

Are Engineers Condemned to Design? A Survey on Software Engineering and UI Design in Switzerland

Ljiljana Vukelja¹, Lothar Müller¹, Klaus Opwis²

¹ University of Applied Sciences Rapperswil, Switzerland
{lvukelja, lmueller}@hsr.ch

² University of Basel, Switzerland
klaus.opwis@unibas.ch

Abstract. In this paper we present the results of a descriptive online survey conducted among Swiss software developers regarding their engineering practices with a special focus on the design and development of user interfaces. This enables an insight into the everyday life of a software engineer and can lead usability practitioners, project managers and clients to a better level of cooperation in designing user interfaces through understanding how software engineers work. While software is developed and tested in a professional way, several problem areas were detected: firstly, software engineers frequently develop user interfaces alone, without the help of Human-Computer Interaction (HCI) professionals. Secondly, they have a limited knowledge of HCI. Thirdly, whilst they have contact to end users, they do not make use of this for user interface design. Finally, usability tests are rare and seldom result in big changes.

Keywords: Software engineering, user interface design, user-centered design, survey.

1 Introduction

Today, user-centered methods in software engineering are well developed and a wide range of literature on this subject has been produced (see e.g. [5-7, 11, 14, 18]. Nevertheless there are complaints about the usability of the products. As illustrated by Nelson et al. [21] poor usability leads to low levels of satisfaction amongst users. What then are the reasons for the apparent low quality of products? On one hand, the user-centered methods might not be good enough; on the other they might not be practiced correctly or even at all. We will, however, leave the discussion of the quality of the methods to other research, and instead will look into the practice of software engineering in Switzerland.

Most previous surveys have concentrated either on software engineering [9] or on Human-Computer Interaction practice [13, 16, 27], but rarely have they examined the relationship between the two. Jerome and Kazman [15] investigated this relationship by asking software (SW) engineers *and* HCI practitioners separately about each other,

reaching the conclusion that there are misunderstandings between the two groups. In our study we will concentrate *only* on the SW engineers and look at their everyday practice. We will describe what they do and how they do it, focusing in particular on the design and the development of user interfaces (UI).

The paper begins with a description of the parameters of our survey and a presentation of the results attained. This is followed by a detailed discussion. Finally, in concluding we will also consider possible further research.

2 Survey

Two key questions underpin the survey:

- *Which* software development methods are used?
- *Who* designs and develops user interfaces and how?

2.1 Method

We conducted a descriptive online survey among SW professionals in Switzerland during March and April 2006. The recipients were addressed through mailing lists and over personal contacts and were asked to fill in an online form. The demographics of addressed mailing lists and personal contacts are comparable, since we used general mailing lists of Swiss SW engineers and similar personal contacts. To avoid bias specialized mailing lists were excluded. Our target group was technically experienced and uses the Internet extensively, justifying an online survey [4]. No financial incentives were offered to those who took part though they were informed that they were contributing to a research project [28]. In order to increase the response rate we sent the survey as a personally addressed email invitation to our own contacts, since emails sent from a known sender are less likely to be tagged as spam [28]. Mailing list recipients were not addressed personally, but we assumed that members would have confidence in the quality of the mailing lists.

2.2 Basic Population and Sample

In 2003 in Switzerland there were 13,597 companies working in the area of computing services. These companies employed an average size of 4.9 employees [3]. This is equal to a total of some 67,174 employees in the area of information technology.

We emailed 1,874 addresses: 74 from personal contacts and 1,800 addresses selected from three mailing lists. Our target group was SW developers (and not HCI experts) in Switzerland, because we wanted to know how *SW developers* work. We received 134 completed forms.

Even though we addressed recipients belonging to general mailing lists on SW engineering there is possibly a correlation between participation and the questions posed in the survey (self-selection bias). It is possible that only those interested in UI design chose to answer. By asking the participants to refer to their current or last

project, we tried to avoid the problem of reporting the "general" feeling about how SW is or should be developed.

Due to the anonymity, it is not known whether some participants referred to the same project, so the number of projects covered may be less than 134. Only a systematic sample could avoid this.

2.3 Questions

The survey contained 25 questions and took an average of 30 minutes to complete. Firstly, we asked for information about the project participants were working on, their company, and the participants' activities within the project. The next section of questions addressed SW engineering methods and asked participants to describe the actual development process. This was followed by questions concerning requirements definition, change management, and testing habits. The next questions dealt with UI design and contact with end users. Finally, we asked the participants about their knowledge in the area of user-centered design.

In total the survey consisted of 13 open-ended questions where participants were asked to answer by writing free text, 3 yes/no questions and 7 questions where more than one answer could be ticked. The vast majority of questions were open-ended, the aim being to let respondents say what they wanted, rather than limiting them to a pre-selected (and possibly biased) choice of answers. We then categorized the responses for further analysis. Perhaps not surprisingly on account of the open-ended nature of some of the questions participants did not always answer these questions. The final two questions asked firstly, whether the participant was interested in the results of the survey and secondly, whether he or she would be willing to take part in an interview on the subject. Following a positive response the participant's identity was requested.

A copy of the survey questions can be attained from the authors.

3 Results

In this section, we will examine the results of the survey. The numbers in **boldface** are those emphasised in the text.

Note that the sample size of every question is different because, as alluded to above, not all of our 134 respondents answered all of the questions.

3.1 Respondents

The initial questions in the survey concerned the company where the participants work and the project in which they are involved. Table 1 displays the types of companies classified according to [1]. A company of micro size has less than 10 employees, a small company has 10 - 50, a medium one 50 - 250, and a large one more than 250 employees. We defined the size of the SW development department (last column in Table 1) to be ten times less in size than that of the overall company. Thus, a small SW development department has less than 5 employees, a medium one

has 5 - 25, and a large department employs more than 25 people. It can be seen that more than 42% of SW development departments are big. Our sample consists mostly of large companies and large SW development departments. This matches well with the countrywide distribution [2].

From our analysis 44.4% of respondents work on a proprietary product, 41.4% on a one-time project for external customers and 18.0% on a one-time project for internal customers. This suggests that a big portion of SW development is being conducted externally by specialized and professional providers.

Table 1. Company types.

Size	# of companies		# of SW development departments
	this survey	country average	
Micro	21 (22.6%)	842,657 (26.3%)	
Small	18 (19.4%)	680,728 (21.2%)	22 (25.9%)
Medium	22 (23.7%)	619,863 (19.3%)	27 (31.8%)
Large	32 (34.4%)	1,063,437 (33.2%)	36 (42.4%)
Total	93 (100%)	3,206,685 (100%)	85 (100%)

Table 2 shows the primary role of the participants along with other activities that participants undertake. In terms of activities the respondents were able to choose one or more answers from a predefined list, whereas for the primary role they were asked to explicitly name their role in free text. According to the responses received 70.1% conduct tasks of a developer. As their primary role 38.5% gave an answer of "developer", and 39.3% said they were "project managers". This confirms that we reached the main target of our study: the SW engineers.

Table 2. Tasks carried out by respondents.

Job	Primary role	Activities
Developer	47 (38.5%)	94 (70.1%)
Project manager (PM)	48 (39.3%)	64 (47.8%)
Software architect	8 (6.6%)	74 (55.2%)
Application tester	-	41 (30.6%)
Customer	-	7 (5.2%)
Requirements engineer (REn)	-	52 (38.8%)
Usability engineer (UEn)	3 (2.5%)	25 (18.7%)
Consultant	3 (2.5%)	-
Other	13 (10.7%)	-

An average team (see Fig. 1) consists of 7.8 people: 4.2 are developers, 0.5 have a profession dealing with the UI (REn, UEn).

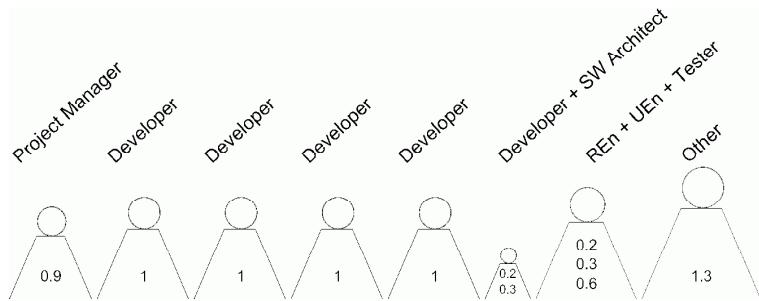


Fig. 1. Average team.

89.6% of teams have no requirements engineer and 85.7% of teams have no UI engineer. Note that in cases where no requirements engineer is present there still might be a UI engineer and vice versa. If we consider requirements engineers and UI engineers to be experts in HCI then from Table 3 it is apparent that 77.9% of SW teams work entirely without HCI professionals. In addition, we examined the data to see whether the size of SW development is correlated with the presence of requirements engineers or UI engineers. E.g., it might be that bigger SW development departments have HCI professionals, whereas smaller ones do not, or vice versa. However, χ^2 analysis of the difference between big and small/medium size SW development was not significant, $\chi^2(1, N = 77) = 1.164$, $p > .05$. I.e. the presence or absence of HCI experts in a team is independent of the size of SW development department.

Table 3. Presence of requirements engineers (REn) and UI engineers (UEn) in companies.

	Size of SW development		Total
	small or medium	big	
no UEn, no REn	37 (82.2%)	23 (71.9%)	60 (77.9%)
UEn or REn or both	8 (17.8%)	9 (28.1%)	17 (22.1%)
Total	45 (100%)	32 (100%)	77 (100%)

3.2 Software Engineering Methods

Software Engineering in General

We defined a software engineering method (SEM) as a "defined and predetermined proceeding in SW development". 63.9% answered that they use an SEM and 36.1% said they do not use an SEM. We wanted to know either which methods they use, or for them to describe freely their procedure of SW development in cases where they stated that they do not use an SEM. Several participants first stated that they do not use an SEM, but in their description of the SW development process we could nevertheless recognize a method. We found 2 agile methods and 21 answers conformed to our definition of other methods (mostly company-defined). As a result we also counted these answers as having an SEM. The results are shown in Fig. 2.

18.8% use no method. Among the SEMs the Rational Unified Process (RUP) is most popular, followed by agile and heavy-weighted (e.g. Hermes, V-model, waterfall) methods. 36.8% use a combination of company-intern or other methods. 1 answer could not be categorized.

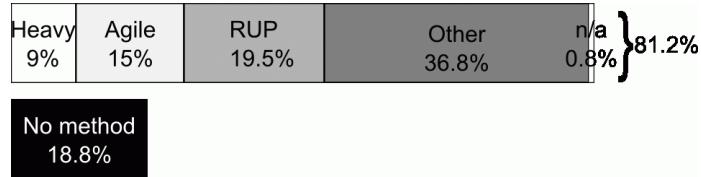


Fig. 2. Software engineering methods.

Testing

According to the results of our survey SW seems to be tested thoroughly and in a serious manner: in 87.3% of cases not only developers test the SW, but also testers, employees in quality management, project managers, those in marketing, business analysts or end users (see Fig. 3a). In 12.7% of cases only the developers do the tests. Modules and systems are both being tested: modules in 76.2% of cases, systems in 98.1% of cases. By testing modules we mean testing the internal correctness, completeness, security and quality of each building block of the system. Testing systems denotes integration tests and functional tests of the complete system. As shown in Fig. 3b tests are mostly (in 77.1% of cases) conducted both parallel to, and at the end of, the development.

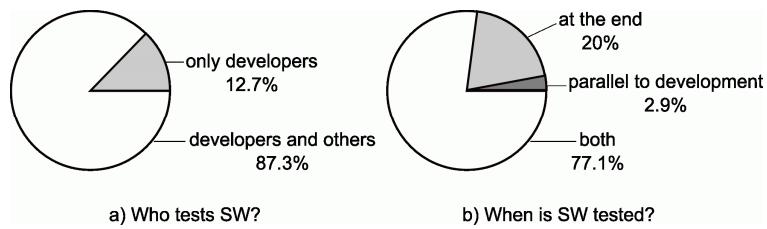


Fig. 3. Testing.

Software Engineering and End Users

A large part of our survey was dedicated to gaining an insight into the relationship of the participants with the end users. Using free text participants were asked to describe how their company ensures that the needs of the end users are met. We analyzed the results and categorized them according to which measures are being taken and who is involved. Table 4 shows the means by which companies try to ensure that their products meet end user needs. These include, among others, involvement for defining requirements and feedback during implementation. The last column shows the overall distribution of the various means. Note that more than one answer could be given. The other columns of Table 4 categorize the answers as to whether the end user or the

customer is involved. Customers buy the product; end users finally use the product after it has been bought.

The numbers shown as "col%" give the distribution within each column. They add up to more than 100% due to multiple selections. When customers are involved to ensure end users' needs (first column) they mostly give feedback during implementation (38.2%), when end users are involved (second column) this mostly concerns requirements (50.0%).

The numbers shown as "row%" give the distribution within each row. In the activity *requirements* customers are involved in 21.8% of cases and end users in 43.6%. This is a positive result as it is in accordance with user-centered design. However, the fact that in 38.5% of cases the acceptance test is only carried out by the customer is alarming.

Table 4. Ensuring end users' needs.

	Who is involved?			Total
	Customer col% row%	End user col% row%	Unknown col% row%	col%
Requirements	12 35.3% 21.8%	24 50.0% 43.6%	19 45.2% 34.5%	55 44.4%
Feedback during implementation	13 38.2% 32.5%	19 39.6% 47.5%	8 19.0% 20.0%	40 32.3%
General communication	7 20.6% 33.3%	9 18.8% 42.9%	5 11.9% 23.8%	21 16.9%
Usability Tests, Reviews w. EU		15 31.3% 100.0%		15 12.1%
Acceptance Test, Feedback to Application	5 14.7% 38.5%	4 8.3% 30.8%	4 9.5% 30.8%	13 10.5%
Bugs, Change Requests	5 14.7% 35.7%	5 10.4% 35.7%	4 9.5% 28.6%	14 11.3%
Not provided			1 2.4% 100.0%	1 0.8%
Unknown	1 2.9% 7.7%		12 28.6% 92.3%	13 10.5%
Total	34 27.4%	48 38.7%	42 33.9%	124

Problem reports

Problem reports in 71.4% of cases are handled through a tool, a dedicated person or helpdesk, and in 28.6% of cases in an informal way.

End User Documentation

Documentation for the end user is written in 34.2% of cases in parallel to and at the end of development and in 65.8% only at the end. 75 free text answers concerning who writes the documentation were categorized according to the professionalism of the documentation process as being:

- "high" when the documentation is produced by a documentation or support department, or a usability professional (25.3%)
- "medium" when it is produced by the developer and/or tester (53.3%)
- "low" when it is produced by the end user, the customer or nobody. (21.3%)

3.3 Direct Contact with End Users

One of the questions asked whether participants had direct contact with end users. The answers revealed that 71.5% have contact with end users but only 27.3% of all respondents utilize this for requirements and/or GUI design. Among the 22 who do not have direct contact with end users 14 think it would be useful to have such contact.

Table 5. Developers' contact with end users.

Contact with end users			
yes	no		
55 (71.5%)	22 (28.6%)		
Used for...	Would be useful?		
Requirements and/or GUI Design	21 (27.3%)	useful	14 (18.2%)
Support, Testing Problemreports, Education	12 (15.6%)	not useful	5 (6.5%)
n/a	22 (28.6%)	n/a	3 (3.9%)

3.4 Requirements

We also posed several questions concerning requirements: who defines them and who requests changes. The answers were, however, difficult to interpret as they did not differentiate the sample well, so we decided to omit them here. A discussion of requirements engineering is presented in [20].

3.5 User Interface Development and Testing

UI Development

The following are involved in the design of the UI: developers in 79.2% of cases, end users in 26.2% of cases, requirements engineers in 25.4%, usability engineers in 20.0%, interaction designers in 14.6%, and marketing personnel in 12.3%. In 11.5% of cases the UI is not explicitly designed, but rather emerges as a byproduct. It should be noted that more than one answer could be given. We analyzed the data in more depth to understand the roles involved in the design and the participation of the end users. Fig. 4 shows the results. The cases where the UI *only* emerges as a byproduct were left out from the analysis. We can see that in 46.8% of cases developers design the UI alone without help of any kind from co-workers who can be classified as usability professionals, namely interaction designers (ID), requirements engineers (RE), usability engineers (UE), or marketing workers (M).

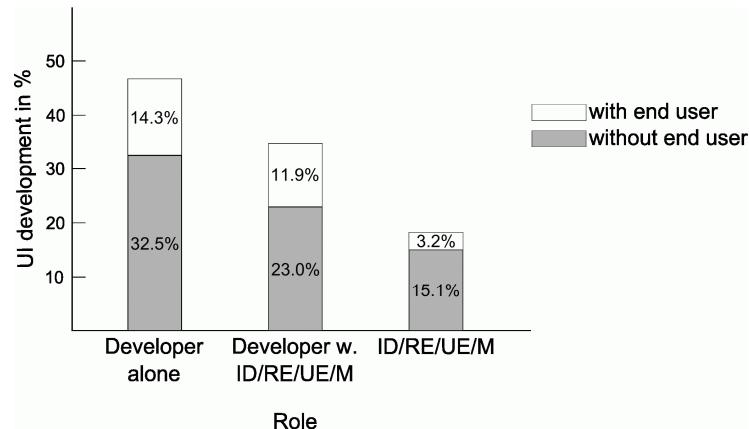


Fig. 4. User interface development.

Knowledge in HCI

As our focus is on the connection between SW development and HCI, we analyzed in depth 84 participants who stated that they are SW developers (as one of their tasks) and gave information about their knowledge in HCI. These respondents gave as sources of their knowledge in HCI the answers displayed in Table 6. The quantity of HCI knowledge was estimated based on free text answers to two questions: one concerning the sources of their HCI knowledge, the other concerning the books they know in this area. For example, a participant who knew many books was rated as "high"; a participant who was not aware of any books in HCI, but declared to have gained HCI knowledge through experience was given the rating "low". *Experience* is defined as learning on the job, *study* as full time education or courses, and *autodidactic* as reading books and articles on the subject. Note that several participants gained their knowledge from more than one source. It is alarming that only 8.3% of respondents have a high quantity of knowledge in HCI.

Table 6. Quantity and sources of HCI knowledge.

Source	Quantity			Total
	high	low	none	
Experience	2 (2.4%)	23 (27.4%)		25 (29.8%)
Study	4 (4.8%)	25 (29.8%)		29 (34.5%)
Autodidactic	2 (2.4%)	17 (20.2%)		19 (22.6%)
None			24 (28.6%)	24 (28.6%)
Total	7 (8.3%)	53 (63.1%)	24 (28.6%)	84 (100%)

Usability Tests

Regarding usability tests (UTs), we received 132 answers. 37.9% do conduct UTs, 62.1% do not. In only 9.1% UTs are carried out *and* result in big changes. In 40.9% UTs are *not* carried out, but are considered useful, in 7.6% they are *not* carried out and are considered useless.

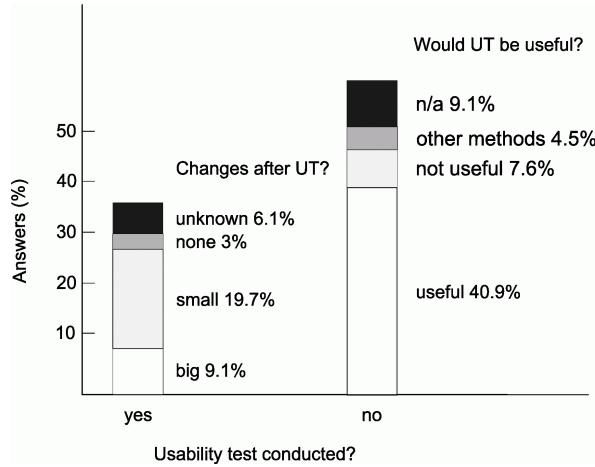


Fig. 5. Usability tests.

4 Discussion

In the previous section we presented the data we gathered during our survey. Some results were as expected, some, however, were surprising. In this section we will discuss our findings as well as compare them with previous work.

Software Engineering

SW development is carried out in a professional way, as can be seen in the widespread use of SW engineering methods. SW engineering teams use mostly RUP or company defined methods, which is in accordance with [13, 20]. Agile methods play a minor role.

Further indicators of professional SW development are the handling of problem reports where tools and defined processes are used, and testing, which is carried out thoroughly with module and system tests.

A weaker point of the SW engineering process is documentation: end user documentation is only written in one third of cases parallel to development and in one fourth of cases by professionals.

Testing is focused on the functional aspect of the SW, reflected by the module and system tests, whereas the usability aspect is greatly ignored: usability tests are conducted in less than 40% of cases and cause bigger changes in less than 10% of cases. Therefore, the often voiced complaint about poor usability is not surprising, since the perspective of the end user is missing in the testing efforts.

Involvement of End Users

Most participants reported contact with end users (71.5%), but only 27.3% reported using it for requirements gathering or UI design. Here too, the perspective of the end user is missing.

SW Engineers and UI Design

Our survey shows that the UIs are designed by the SW engineers: in 46.8% of cases they do this alone. As we have seen the quality of education in HCI and the amount of knowledge about this area are rather low among SW engineers. Earlier research [12] allowed SW engineers to rate their knowledge in Human Factors. 49% of SW engineers rated themselves as "expert" or "good". Our results suggest, however, that in this self-rating SW engineers overestimated the extent of their knowledge. Poorly designed UIs and low usability of products therefore come as no surprise.

Why do the SW engineers design the UI alone? Why do they not ask experts for help? Is it that there are not enough experts or rather that SW engineers do not seek contact with them? [12] report that in large companies which employ Human Factors professionals, SW engineers

- encounter Human Factors problems in 92% of cases
- have access to Human Factors experts in 21% of cases
- contact these experts and get advice from them in 12% of cases
(in fact most advice comes from other SW engineers)
- consider this advice helpful in 52% of cases

In companies which do not employ Human Factors experts these numbers can be expected to be much worse. Probably organizational or cultural changes are necessary to improve access to, and acceptance of, Human Factors experts.

[23] lists a whole range of human-centered design skills needed by SW engineers in order to produce good UIs. In our survey we did not go into this level of detail, but if we assume that what we labeled "high quantity of knowledge" corresponds to the skills stated in [23], then only 8.3% of the developers are able to produce UIs of good quality.

On the other hand, according to [8], SW development is a very engaging activity and we cannot expect programmers to do the job of HCI experts in addition to their main activity of SW engineering. [24] states in the introduction: "While software developers may have high-level familiarity with such basic concepts as requirements

analysis and usability testing, few understand the complete process well enough to incorporate it into the larger software development lifecycle."

Role of User-Centered Design

According to the results of our survey user-centered design (UCD) does not play a prominent role in current SW engineering practice in Switzerland. None of the user-centered methods was named as a SW engineering method and in the descriptions of the development process only a few facets of UCD were mentioned.

User-centered design focuses on knowing the end users, gathering their requirements and explicitly designing the interaction. In only 22.1% of cases there are requirements engineers and/or interaction designers within the development team, so that some of these activities are carried out by experts, otherwise they are carried out by the developers.

We have already mentioned that only 27.3% of the developers use their end user contact for requirements gathering and/or UI design. But without good knowledge of how to gather requirements from end users and how to involve end users in interaction design, their participation may even have a negative effect. Most end users simply see their own work activities, but lack a broader view of the complete workflow [14, 26]. Given today's fast changing technologies they are unable to find the best solution. They are neither designers nor usability professionals.

But we also found some positive results indicating hope for improvement: two thirds of developers with no end user contact say that it would be useful, 40.9% of cases with no usability testing consider it useful. In these cases developers would accept or even welcome a better incorporation of end users in the development process. SW engineering methods are widely practiced and are adopted to specific company and project needs: they could be easily augmented with user-centered elements.

5 Conclusion

By conducting the survey we found that SW engineering in Switzerland is practiced professionally. With respect to UI design we encountered the following major problems:

1. UIs are designed by SW engineers: in 46.8% they do this alone and the majority of teams (77.9%) contain no UI specialist
2. poor knowledge of SW engineers: only 8.3% have high HCI knowledge
3. insufficient involvement of end users: of the 71.5% of engineers with end user contact only 27.3% use it for requirements or UI design; usability tests are rarely conducted

We did not investigate the reasons for these problems, but believe them to be due to poor awareness of the importance of UIs among engineers and management and a lack of education in this field. As a consequence, user-centered design is not integrated into the SW engineering methods used.

How could the situation be improved in the future? We believe that a set of measures is necessary which address the problems discussed:

- Return of Investment calculations to prove the usefulness of efforts on good UI design [17]¹
- a better, possibly mandatory education in HCI for engineers, including user-centered design to improve their awareness and their ability [10]
- integration of user-centered design into SW engineering methods to ease the necessary cultural and organizational change [25]

Acting on all these levels simultaneously would make SW products more user-friendly.

Acknowledgments. We thank Peter Schmutz and Robbie Aitken for tool support and proofreading.

References

1. Sme definition.
http://ec.europa.eu/enterprise/enterprise_policy/sme_definition/index_en.htm, accessed on January 22, 2007.
2. Unternehmen, kennzahlen. <http://www.kmu.admin.ch/kmu/00478/index.html?lang=de>, accessed on January 22, 2007.
3. Unternehmensdemografie daten 2003 bestellnummer 727-0300.
<http://www.statistik.admin.ch>, accessed on January 22, 2007.
4. B. Batinic, U.-D. Reips, and Bosnjak M. Online Social Sciences. The Morgan Kaufmann Series in Interactive Technologies. Hogrefe and Huber Publishers, 2002.
5. Hugh Beyer and Karen Holtzblatt. Contextual design: defining customer-centered systems. Morgan Kaufmann Publishers Inc., 1998.
6. L. L. Constantine and L. A. D. Lockwood. Software for Use. Addison Wesley Longman, Inc., 2000.
7. A. Cooper and Reimann R. About Face 2.0. Wiley Publishing, Inc., 2003.
8. Alan Cooper. The Inmates Are Running the Asylum. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 1999. Foreword By-Paul Saffo.
9. Bill Curtis, Herb Krasner, and Neil Iscoe. A field study of the software design process for large systems. Commun. ACM, 31(11):1268–1287, 1988.
10. Gesellschaft für Informatik. Curriculum für ein basismodul zur mensch-computer-interaktion.
11. John D. Gould and Clayton Lewis. Designing for usability: key principles and what designers think. Commun. ACM, 28(3):300–311, 1985.
12. J. Grudin and S. E. Poltrock. User interface design in large corporations: coordination and communication across disciplines. SIGCHI Bull., 20(SI):197–203.
13. Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor, and Lena Herulf. Making a difference: a survey of the usability profession in sweden. In NordiCHI ’04: Proceedings of the third Nordic conference on Human-computer interaction, pages 207–215, New York, NY, USA, 2004. ACM Press.
14. K. Holtzblatt, Burns Wendell J., and Wood S. Rapid Contextual Design. The Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann Publishers, 2005.
15. B. Jerome and R. Kazman. Surveying the solitudes: An investigation into the relationships between human computer interaction and software engineering in practice. Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, 2005.

¹ Instead of ROI one can also consider TCO [22] or the Kano model [19].

16. Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. User-centered design methods in practice: a survey of the state of the art. In CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research, page 12. IBM Press, 2001.
17. Aaron Marcus. Return on investment for usable user-interface design: Examples and statistics.
18. Deborah J. Mayhew. The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design (The Morgan Kaufmann Series in Interactive Technologies). Morgan Kaufmann, April 1999.
19. F. Takahashi N. Kano, N. Seraku and S.i. Tsuji. Attractive quality and must-be quality. *The Journal of the Japanese Society for Quality Control*, 14(2):39–48, 1984.
20. Colin J. Neill and Phillip A. Laplante. Requirements engineering: The state of the practice. *IEEE Softw.*, 20(6):40–45, 2003.
21. R. Ryan Nelson, Michael W. Kattan, and Paul H. Cheney. Training, ability, and the acceptance of information technology: an empirical study of is personnel and end users. *SIGCPR Comput. Pers.*, 13(3):20–32, 1991.
22. Daniel Rosenberg. The myths of usability roi. *interactions*, 11(5):22–29, 2004.
24. Ahmed Seffah. Learning the ropes: human-centered design skills and patterns for software engineers' education. *interactions*, 10(5):36–45, 2003.
23. Ahmed Seffah. Learning the ropes: human-centered design skills and patterns for software engineers' education. *interactions*, 10(5):36–45, 2003.
24. Ahmed Seffah, Jan Gulliksen, and Michel C. Desmarais. Human-Centered Software Engineering: Integrating Usability and Software Development Lifecycle, volume 8 of Human-Computer Interaction Series. Springer, 2005.
25. Kenia Sousa, Elizabeth Furtado, and Hildeberto Mendonça. Upi: a software development process aiming at usability, productivity and integration. In CLIHC '05: Proceedings of the 2005 Latin American conference on Human-computer interaction, pages 76–87, New York, NY, USA, 2005. ACM Press.
26. Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. Getting the right design and the design right. In CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1243–1252, New York, NY, USA, 2006. ACM Press.
27. Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 471–478, New York, NY, USA, 2002. ACM Press.
28. Bandilla W. and Bosnjak M. Querschnitt – Festschrift für Max Kaase, chapter Online Surveys als Herausforderung für die Umfrageforschung: Chancen und Probleme. Peter Ph. Mohler und Paul Lüttiger, 2000.