 51-56.
14. Henninger, A., and Sieber, A. More User Participation by New Media? Media Use in Small Software Firms. in *Proc. of PDC’00*, CPSR, Palo Alto, 2000, 273-277.
15. Henninger, A. Lokal Heroes im globalen Informationsraum. in Hirschfelder, Huber (eds.) *Die Virtualisierung der Arbeit*. Campus, Frankfurt, 2004, 89-113.
16. Hornecker, E., and Bittner, P. Vom kritischen Verhältnis zur Berufspraxis in der Informatik – Ergebnisse einer Befragung. *FIfF-Kommunikation* 17(1), 2000, 33-39.
17. Hyysalo, S., and Lehenkari, J. Contextualising Power in a Collaborative Design Project. in *Proc. of PDC’02*, CPSR, Palo Alto, 2002, 93-102.
18. Kling, R. Beyond Outlaws, Hackers and Pirates. in Kling, R. (ed.) *Computerization and Controversy*, 2nd ed., Academic Press, San Diego, 1996, 848-869.
19. Krause, E. *Death of the Guilds: Professions, States and the Advance of Capitalism, 1930 to the Present*. Yale University Press, London/New Haven CT, 1996, ix.
20. Lynch, W.T., and Kline, R. Engineering Practice and Engineering Ethics. *Science, Technology and Human Values* 25(2), 2000, 195-225.
21. O’Neill, E. *User-Developer Cooperation in Software Development*. Springer, London, 2001.
22. Picht, G. Der Begriff der Verantwortung. in Picht, G. (ed.) *Wahrheit, Vernunft, Verantwortung – Philosophische Studien*, Klett, Stuttgart, 1969, 318-342.
23. Twisselmann, U. Informatik und Arbeitsumgebungen. *FIfF-Kommunikation* 17(1), 2000, 28-31.
24. Wagner, I. A Web of Fuzzy Problems: Confronting the Ethical Issues. *Communications of the ACM* 36(4), 1993, 94-101.
25. Webb, J., Rackley, L., and Betts, J. Pressures to Behave Unprofessionally. in Myers, Hall, Pitts (eds.) *The Responsible Software Engineer*. Springer 1997, 265-272
26. Wenger, E. *Communities of Practice*. Cambridge University Press, Cambridge, 1999.