Discussion

Michael Koch*, Claudia Müller, Susanne Boll, Volker Wulf and Albrecht Schmidt

People and practices must be at the center of IT research and development in a digitized world

Human-computer interaction and the practice of building interactive systems – what is still going wrong?

https://doi.org/10.1515/icom-2024-0061
Published online December 5, 2024

Abstract: This article has been written as an attempt to partially answer a question we as HCI researchers often hear: What has HCI contributed to the practice of developing software – or: why do (complex) systems still often fall short of being usable? We are trying to frame the topic and list some problems, but also some upcoming solutions. As to be expected, the problem is not that there is no knowledge available about how to build usable systems, but the knowledge about the importance of this topic has not yet arrived everywhere in software development.

1 The digitization of society

Digital systems have been pervading our everyday lives for a long time. The very early systems often were initially developed by experts for special applications, to fly to the moon, to network banks internationally. Gradually, digital systems have become part of our everyday professional lives, digital documents for business correspondence, the goods management system of a logistics company, the checkout system for a retailer, the flight booking system for international airlines. With the enormous developments of the Internet and the World Wide Web, the availability of wireless communication and miniaturization, we have entered the digital age, a kind of digital gold rush. Everything is possible, everything is networked, everything is always accessible. In the past 20 years, digital systems have entered our lives everywhere, when we communicate, when we shop, when we travel, when we work, when we learn, and even when we play sports. In other words, everywhere.

However, despite the long experience in developing interactive systems, in practice we still find many examples where IT user interfaces are developed without people in mind. These poorly designed systems can lead to many unexpected problems.

Insufficient user orientation and user involvement is emphasized in the literature as a central factor for faulty and failed IT projects. The Standish CHAOS report series, which identifies success and failure factors in (bi-)annual surveys of American IT projects, is cited prominently.[1] Even if the analyses carried out since 1995 are sometimes criticized for their methodological rigor, a sad picture emerges for the last almost 30 years: the figures from 1995 have hardly changed by 2023: approx. 20 % of projects are successfully completed, 50 % with massive changes in the course of the project and 30 % are never completed. The report series presents massive cost increases as the results of those projects with changing requirements as well as for those that are never completed.[2] Authors who refer to the reports or conduct own empirical studies on success factors for IT projects repeatedly highlight lack of user involvement as the main cause of failure in IT projects.[2,3] Interestingly, or better sadly, Gulliksen et al., referring to a current figure on the vast amount of functionalities implemented in software that are never being used with showing that “close to half of the functionality built are never used by anybody” (p. 113). They suggest this observation as a thought-provoking idea

***Corresponding author: Michael Koch**, Universität der Bundeswehr München Fakultät für Informatik, Neubiberg, Germany,
E-mail: michael.koch@acm.org. https://orcid.org/0000-0002-9694-6946
Claudia Müller and Volker Wulf, , Universität Siegen Fakultät Wirtschaftswissenschaften, Siegen, Germany,
E-mail: Claudia.Mueller@uni-siegen.de (C. Müller),
volker.wulf@uni-siegen.de (V. Wulf). https://orcid.org/0000-0001-8534-546X (C. Müller). https://orcid.org/0000-0002-1214-1551 (V. Wulf)
Susanne Boll, Department of Computing Science, Carl Von Ossietzky Universitat Oldenburg Fakultät II für Informatik Wirtschafts- und Rechtswissenschaften, Oldenburg, Germany, E-mail: Susanne.Boll@offis.de.
https://orcid.org/0000-0003-4293-1623
Albrecht Schmidt, Ludwig-Maximilians-Universität München, Munich, Germany, E-mail: albrecht.schmidt@um.ifi.lmu.de. https://orcid.org/0000-0003-3890-1990

to think about how much of the development time could have been used for strengthening the most frequently used functionalities.

In contrast to the examination on software failures, studies such as[4] with a literature review on user involvement in software projects, emphasize benefits and challenges where the benefits clearly outweigh the challenges.

2 Why aren't we there yet?

Let us go back a few decades to find out why we are not yet there. People develop software and digital systems for people. But the developers' view of the users was not always at eye level. For a long time, the view was that the core of digital systems was computer science, programming, and function. As a powerful digital department store or efficient cash register system. The question of use was pushed to the back of the development. Use had to follow practical criteria; questions of usefulness or ergonomics or even the effects of poor user interfaces on users were not the focus. User interfaces were subsidiary, with which one dealt only later in the development, that one could hand over to interface design agencies.

The unspeakable notion that systems should be developed for the "dumbest assumable user",[1] the DAU, also dates from this time. The mindset behind this is to be viewed critically in many respects. It is expressed that the developers, the computer scientists and the engineers are the ultimately wiser ones and know what users need, and users certainly do not know what they need. Henry Ford is often quoted saying, "If I had asked people what they wanted, they would have said faster horses." The quote is used to confirm to oneself that people cannot design new technologies because they lack the vision. If you ask people on the street, how would you know what science or research is capable of. But what they can express very well, and what is expressed in this quote, is the following: They want a faster horse and that is what a car is in fact. The notion of the "dumbest user" raises another problematic point: if you cannot use the system, or can't use it properly, then you are supposedly dumb. After all, smart developers have developed it well, so why shouldn't it be usable? Furthermore, it could be assumed that being able to use a system is perhaps a sign of cleverness, and that "stupid" people simply cannot do that. At the core of this is the self-image of computer science and the need to critically examine its own scope of ideas that have been developed in the lab or at the desk and to bring them into a better fit with the real practice of people in everyday life, at work, while traveling, etc. This can only be done well with the involvement of users and research in real practice contexts.

3 Recognizing the importance of user-centeredness

It is important to create an awareness that one has to start with the users and their needs – to put these issues in the center from the beginning. Failures or problems may lead to "remembering" – but the next project will again bypass the users. This is because user and practice orientation are often perceived as a nuisance – as too much effort in the development process. The image that technology is developed by experts and then "handed over" to users has changed significantly in recent decades. Users increasingly play a central role in development. It is well documented – whether for large projects or smaller IT products – that due to a lack of involvement of users and too little knowledge about the practice in which the future technology is to be used, projects fail and get out of hand in terms of costs and time. Using digital interactive systems does not have to be a challenge, but must be easy for the task at hand. The field of Human-Computer Interaction and the development of professional communities such as the Usability Professional Association have contributed enormously to this.

Both at the political level and in the meantime in many research groups in the field of HCI, the approach of user orientation and participation is regarded as an essential element of a successful design of easily usable and meaningful IT products for all user groups. On the one hand, the focus is on good usability of the products themselves, but on the other hand also on the processes of use and appropriation. For example, many research groups collaborate with organizations and stakeholder groups that aim to use the digital world for older people and groups of people who are not familiar with digital technology and, for example, to establish local learning and experience spaces for older people in urban and rural regions.

In addition to science and practice, teaching has also long taken up the topic with now diverse bachelor's, master's and continuing education programs around Human-Computer Interaction.

In a piece for the Interactions magazine Saul[5] wrote about a course for teaching HCI to programmers. In the introduction he reported that at the University of Calgary HCI only recently changed from a specialist course

1 In German "DAU" is an acronym for "Dümmster Anzunehmender User" (dumbest assumable user) – a better translation of the term might be „worst case user".

to a heavily attended mainstream course. In his proposal for a course outline he both includes "designing without the user" and "designing with the user" as central topics. Elisabeth F. Churchill, Anne Bowser and Jennifer Preece also wrote about "Teaching and Learning Human-Computer Interaction".[6]

4 Impulses from research

The research field Human-Computer Interaction (HCI) is situated at the intersection of computer science, behavioral sciences, design, media studies, anthropology, and several other fields. The field focuses on the human side of interactions with computer systems so that we have the best possible experience when "communicating" with machines. HCI emerged in the 1970th (see for example[7–9]) and was predated by JCR Lickliders forcast of three phases for how humans relate to machines: human-computer interaction, human computer symbiosis, and ultra-intelligent machines.[10]

In addition to scientific results as nicely categorized by Wobbrock J. & Kientz J.[11] there are results from the field aimed at practitioners, like the eight golden rules for successful Human-Computer interaction listed by Ben Shneiderman in his book "Designing the User Interface"[12] (first published in 1987):

1. Strive for Consistency
2. Enable Frequent Users to Use Shortcuts
3. Offer Informative Feedback
4. Design Dialog to Yield Closure
5. Offer Simple Error Handling
6. Permit Easy Reversal of Actions
7. Support Internal Locus of Control
8. Reduce Short-Term Memory Load

In Germany the topic of involving (end) users in software development can be traced back to different roots. One strand is about "software ergonomics" – forced forward by researchers like Wolfgang Dzida. Already in April 1977 he published a report on the usability of dialog systems in German.[13] In 1978 Dzida followed up with a publication in the Proceedings of the 3rd international conference on Software engineering.[14] This opened the discussion of ergonomic fundamentals in dialog design – nationally in Germany and internationally.

Another early strand of work on user involvement is by Christiane Floyd. She was one of the first scientists in Germany to draw attention to the fact that digitization and computer science must keep people in mind. Christiane Floyd had already been appointed Professor of Computer Science (Software Engineering) at the TU Berlin in 1978 and worked in Berlin and later in Hamburg, among other things, on the STEPS process model, which understands software development as an evolutionary activity that involves the user in the design process. So early on, science pointed out how development should be done – and also presented concrete solutions. Floyd was followed by many other approaches and calls for participation around the world – some of them combined with field-tested solution modules.

Floyd's work is closely related to approaches of the Scandinavian School of Participatory Design. The Scandinavian school of participatory design also pays special attention to the development of socio-digital environments that include all affected groups of people – i.e., especially marginalized user groups as addressees of technology development.

The important sets of rules developed for decades in research and practice are not least anchored in inter/national standards, e.g. ISO 9241-210:2019 (Ergonomics of human-system interaction–Part 210: Human-centered design of interactive systems (ISO 9241-210:2019); German version EN ISO 9241-210:2019)). The application of ISO standards for the design of interactive systems can nowadays be considered a standard procedure. There are also special test procedures for safety-critical systems, e.g. in medicine or for air traffic control. With usability and user experience engineers, a job description has been professionalized in practice and is now firmly established.

5 Conclusions

In conclusion, it can be stated that access to digital offerings is still in need of improvement. In our opinion, however, this is not so much due to a lack of methods, concepts or rules. It is more due to the fact that the development processes are not sufficiently oriented to the real everyday living or working practice of people. Practice- and user-oriented research must be given much more focus. This requires a better understanding of participatory and interdisciplinary approaches, and it needs more space and time for processes of understanding.

More understanding is needed of how rules and regulations can be combined with suitable procedural methods to ultimately design IT products that provide users with positive support in their everyday life and work.

And finally, the ongoing diffusion of the topic into curricula for software developers has to continue. Because still, building systems is still too much seen as a technical endeavor. When looking at computer science curricula, these are still focused on technical topics. Computer science is seen as a technical science.

What might help here is to imagine computer science not merely as an engineering discipline but as an interdisciplinary design-oriented field – particularly to include empirical methods and social science courses in the computer science curriculum.[15,16] Other lines of current development to bring social practices and technical development into a better alignment are being provided by the concept of "Practice-based Computing", featured by the European Society of Socially Embedded Systems (EUSSET, https://www.eusset.eu/) or by the field of Socio-Informatics.[17]

An early version of this article in German has been published in October 2023 in the blog of the German Computer Society (GI) at https://gi.de/themen/beitrag/menschen-mitdenken-human-computer-interaction.

References

1. The Standish Group The Chaos Report, 1995. Available from: https://www.csus.edu/indiv/v/velianitis/161/chaosreport.pdf.
2. Gulliksen, J.; Hussain, D.; Göransson, B. Collaboration between HCI Design and Software Engineering. In *Designing for Usability, Inclusion and Sustainability in Human-Computer Interaction*; CRC Press: Boca Raton, 2024; pp 78—117.
3. Lemon, W. F.; Liebowitz, J.; Burn, J.; Hackney, R. Information Systems Project Failure: A Comparative Study of Two Countries. *J. Global Informat. Manag. (JGIM)* **2002**, *10* (2), 28—39.
4. Kujala, S. User Involvement: a Review of the Benefits and Challenges. *Behavi. Informat. Technol.* **2003**, *22* (1), 1—16.
5. Saul, Greenberg "Teaching human computer interaction to programmers," *Interactions 3* (4): 62—76.
6. Churchill, E. F., Bowser, A., Preece, J. Teaching and Learning Human-Computer Interaction: Past, Present, and Future. **2013** Interactions, March + April 2013.
7. Carroll, J. M. Human Computer Interaction (HCI). In: *Encyclopedia of Human-Computer Interaction*, M. Soegaard and R. F. Dam (eds.), The Interaction Design Doundation, Aarhus, Denmark, 2009: http://www.interactiondesign.org/encyclopedia/human_computer_interaction_hci.html
8. Grudin, J. Brian Shackel's Contribution to the Written History of Human-Computer Interaction. *Interact. Comput.* **2009**, *5—6*, 380—374.
9. Myers, B. A. A Brief History of Human-Computer Interaction Technology. *Interactions* **1998** *5*, 2, 44—54,
10. Licklider, J. C. R. Man-computer Symbiosis. *Trans. Human Fact. Electron.* **1960**, *HFE-1*, 4—11.
11. Wobbrock, J. O.and J. A. Kientz . Research Contributions in Human-Computer Interaction. 2016 38—44. interactions 23, 3 (May + June 2016). https://doi.org/10.1145/2907069.
12. Shneiderman, B.; Cohen, M.; acobs, S.; Plaisant, C.; Diakopoulos, N.; Elmqvist, N. *Designing the User Interface —Strategies for Effective Human-Computer Interaction*, 6th ed.; Pearson: London. 2016 (first edition was by Ben Sheiderman as only author in 1987).
13. Dzida, W.; Herda, S.; Itzfeldt, W. D.; Schubert, H. *Zur Benutzerfreundlichkeit von Dialogsystemen*, Vol. *35*; GMD Bericht, GMD: Bonn, April 1977.
14. Dzida, W.; Herda, S.; Itzfeldt, W. D. User-perceived Quality of Interactive Systems. ICSE '78: Proceedings of the 3rd international conference on Software engineering, 1978.
15. Bernstein, A. Informatik ist auch eine Sozialwissenschaft!. *Informatik Spektrum* **2013**, *36*, 461—462.
16. Connolly, R. Why Computing Belongs within the Social Sciences. *Commun ACM* **2020**, *63* (8), 54—59.
17. Wulf, V.; Pipek, V.; Randall, D.; Rohde, M.; Schmidt, K.; Stevens, G., Eds. *Socio-Informatics*, online edn; Oxford Academic: Oxford, 2018.