

How do developers meet users? – Attitudes and processes in software development

Jan Gulliksen¹

¹ Uppsala university, Department of IT/HCI, PO Box 337, SE-751 05 Uppsala, Sweden
Jan.Gulliksen@hci.uu.se

Abstract. This keynote paper argues for an increased understanding of the various roles involved in the development work to be able to achieve the goal of developing increased usability. Human Computer Interaction (HCI) has for a long time been arguing for the sake of the users, but to be able to deliver methods, tools and processes that better fit the needs of developers we should study and understand development work to a larger extent. This paper discusses what HCI and software engineering contributes to each other's disciplines as well as what research contributes to practice. This also means an increased understanding of what research that counts in the academic context. Finally I argue that research should focus more on real problems of real development work, rather than on research that is easily publishable.

Keywords: Usability, User-centered systems design, process, attitude

1. Introduction

Why do we do what we do? How come we started doing research? What do we think we contribute and who cares?

A couple of years ago I was an opponent on a PhD defense. I asked the defender: “*What is the real objective behind your research?*” The thesis was put together in a standard way, posing research questions in the beginning, examining the state of the art, setting a methodology for answering the research questions, presenting results and discussing them in relation to previous research. However, there was no discussion on the main objective behind the research. I repeated:

What is the main research objective? To make it easier for you, I will give you three options,

- 1. You want to change the world*
- 2. You want to contribute new knowledge to the world*
- 3. You want to get your PhD*

The defender tried to get out of the situation by proposing a fourth alternative. I said: “*Please do, but if you do that I will show that whatever you propose actually is one of the three options I have proposed*”. So, the defender then claimed that he wanted to change the world, and my immediate response was to ask him to explain why. During his explanation as to why his research would change the world it became obvious for him that *that* was not the main objective of his research. He then said that he wanted

to contribute new knowledge to the world, and of course I asked him to motivate his contribution. Unfortunately the work he had done had some methodological flaws meaning that the only conclusions that could be drawn from the work was that if he had conducted the experiments a bit differently one might have been able to draw some conclusions over the results. As a consequence of the discussion he finally said: *“I think I want my PhD”*.

It is important here to clarify that this story does not have the purpose of making fun over someone’s work or studies, but rather to have this serve as an illustration of the utmost importance of this discussion. To clarify, I mean that any of the answers are equally good. It is excellent if people do want to change the world – that is where and when we really can make a difference. Also it is very good if people do contribute new knowledge to the world. HCI is a field with quite a few white spots on the map, why new knowledge is essential. Also it is very good if people do want there PhD, because industry has a great need of well educated and experienced HCI researchers (even if they are not fully aware of it yet). The important thing is that by asking yourself what your main research objective is before starting you might be able to focus on the really important questions rather than focusing on what is easy to get valid results out of.

2. What research is promoted?

What research is promoted? Given that the competition at conferences within the field has risen tremendously, and given the relative importance of getting publications from a personal merit value point of view, the research questions that people investigate are often set to fit publications not necessarily meaning that it focuses on what is important for the world. Conferences and journals promote quantitative studies before qualitative, positivist research before hermeneutic studies, experimental research before case studies and action research, and hence the research questions that people select are biased towards what easily can be published. Let me take an example. Defining a new evaluation method is relatively easy to do and to evaluate the validity and reliability of such a method is also relatively straightforward and can be done in an experimental setting. But, do we really need a new evaluation method? If the goal is to increase the overall usability of a product is it really the new method that makes the difference? Most likely not, it is the work and knowledge of the people involved in the design and development of the system that makes the difference, and methods may of course provide the hooks that those professionals can hang their knowledge upon, but it is most likely that the knowledge and experience of the person doing the work that makes the difference. On the other hand to really improve a development process in practice will inevitably pose a much bigger challenge. And it is the practice, the context it is set in that pose the challenges. To my experience in 9 cases out of 10 things will happen in research in practice that aggravates the scientific results, yet there are so much knowledge and experiences to be drawn from such cases, yet they are much more difficult to get published in a scientific manner. My argument here is that we should make more efforts of promoting research that actually contributes to changing (and hopefully improving) the world than what is easily

evaluated and justified. Research should also be judged based on its contribution to the development of practice or knowledge that is useful for practice.

One of the reasons why this is problematic is that computer science research to its nature is a positivist type of research and therefore research into computer science seldom reaches the level in which it becomes important to judge qualitative terms. One of the most well known professors at my university responded to my discussions around this problem: *“I do not understand your problem. In my research group we invent a new processor and we can measure that it is 5 % quicker than previous processors. That is what I call research.”* Hence the problem of improving the practical utility of the research is also a development of the scientific knowledge and breadth of the knowledge from philosophy of science.

3. How can research contribute to the development of practice?

I have for a very long time been cooperating with public authorities to improve their system usability and computer supported work environment. The common denominator of a lot of that work is that it is very difficult to specifically point out what our contribution have been? There is not a particular method, tool or technique that they adopt today that directly comes out of our research cooperation. But, at the same time it is clear that our cooperation has had a tremendous impact on the way that they now perform their development. It is clear that the awareness and knowledge about usability and various approaches has developed significantly, at all levels in the organization. But, none of the developed or proposed methods were actually adopted as they were. Rather the organization needed to feel that they were in control and hence they invented and developed their own approaches for these things. To a large extent the approaches were adapted from the approaches proposed from research, or inspired by research activities, but it was very important that the initiative came from the organization.

One of the important contributions that the research has on practice is to emphasize and develop the conceptual framework. This relates both to HCI-related terminology as well as domain terminology. As we are very well aware of, HCI-related concepts such as prototypes, design, evaluation, etc. has very different connotation to different people within the organization. Research can here play an important role in specifying and elaborating the conceptual framework, but it needs to be based on the practice of the developers. This means that we must understand how various concepts are interpreted by the different professionals using it, rather than how the dictionary defines it. We can also make use of this understanding and base the approaches we suggest on the commonly agreed upon definitions. Let me take a couple of examples:

One important distinction one can make is between a user test and a use test. User testing has been used to a large extent to emphasize the importance of letting users test the products and gather knowledge on how systems can be improved based on this. On the other hand, others feel that by using the concept of user testing, we might risk a false understanding that it is about testing the users. Of course this is not the case, and I doubt that many people misunderstands this. The concept of use test would of course be much better in this sense, but then on the other hand some people might

mistake it for testing where you actually do not need to involve users. As an expert you might be able to apply a use test.

Another conceptual difference that is often discussed is the variety of concepts relating to any type of User-centered design (UCD). The ISO 13407 standard calls it Human-centered design, arguing that human is a much broader concept than users. On the contrary I have seen examples of the development of cow milking robots that was developed in a user-centered fashion that definitely did not focus on the humans. Another commonly used term is usability engineering, but to my experience usability engineering is much more focused on improving the usability through evaluation rather than focusing on the design in an iterative fashion. I have in other publications below elaborated on the conceptual framework at length and will not go into further detail here. The point is merely that if we want research to improve practice, we need to focus more on establishing a mutual understanding of the language used in the organization/project rather than on the academic conceptual discussion on the definition of various concepts. What practice needs is a common mutual understanding of the basic constructs among those involved in their development?

In addition to understanding the conceptual framework case studies and action research provides some of the most interesting examples of success and failures that we can learn a lot from. Extracting and generalizing the knowledge from various case studies and reapplying different variants of the approaches in other real life action research projects can move us towards a more successful development approach. However, action research and case studies are not very easy to document and publish and may often be open to critique due to the nature of the research approach. This must not stop us as researchers from using the practice as our field of research and generalize and extrapolate from these cases to a more general theory.

Successful UCD requires an understanding of the nature of the work of the people involved. HCI has for a long time been concerned about the users' situation, which has given us several very important studies and theories on the user's situation. However, in terms of having this knowledge contribute to development in practice we have not succeeded very well. Still with a number of user studies, user testing, etc. developers still continue to develop systems the way they have always done. HCI activities always become an add-on to the well-established processes that developers deploy. Therefore I believe that if we want the research to have any impact we need to focus on the main users of all our research results, namely the developers. We must understand the nature of software development and the basic values and attitudes among the developers and how these may be influenced because of the ways in which we communicate and disseminate our research into practice.

4. Understanding users

Understanding users, analyzing users, living with users, modeling users have been issues that over the years have received a lot of attention. Literature on users shows very different basic values and perspectives of how people view users and their contribution to the development work. This has led to a wide variety of different myths about users and their involvement in the process:

- **Users have unrealistic requests; we must manage their expectations.** It is not the question of developing exactly what the users requests. It is more the issue of interpreting the users wishes and in cooperation with them analyze what requirements that actually should be put on the system to meet their needs. And, by involving users on a regular basis you avoid unrealistic expectation and the need for managing expectations.
- **Users are not designers.** On the contrary we have experimented with users as designers of the conceptual design and interaction design of their system, with tremendously good results both in terms of the results that these design workshops gave and the efficiency of the process. But, such sessions do not, and should not, deliver formal specifications of the user interface design. They produce low-fidelity prototypes and scenarios and storyboards that show the work with the systems in a way that is understandable to users and that gives developers knowledge of the work beyond what can be captured in the formal models.
- **Users do not want change.** This is partly true – initially the users do not want change but when you start to work with the users on how they may develop and improve their work situation, most users are prone to change and do want to develop the ways in which they work. And if they don't you have either selected wrong users, or not given them the feeling that they actually have the opportunity to contribute.
- **Users adapt to everything.** True, but this should not be taken as an argument that the user's needs and wishes are of less value. Users can adapt to lousy situations and often do so, but their performance, and sometimes even the security, may be heavily influenced if you don't use the user adaptation as an excuse for producing systems that are easier to develop, rather than systems that are easy to use.
- **Users rarely adapt their systems to their special needs.** Also true, but mostly because the customization and individualization opportunities of the systems often are added on after the initial design of the system and not seen as an integrated part of the system. Careful considerations are required also in the design of the ways in which users can adapt their systems to their special needs.
- **It is all about education and implementation.** Education and a proper introduction of the system to the work setting is of course very important and needs to be planned and carefully conducted. But be ware that the implementation process might reveal a lot of usability problems that should be dealt with to increase the quality of the product. But far too often this is not possible due to delays earlier on in the process.

These different myths have also been conserved and emphasized in various processes and industrial applications. For example in the development of Objectory (the predecessor to Rational Unified Process – RUP), Ivar Jacobsson changed the concepts from users to actors in the description of the method. Originally he intended the methods to focus on the users and their needs, but when this was changed to actors, it also became the concept for a number of other things, leading to a view on actors

where there didn't need to be any users available at all. Consequently one of the biggest problems for heavy RUP users is that they risk losing the perspective that they are actually developing for users. The process became the goal in itself. Of course, processes are useful and necessary, but right now we need to fight back the user perspective in software engineering processes, as this was lost years ago.

Unfortunately this has led to a deeply rooted basic value of users and what their role is in development that devalues their knowledge and potential contribution. I often meet with developers that work hard to not have to meet with users and consider it as a very tough task when they have to face the actual users of what they have developed. This has also spread into the research community where several researchers argue against user involvement, because it becomes unpredictable and decreases the orderliness of development work. In a personal communication with Larry Constantine a number of years ago he concluded (sort of in an ironic matter): *"Frankly it's about keeping the users on an arm lengths distance"*

5. Understanding software development and developers

If we want to develop methods, processes, tools and techniques that can help developers in producing usable systems, we must understand software development and understand how software developers work. The HCI community has for a long time been advocating a user-centered approach to the development of applications for end-users, but have failed to adopt a user-centered perspective when it comes to looking at what developers need and can make use of to meet the goal of developing usable systems. I would therefore argue that an immature view on developers is potentially very harmful to the goal and our views and perspectives on developers need a much more serious attention.

Over the years in HCI conferences I have heard stories about "technicians" describing them as asocial individuals that never wash or shave, and that smells. In a review of a journal paper I wrote, one of the reviewers gave the following comment:

" Don't waste time with the geeks, unless you like hanging out with them. You'll never get their money. The proper self-assigned job of the authors ought to be to work out how HCI can make its deserved living from guaranteeing increased operational effectiveness, i.e. Value For Money - something the programmers don't give the project."

I have even heard presentations in an ironic manner referring to software developers as psychopaths, according to the definition of psychopaths as people who lack the ability to put themselves in another person's situation and lacking empathic abilities. What is the effect of such an attitude towards developers? Will we ever reach the goal of increasing the usability of systems, if we approach those who actually are going to construct and build the systems in this way. Of course not, and that is probably why we see the few researchers within HCI that deal with the practical application of HCI in practice focusing, not on issues relating to software development, but rather

relating to power relations, such as the work on procurement issues and power relations as a way to clarify who should be in charge.

I therefore strongly believe that we need to better understand the nature of the work of software developers. We need to meet the developers in their actual tasks and provide tools, methods, techniques and processes that contribute to their work. *“It’s all in a days work of a software developer.”*

6. What does/can software engineering contribute?

“... there is a prolonged period of confusion at the start followed by a frantic scramble to finish on time at the end” (McCoy, 2002)

Many software engineers that I have met act as if the world would have been much better without people. The ideal software development tool would be a machine that as input took requirements formulated according to a specific syntax and that delivered a perfectly functioning and usable system as output. Even if the utopia is achievable in the future it does not mean that software development in practice in the future will rely less on the activities of individuals involved in the process.

So, in order to understand what software engineering and software engineers can contribute, we need to study what it actually does contribute today, and what its major problems are. What does software engineering contribute (among other things):

- Requirements gathering
- Development processes
- Tools

What does it lack (among other things):

- Ways of facilitating and promoting user involvement
- Ways of supporting informal activities such as design
- Understandable formats for documenting designs

Software engineering is an excellent discipline in bringing orderliness to a tremendously complex process that lack transparency, since software during development rarely gives the users an image of the system to be. But software engineering also lack the humility of seeing its limitations and seeing the vast amount of HCI knowledge that is available for inclusion, rather arguing that HCI must go 90 % of the way towards the integration into software engineering (personal communication with Bonnie John). I believe that until the research has changed its basic values when it comes to these issues, the practice won’t change.

7. What’s the contribution of HCI and what should it be?

I attended a keynote address that Bill Buxton gave at the Local SIGCHI in Toronto in 2002. In his talk he listed the major inventions of HCI and concluded that all of them were invented before 1982, when the CHI conference was born and when the field

became a field of its own. His ironic conclusion was that the academic field of HCI had not contributed much to the practice of HCI. I might agree, that most of the development within the field has taken place outside of academia. However, I doubt that the development would have happened without the birth and growth of the academic field of HCI.

But, why have we not managed to make a significant contribution to the development of practice? One well-known example in Sweden is the development of systems for managing medical records. Despite studies showing the risks involved in developing medical records without considering the user perspective decades ago, organizations keep repeating the same problems over and over again. How come nobody listens to the warnings from researchers within the HCI field? Why have we not succeeded in getting the authority that is required for others to listen to what we have to say?

I believe that the field of Human Computer Interaction, as well as the practice of usability professionals, has a lot of things to contribute to the development process. For one thing the approaches gathered under the umbrella of User Centered Design (UCD) provides several methods and techniques for the process of developing interactive systems with the focus on usability and on the perspective of the users. However, UCD is not easily achieved, given the problematic contexts in which it needs to take place. Therefore UCD in itself has some specific requirements. For successful UCD we need to:

- Get the right support from management
- Determine the importance of usability in the organization
- Actively involve users in the design process
- Use understandable design representations in the development
- Involve usability designers.

Therefore we need to understand the usability profession as well. After all, these are people we must not only cooperate with, but also support. The usability profession is perhaps the most significant impact the HCI community has had on practice. As every profession they are faced with some problems. Some of the major problems we have encountered with the usability profession are:

- Lacking impact in the development process
- Lack of time, resources, knowledge, interest, etc.
- Unclear responsibilities
- Lack of consistency in naming

These problems are neither new nor unique for a newly established profession. Rather they are problems that probably would be easy to overcome if dealt with in the right way. An initial problem with all new roles and professions is that they need to justify their role to a higher extent than other already existing roles. One way of doing this is to alter the work more towards contributing to the design rather than commenting on and evaluating other people's work. The usability professionals must have "*skin in the game*" (Cooper, 1999). UCD professionals who focus on doing "studies", as opposed to generating designs and products, will always be perceived as peripheral."

(Siegel & Dray, 2003). The usability person must participate in all the user-centered activities, to prevent valuable information from being lost in the transitions between the activities, in accordance with the principle of integrated design (Gould et al., 1997).

8. Conclusions

At the end of the road, research in HCI and SE, as well as the work of usability professionals all deal with the problem of developing usable systems. I strongly believe that if we want to have any impact on practice, we need to focus more on how our research is implemented in practice and why it fails, and then make use of that knowledge to define new research questions. Over the years our research has produced new methods, tools, processes and roles – yet these have not changed the world to any larger extent. Very few of the methods that have been developed by researchers have been applied by anybody else than the inventors themselves. What is the reason behind this? How come others do not apply the methods to the extent that they deserve?

I believe that the main reason for this is that the development of such methods mainly are based on the market of science – by developing these methods you will get a neight study that is relatively easy to publish. On the other hand a good action research study or case study may be interpreted as theoryless, unstructured and even “sloppy”, and hence the opportunities of getting it published is much less. But the potential utility for practitioners is unquestionable. In other words, if you want to change the world a new process or a new tool will most likely not change the practice of developers. The real improvement or development comes when practitioners change their ways of work and ultimately maybe their view on the work and the context for which they are developing. Users become a necessary source of information, not only on the work of today, but also on the potential of the future, instead of being a source of aggravation and disturbance. Usability doesn’t happen by accident, but rather comes as a result of a thoughtful and enlightened design process focusing on usability. And how can we manage such a development process?

We need to work on the attitudes and basic values among everybody involved in IT development and use. We need to determine how important usability is for everybody? Further we need to determine who is responsible for usability and how do people respond to this responsibility? Finally, we need to get an understanding of what type of management support (project and overall management support) usability do have? To our experience every developer should get first hand experience from the users using the system. If practitioners can get immediate and first-hand experience of this perspective, that is when the people start to develop and when people tend to change

9. References

Cooper, A., (1999) *The Inmates are Running the Asylum*, SAMS, Indianapolis.

Gould, J.D. Boies, S.J., Ukelson, J. (1997) How to design usable systems, in: Helander, M. Landauer, T.K., Prabhu, P. (Eds.), Handbook of Human-Computer Interaction, Second revision, Elsevier Science B.V., Amsterdam, pp. 231-254.

Jacobson, I., Booch, G., Rumbaugh, J., (1999) The Unified Software Development Process, Addison Wesley Longman Inc., Reading.

McCoy, T. (2002). Usability: Who Cares? In Usability Gaining a Competitive Edge. IFIP World Computer Congress 2002. Hammond, J., Gross, T., Wesson, J. (Eds), pp 283-294. Kluwer Academic Publishers.

Siegel, D., Dray, S., (2003) Living on the edges: user-centered design and the dynamics of specialization in organizations, Interactions, X.5. pp. 19-27.

10. Bibliography

Following are some of the research that I have been involved in that may provide additional guidance to the issues that this keynote paper is dealing with. I have tried to order them into categories of fields that may seem relevant to your needs.

User Centred Systems Design:

Key Principles for User Centred Systems Design – Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson & Åsa Cajander (2003) Behaviour and Information Technology, Vol. 22, No. 6, pp.397-409

Usability professionals:

Usability Professionals – Current Practices and Future Development – Jan Gulliksen, Inger Boivie & Bengt Göransson (2006) Interacting with Computers, Vol. 18, No. 4, pp. 568-600.

The Lonesome Cowboy – A Study of the Usability Designer Role in Systems Development – Inger Boivie, Jan Gulliksen & Bengt Göransson (2006) Interacting with Computers, Vol. 18, No. 4, pp. 601-634.

Making a Difference – A Survey of the Usability Profession in Sweden – Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor & Lena Herulf (2004) In Hyrskykari A. (ed.), Proceedings of the 3rd Nordic Conference on Human Computer Interaction, NordiCHI 2004, Tampere, Finland, ACM Press, pp. 207-215

Basic values

Management Perspectives on Usability in a Public Authority – Åsa Cajander, Inger Boivie & Jan Gulliksen (2006) Proceedings of the 4th Nordic Conference on Human Computer Interaction, NordiCHI 2006, Oslo, Norway, ACM Press

Usability And User's Health Issues In Systems Development - Attitudes And Perspectives – Åsa Cajander, Inger Boivie and Jan Gulliksen (in press) In Effie Law, Ebba Hvanberg, Gilbert Cockton and Jean Vanderdonck (eds.) Maturing Usability: Quality in Software, Interaction and Value. Springer

SE and HCI

Human Centred Software Engineering: Integrating Usability In The Software Development Lifecycle – Ahmed Seffah & Jan Gulliksen, Michel Desmarais (2005) Kluwer Academic Publishers.

The Usability Design Process – Integrating User-Centered Systems Design In The Software Development Process – Bengt Göransson, Jan Gulliksen & Inger Boivie (2003). In the Special Issue on Bridging the Process and Practice Gaps Between Software Engineering and Human Computer Interaction edited by Rick Kazman and Len Bass. Software Process Improvement and Practice, Vol. 8, No. 2. pp. 111-131 Wiley

Usability Design—Extending Rational Unified Process With A New Discipline –Bengt Göransson, Magnus Lif & Jan Gulliksen. (2003) In LNCS volume 2844 "DSV-IS 2003: Issues in Designing New-generation Interactive Systems Proceedings of the Tenth Workshop on the Design, Specification and Verification of Interactive Systems" DSV-IS 2003.