

Responsibility and the Work of IT-Professionals

From Academia to Practice

Peter Bittner¹, Eva Hornecker²

¹Humboldt-University at Berlin, Department of Informatics, Berlin, Germany

²University of Bremen, Research Centre artec, Bremen, Germany

bittner@informatik.hu-berlin.de, eva@artec.uni-bremen.de

Abstract: Unlike most computer-ethics discussions on issues like hacking, software piracy, or “big” ethical issues, we want to discuss the routine work of IT-specialists and the context in which it is situated. The work of creating and facilitating use of IT-systems offers many routine opportunities to “do the right thing” and many contextual factors hindering this. Thus our analysis starts looking at everyday practice, and *re-constructing* “responsibility”. With this approach we hope to expose new ethical issues. 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, causal influence, (care) responsibility, and attribution and discuss how these relate to the practice of IT practitioners. This discussion aims to give some structure to the messy entanglement of practice, pointing to general problems and issues like: workplace and computing culture, the ability for ethical judgement, showing how global and local actions, collective and individual choices supplement each other.

Key words: social responsibility, accountability, computing & workplace culture, risks, preventive ethics, problem awareness, moral judgement and improvisation

Talking to practitioners, we see how ethical issues are paramount, although they tend to face up in the small. But for practitioners the usual discussion on “big” ethical issues or topics like hacking and software piracy is of little help – they have to decide quickly, constrained by demands and restrictions. We discuss similar aspects like [13], but with a different conceptual and methodical approach - by *re-constructing* “responsibility”. Reflecting on the evolution of the concept “responsibility”, which is coupled with the advance of technology we avoid a computing-centred perspective, which focuses on consequences of specific technologies (cp. [16]).

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35609-9_29](https://doi.org/10.1007/978-0-387-35609-9_29)

1. ON THE GENEALOGY OF RESPONSIBILITY

The concept of responsibility is often used very undifferentiated in the literature on computers, society, and responsibility; questions of ethics and morale are mingled. First we will uncover – according to Bayertz [1] - the rich structure and history of the term responsibility. Bayertz [1:4-5] states the following theses underlying his *re-construction*: (a) the notion of responsibility evolved as a specific solution within European society for the problem of attribution. (b) Attribution is not self-evident, but a result of social “construction”. (c) Different social conditions necessarily result in different constructions. (d) Central elements for the conditions of responsibility are structure and range of human action. (e) The idea of human freedom and autonomy is constitutive for “attribution as responsibility”.

1.1 Responsibility - the classical model

When the cause of an event is a human being or it can be traced back to human action, responsibility gets a topic. Thus, the subject (or person in Kant’s terminology) of the action is responsible, “bad” consequences, which causally follow from his/her actions are accounted to him/her.

Often forgotten and unquestioned, but necessary prerequisites for this “classical” model are: *causality*, *individuality* and *a sharp distinction between humankind and nature*. According to Kelsen [11], causality⁹ had to emancipate itself from retaliation. Advanced moral thinking also considers the conditions of action (the inner view of the “culprit”). The question of “just(ified)” attribution takes into account (besides causality) the *intentions* of the actor and his/her *possibility of foresight into consequences*. In Aristotle’s *Nicomachean Ethics* *voluntariness* (as personal authoring of the actor) is an important factor for evaluating a situation. Nevertheless “voluntary” and “involuntary” do not mark distinct classes of actions. The given discretionary powers of decision present a heavy argument for the constructional character of responsibility.

The attribution of responsibility always involves a value judgement – as attribution itself is only descriptive and without moral significance. Here we deal with implicit and explicit (but not generally accepted) norms.

⁹ Causality is not a “natural” way of thinking. Blood-revenge between families shows that the notion of attribution to an individual subject can not be taken for granted as well. Responsibility also results from the specific position of human causality (in the middle ages animals were sentenced to death for ‘murder’).

“Responsibility” can be seen as a network of the subject of responsibility, the object of responsibility and a system of value judgements [1]. This network can be further refined according to Lenk & Maring [14:229].

Instead of taking an ontological view of responsibility, where the event itself points onto the bearer of responsibility (see [21]), one should strengthen the constructional character of responsibility. Why construction?

(a) Human activity is not naturally given. “Actions” originate primarily from post-hoc interpretation of events, interpreting them as manifestation of a subject, which is in principle responsible for its behaviour. (b) Freedom of action (as a prerequisite) is not empirically ascertainable, it is (normatively) assumed. (c) The focus on specific actions (those with negative consequences) fulfils certain social means and goals: to call the culprit into account. Punishment attempts to direct human behaviour into socially accepted tracks. (d) It is possible to exclude certain actions: cutting out rivals on the marketplace is not judged offensive.

1.2 Responsibility - term formation and modern changes

Although these essentials of responsibility have undergone a long process of development, concept and term “responsibility” do get relevant only as late as the second half of the 18th century in ethical literature (Lévy-Bruhl [15], Nietzsche [18]) and public debates (e.g. about the explosion of steam boilers). Bayertz [1:24f.] states that the rise of the concept “responsibility” as a central ethical category should be understood as a consequence of reflection on the fundamental changes in structure and type of human activity, resulting from industrialisation. Attribution of negative consequences of actions to someone is rendered difficult by two processes: (1) the advance of technology, (2) the intensified division of labour.

As a new phenomenon, damages with a social dimension are focus of discussions, e.g. about who is to be held responsible for the pauperisation of the working masses. Mechanisms and possibilities of self-observation (communication media, institutions) are an important resource for publicity. Discussion about the duty of government to prevent these kinds of risks started about that time (e.g. laws concerning steam boilers). The government’s responsibility interferes with private business, when security and welfare of the (entire) public is affected (see [3:332]).

Accidents in the domain of technological activity seem to occur independently of human action and will. Technology gains in autonomy and withdraws itself from human control – in particular wherever mediating elements (tools, machines, technical systems) play a major role. But who is liable for damage resulting from *system failure*, if certain accidents cannot

be causally attributed? When we resolve the originator/the-party-responsible-pays principle (causal attribution) with *strict liability* (for risks and damages), we regain access to (a) responsible person(s) – but we cannot distinguish any more between intended and unintended consequences. Liability is thus dependent on how society decides to deal with social problems resulting from technological risks.

Besides of technification industrialisation initiated a progressive *division of labour* (with groups and institutions becoming subjects of action) not only inside of business units but as well on the level of regional, national, and international market places. Partaking individuals' contribution to the results of production processes drops. These co-operations/organisations (necessary for the division of labour) can fail. Attribution to an individual is not or hardly possible in these cases. The classical question "Who is responsible for damage?" is now supplemented with the question "Who is obliged to fulfil certain tasks?" (see [1:32]) The subject of action is not responsible for negative consequences, but for a positive condition guaranteeing smooth fulfilment of (clear-cut allocated) tasks. A role-based type of responsibility gains importance, which is oriented prospectively and defines care and custody. *Care responsibility*, closely connected to "duty", is reflected perceptibly less in ethical literature than "responsibility for consequences of action". In any case one can only be accountable for a certain (normatively positive valued) condition, if one (1) *possesses causal influence* on the respective issue (possibility and ability) and (2) is in a specific, normatively relevant relation to it (by higher mandate, self-commitment, or the particular value of the object) and is thus obliged to fulfil the task [1:33].

The non-classical concept of responsibility turns out to be a manifestation of the problems of co-ordination and control that result from modern societies and complex organisations with division of labour [1:34]. Tasks with high responsibility typically demand high competence and can only be adequately fulfilled having a certain discretionary power. This change in the concept of responsibility bears the risk of playing off against each other moral substantial responsibility and functionally oriented accountability.

ICT prominently contribute to the noticeable intensification of problems of attribution, comprehensibility, irreversibility.... The factual and often not reinsured and uncontrollable dependence on information systems grows. This vulnerability should be reason enough for computing professionals to invest in reflection and discussion of the diverse facets of responsibility. A change in emphasis concerning value judgements can be discerned because IT affects more areas of life as former technologies. While engineering ethics used to focus on issues of public safety and risks to life and health, IT-code of ethics refer to the public interest in general, explicitly including social aspects (e.g. [7, 8]). How the concept of "responsibility" may respond

to the new concerns raised by globalisation and virtualisation remains an open question, which we do not attempt to answer here.

2. RESPONSIBILITY IN PRACTICE

The concepts described help us to understand some of the phenomena and structures we experience within the computing profession/business. We now can use essential notions like intention, voluntariness, autonomy, obligation, possibility of foresight, causal influence, (care) responsibility and “attribution as responsibility” when discussing the following issues: (a) What does the notion of insight mean with regard to the process of attribution? (b) Insight into consequences of actions does not suffice to know which action is appropriate. (c) How is problem awareness related to public perception? (d) What means strict liability for risks in the field of system development?

2.1 Insight: In-between foresight into consequences of action and the problem of attribution

“Human failure” is seldom sole cause of catastrophes and accidents, rather the last link of a chain. Therefore design of technological artefacts should minimise risk and consequences of error [19]. Reason [22] discerns active failure (of front-end actors, e.g. operators) and latent failure. Latent failure originates from preceding actions, involves working conditions and load, competing demands, and is caused by designers, developers, decision-makers and managers. Thus, accidents usually result from a mesh of effects and interacting causes; responsibility for failure or accidents is indirect and spread over many people and institutions. Complex systems and division of labour limit insight into consequences and influence of single actors. Therefore the individual must be relieved from a disproportionate sole guilt. As side effect responsibility runs the risk of being diffused.

What about the responsibility of IT-professionals (we use the term “profession” in a broad sense, avoiding a discussion about computing as profession) and developers of software? As designers of technical artefacts they take part in responsibility for latent failures. Present law and professional practice do not meet this. IT-companies, different from engineering and construction business try to enforce limited liability for product deficiencies or defects. Usually users and customers are not able to detect program flaws and code is not accessible. One hears stories of

internally known program flaws, which are kept secret to the customer. In some respects customers share responsibility: often orders are given to manufacturers promising quick and cheap delivery instead to those with sound calculations, including methodical requirements analysis and testing. Users and the general public have been conditioned by the software industry to accept bugs and bad usability like laws of nature (cp. [4]). These are some characteristics of the computing culture, which lead to latent failure and (in part) can be read as a refusal (of IT) to take responsibility or as overcharge and lethargy of society to attribute accountability (to IT).

But without time resources, detailed analysis of the usage context, and subsequent testing, developers have little chance to foresee consequences of software usage. Twisselmann [23] describes how time pressure, non-adequate processes of software development and educational deficits lead to software with unintended impediments for work processes. Involved persons on all sides seldom realise how far reaching the systems influence will be. Involving separate groups of people for project phases (consultants for requirements analysis, developers coding) interrupts communication and knowledge flow. Frequently, software developers can only inquire mediating persons about the usage context. Thus, it is almost impossible for them to develop an adequate conception of the work context. Sometimes only after installation it is discovered that specified functionality is not sufficient or does not fit into existing work processes. Here we obviously deal with a systemic conditioned source of latent failure which leaves actors little chance of insight into consequences of design actions.

2.2 Action: In-between possibilities to take action and their suitability

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. "Time is money and thus responsibility is overridden under time pressure. (...) competition was so hard, that in case of conflict companies always dispensed with criteria like privacy or minimising strain" (translated from [2:24]). In an e-mail inquiry work load and time pressure were mentioned most often as hindrance to put good intentions (usability, privacy) into practice [9]. Conflicts of interest result, considering not only acting in the best of customers and users, but also the personal environment (protection of colleagues and subordinates

from work load) or the company (keeping deadlines to prevent fines)¹⁰. Loyalty/Allegiance to the employer (contractual obligation) often compels employees to conceal knowledge and give non-optimal recommendations to customers: “When I/my employer is contracted to sell certain products, then I must recommend it, even if there might be a better choice. If the company earns from outsourcing, I cannot recommend the customer to keep his data and train his own people.” (translated from [9]) The only option is indirect hints. The individual’s position inside the organisation discerns which path of action is feasible and how well risks can be calculated.

The US law provides the construct of “whistle blowing” for extreme cases involving a risk to the public. Although judges often use their discretionary powers in favour of workers (see [5]), this construct does not exist in German law. But when exactly does risk to the public begin? And what about the proportionality of action? An example: If the design of an information system ignores privacy laws, while this is not exploited in system usage, this represents no “acute danger”. The employee should point it out to superiors. What next? Can we demand to intervene further and thus to endanger the relation of trust to superiors and to jeopardise his/her job? The individual has to reflect whether the risk to the public outweighs his/her interests and what his/her conscience can live by.

It is important to transcend the concept of “whistle blowing” which is imbued with tragic heroism. These cases are only the tip of an iceberg, plus having antecedents. As Lynch & Kline [17] analyse for engineering practice, it is workplace culture and routine decisions, which in the long run lead to unintended results. We need to attend to the social context of everyday workplace and company culture, where small incremental adjustments, which appear rational at every stage, accumulate and in result decide over the trajectory of events. When in retrospective this trajectory is perceived as false, these precedents of small routine decisions have been established. Thus moral dilemmas have a long period of “incubation”. At the same time engineers are active producers of mundane practice and workplace culture and thus shape available options. Therefore it is a crucial skill (and should be goal of teaching ‘ethics’) to recognise implicit assumptions and everyday ethical issues in poorly structured problem fields, to develop creative solutions, and to identify available resources (cp. [6]). Much alike us, Lynch

¹⁰ The Software Engineering Code of Ethics [8] deals with many of these issues and the mesh of responsibilities for different people or issues. The public interest comes first and can serve as argument to depart from other obligations. Nevertheless, Codes of Ethics tend to include primarily items referring to professional behaviour (and laws, which should be obeyed anyway) rather than to morality or ethics (see [24]). This is true for [8] and the GI ethical guidelines [7]. Reconsidering the described characteristics of the computing trade, even true professional behaviour seems rare.

and Kline argue for preventive ethics instead of crisis ethics¹¹. Computing culture as well is result of small routine decisions. Thus computing professionals should be reminded of actively shaping workplace culture; Codes of Ethics might fulfil this function of reminders.

2.3 Problem awareness and public perception

Whenever problem awareness for certain risks increases, the perceived pressure to act and to accept consequences rises, too. Whenever there is public discussion about a risk, similar cases are more easily uncovered due to enhanced awareness. Handing over originators to justice gets socially accepted. In addition the originators get under pressure to change their priorities. Sadly there is little public debate relating to information technology, its risks, and the responsibility of its protagonists. Exceptions are some discussions about vulnerability and IT-security (e.g. viruses and Y2K). But even these showed a high level of inertia: media noise (and IT bustle) about Y2K started not until a year before the “dangerous date”. Usually it was analysed in terms of economic consequences, only seldom as a problem of ethical relevance (see [10]). Any detailed analysis afterwards seemed to be rather taboo. Interest about viruses, perceived as immediate menace, is stirred quickly and loses impact quickly. So: “Why are risks associated with IT-technology so often hushed up?” We believe that it is necessary to stir up discussion and sharpen awareness inside our trade and – even more important – in the concerned (and afflicted) public.

2.4 Obligation for precaution and strict liability

Strict liability makes a systems’ operating authority liable for risks and imposes obligation for precaution upon it. This heightens motivation to take every precautionary measure. However this diffuses the distinction between intended and unintended consequences and diminishes the importance of the ability to foresee consequences of action. Both were essential prerequisites for individual responsibility. A further question is to *whom* strict liability applies – also to developers of a system? A developer writing a database does not know exactly, what kind of data will be put into it. The administrator cannot always verify which items users put into which fields,

¹¹ In addition they explain why cases of whistle blowing are detrimental in classroom. Giving students only “all-or-nothing” cases makes them feel that ethics “involves nothing more than a trade-off between sacrificial heroism and amoral self-interest” [17:209].

what this means, and how the data is used and interpreted. For developers of general technical systems it is nearly impossible to prevent misuse of products [12]. There is no chance to prevent indirect effects of tools, e.g. compilers can be used to build products, which then are put to unintended use.

3. AWARENESS OF CHOICES

This discussion of the in-betweens was to give some structure to the messy entanglement of practice, pointing to general problems and issues. Structural problems like computing work conditions and practices of industry towards customers demand collective approaches. Changing these conditions would make ethical choices easier feasible. But opening decision space does not suffice. One has to believe in the existence of choices to search and exploit them. If enough local changes lead to synergy, changing local workplace culture can have higher-level effects. Our discussion pointed to the possibility of choices and to problems in recognising and using them. A fine awareness of both is necessary, as ethical judgement is inescapable in (messy) practice. Yet this awareness is widely missing – it is not cultivated as part of computing culture. When presenting scenarios like those mentioned in this paper to computing students, these often did neither recognise that moral issues were involved nor did they see any choices.

Both Lynch & Kline [17] and Forester [6] recommend focusing on everyday practice in classroom. [17] recommend using hypothetical, but realistic cases and use of role-play to help students improve awareness for features of ordinary practice, recognise everyday ethical issues, and develop creative solutions. Forester points to the value of practical stories for students: stories with “moral depth provide substance for their inquiries” [6:243]). Their messiness is part of their power, giving examples of the work of identifying issues and values, judging and improvising morally. Unfortunately, telling such stories has no tradition in computing. Either people feel their stories to be insignificant, or they are in danger of disclosing company secrets, or they simply miss time for writing them up. We not only need stories of conflict, but also stories of what is possible, stories of good choices. We need to cultivate ethical judgement and moral improvisation in the computing culture, much like demanded [6] for public sector mediators and planners. *Ethical judgements* means to “fit action to circumstance, to see general principles in the light of contextual details (and vice versa)” [6:223]. *Moral improvisation* refers to the ability of acting

situated, contextualised, while following principles¹², to recognise and respond to the richness of problems (both in details and in principles and obligations), “not just get the facts, but the facts that matter”, while “resisting the rush to interpretation”. Although the extent of these demands on behaviour and self-perception should not be transferred one-to-one, we feel that computing professionals have comparable responsibilities, especially when collecting requirements or translating them into system architectures (cp. [23]). The following statement could as well refer to computing professionals: “Which facts to take as significant, and which rules and responsibilities, goals and obligations, promises and understandings, to fulfil in what ways – these are inescapably moral matters that practitioners must face all the time.” [6:241]

We want to thank the members of working group Betula at IRIS 24, Sara Eriksén and Günter Ropohl for discussions and remarks on earlier versions.

REFERENCES

- [1] Bayertz, K. (1995), Eine kurze Geschichte der Herkunft der Verantwortung, in K. Bayertz, ed., ‘Verantwortung: Prinzip oder Problem?’, WBG, Darmstadt, pp. 3–71.
- [2] Boedicker, D. & Biskup, H. (2000), ‘Grau ist alle Theorie’, *FfF-Kommunikation* 2000(1), 22–25.
- [3] Burke, J. (1975), Kesselexplosionen und bundesstaatliche Gewalt in den USA, in K. Hausen & R. Rürup, eds., ‘Moderne Technikgeschichte’, Kiepenheuer & Witsch, Köln, pp. 314–336.
- [4] Constantine, L.L. (2001), ‘Back to the Future’, *CACM* 44(3), 126–129.
- [5] Däubler, W. (2000), ‘Wann darf ein Arbeitnehmer ‘Nein’ sagen?’, *FfF-Kommunikation* 2000(4), 29–31.
- [6] Forester, J.F. (1999), *The Deliberative Practitioner. Encouraging Participatory Planning Processes*. MIT Press, Cambridge/London.
- [7] Gesellschaft für Informatik (GI) (1994), Ethische Leitlinien (Ethical Guidelines), Bonn.
- [8] Gotterbarn, D., Miller, K. & Rogerson, S. (1999), ‘Software Engineering Code of Ethics is Approved’, *CACM* 42(10), 102–107.
- [9] Hornecker, E. & Bittner, P. (2000), ‘Vom kritischen Verhältnis zur Berufspraxis in der Informatik – Ergebnisse einer Befragung’, *FfF-Kommunikation* 2000(1), 33–39.
- [10] Kappelman, L. & Cappel, J.J. (1999), ‘The Year 2000 Problem: Ethical Implications’, *Information Systems Frontiers* 1(2), 127–139.
- [11] Kelsen, H. (1941), *Vergeltung und Kausalität – Eine soziologische Untersuchung*, W.P. Van Stockum & Zoon, The Hague.
- [12] Kern, R. (1990), Mißbrauchsschutz für Software unter Beachtung rechtlicher, technischer und organisatorischer Aspekte, diploma thesis, FB Informatik, TH Darmstadt.

¹² Forester refers to Nussbaum’s [20] usage of the metaphor of theatrical improvisation for explaining the idea of moral improvisation. Theatrical improvisation does not mean acting ad-hoc and without principles, but requires even more attention.

- [13] Kling, R (1996), *Beyond Outlaws, Hackers and Pirates*, in R. Kling, ed., *Computerization and Controversy*, 2nd ed., Academic Press, San Diego, pp. 848-869.
- [14] Lenk, H. & Maring, M. (1993), *Verantwortung – Normatives Interpretationskonstrukt und empirische Beschreibung*, in L. Eckensberger & U. Gähde, eds., 'Ethische Norm und empirische Hypothese', Suhrkamp, Frankfurt am Main, pp. 222–243.
- [15] Lévy-Bruhl, L. (1884), *L'Idée de Responsabilité*, Hachette, Paris.
- [16] Lutterbeck, B. & Stransfeld, R. (1992), *Ethik in der Informatik – Vom Appell zum Handeln*, in W. Coy & al, eds. (1992), 'Sichtweisen der Informatik', Vieweg, Braunschweig/ Wiesbaden, pp. 367-378.
- [17] Lynch, W. T. & Kline, R. (2000), 'Engineering Practice and Engineering Ethics', *Science, Technology and Human Values* 25(2), 195–225.
- [18] Nietzsche, F. (1887), *Zur Genealogie der Moral*, Verlag von C.G. Naumann, Leipzig.
- [19] Norman, D. (1988), *The psychology of everyday things*, Basic Books, New York.
- [20] Nussbaum, M. (1990), *Love's Knowledge*, Oxford University Press, New York.
- [21] Picht, G. (1969), *Der Begriff der Verantwortung*, in G. Picht, ed., 'Wahrheit, Vernunft, Verantwortung – Philosophische Studien', Klett-Verlag, Stuttgart, pp. 318–342.
- [22] Reason, J. (1990), *Human Error*, Cambridge University Press, Cambridge, MA.
- [23] Twisselmann, U. (2000), 'Informatik und Arbeitsumgebungen', *FifF-Kommunikation* 2000(1), 28–31.
- [24] Weber, K. (2002), 'Grenzen von Ethikcodizes'. *Kriterion* 2002(15), 3-12.