

Mind the Gap!

Software Engineers and Interaction Architects

Matthias Müller-Prove

Sun Microsystems, Sachsenfeld 4, 20097 Hamburg, Germany

mprove@sun.com

Abstract: This paper points out the different mentalities of software engineers and interaction architects, and shows how these can have a negative impact on the communication in product teams.

Keywords: software engineering, human-computer interaction, heterogeneous product teams, project management

1 Introduction

Software engineers and interaction architects need to cooperate with each other in order to create software products that work, and that are usable and useful for the target audience. A look at reality shows that the cooperation does not function as smoothly as it should. The cause for this can be on the engineer's side, or on the designer's – or on both sides. This paper identifies some differences in the mentalities that make it difficult to work together in one product team.

It needs to be said that successful product teams have many more components than just engineering and user interface design. To use Don Norman's metaphor: it is technology, marketing, and user experience that make up the three legs of a solid product (Norman, 1999, pp. 40). We also have to add product management, quality management, documentation, customer support, and upper management to the picture. Nevertheless, this paper focuses only on the relation between developers and HCI professionals.

2 Software Engineers

Software engineers live in their own world.¹ With few exceptions, they only focus on computers and themselves. A problem is seen as solved as soon as the algorithm is correct and the compiler does not come back with any syntax errors. As far as usability

is concerned they are at best clueless, because usability was not part of their education. Many engineers fall into the trap of taking themselves as typical users by saying, "if I can use it then every other normal person can use it, too." – In fact, engineers are not normal or average people. Programming is a highly specialized profession that needs a lot of practice. The term software engineering reflects the scientific methods that are necessary to develop complex software systems. Sometimes software engineering even reaches the abstraction level of mathematics.

Engineers are the core members of a product development team. Nobody can do without them. And no one else is as difficult to replace. During the development process they collect so much detailed information about the product that it is even hard for another skilled engineer to continue the work at that point.

Sometimes engineers take advantage of their privileged position in order to steer the development process in a direction that seems most pleasant to them. They do not necessarily act intentionally. It is just too easy to become mixed up in this self-referential circle. Alan Cooper honored this position when naming his book *The Inmates Are Running the Asylum* (Cooper, 1999).

The attitude software engineers have is without a doubt fine in their own minds, and their standpoint is rarely challenged by other people in the product development process. Therefore, it is not surprising that engineers see no reason to take the suggestions of interaction designers seriously, since they usually just mean more work. Engineers block suggestions

¹ This section is based on *Usability im Unternehmen* (Müller-Prove, 2003, *Usability in Corporations*).

by responding that they are not feasible for technical reasons.

3 Interaction Architects

The term *interaction architect* was introduced by Bruce Tognazzini in one of his recent columns (Tognazzini, 2003). With this proposal he aims to give one unifying title to all those software designers, interaction engineers, human interface folks, user-experience flower-hoppers, or “whatever we are calling ourselves today.” Interaction architects are in charge of designing the gestalt of the software product. They need to be creative in order to find solutions for product requirements that conform to specific style guides. Interaction architects tend to perceive themselves as artists.

On the other hand, *usability professionals* are those HCI experts that test the products in usability labs, and use other means like expert evaluations. They take the mockups and prototypes and criticize the usability of the artifacts. According to Tognazzini, the “usability professionals should report their results to the design team only, and then it’s up to the design team how and when and where they want to incorporate those results (...)” (Dykstra-Erickson, 2000).

Now we are back at the point where interaction architects have to communicate with software engineers. For the interaction architect it is important to understand the engineer’s attitude in order to be able to cope with the situation. This is less complicated for usability experts that have their roots in computer science than for those who come from graphics design or psychology. The latter per se do not speak the same language as the engineers. This additionally widens the gap between those groups and causes the usability experts to not be treated as fully accepted members of the product team.

Interaction architects need competence and substantiality. They have to take into account that some colleagues have no conception of the techniques of usability engineering while at the same time maintaining their point of view. This is a challenge for interaction architects. They have to continuously propagate the benefits of usability. They also have to explain the way in which they come to their design proposals in order to demonstrate a professional attitude. Showing highlights from usability lab studies can open the eyes of engineers. Also, supporting findings with

quantitative data shifts the level of acceptance, because numbers are the native language of engineers.

A thorough approach is taken for instance by SAP (Latzina/Rummel, 2002). In-house training for engineers is conducted with the objective to build a conception of HCI methods. Role-plays are not just fun – they convince the engineers of the use of paper mockups for prototyping and other typical HCI techniques.

The conclusion is that it is necessary for both groups to gain respect for the different areas of expertise and have an understanding of each other’s perspective. Otherwise, the fruits of beneficial cooperation will be a long time coming.

About the Author

Matthias Müller-Prove is a computer scientist. His reputation as a human-computer interaction architect is based on ten years of direct experience designing professional software interfaces for international operating companies. Before joining the StarOffice User Experience team at Sun Microsystems in 2002, he played a significant role in designing the user interface of Adobe GoLive.

References

- Cooper, A. (1999). *The Inmates Are Running the Asylum*. Indianapolis: Macmillan Computer Publishing.
- Dykstra-Erickson, E. (2000). Interview with Bruce Tognazzini. *Interactions*, 7, (2, Mar.), 41-46.
- Latzina, M., & Rummel, B. (2002). Collaboration-Based Usability Training for Developers. In M. Herczeg, W. Prinz & H. Oberquelle (Eds.), *Mensch & Computer 2002* (pp. 285-291). Stuttgart: Teubner. http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2002/konferenzband/mc2002_05_paper/mc2002-26-latzinarummel.pdf
- Müller-Prove, M. (2003). Usability im Unternehmen. In S. Heinsen & P. Vogt (Eds.), *Usability praktisch umsetzen - Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*. München: Hanser.
- Norman, D. A. (1998). *The Invisible Computer*. Cambridge, MA: The MIT Press.
- Tognazzini, B. (2003). *It’s Time We Got Respect*. AskTog. <http://www.asktog.com/columns/057ItsTimeWeGotRespect.html>