Increasing Empathy toward Users' Needs by using Usability Evaluations: A field-experiment

Conference Paper · August 2014

Fulvio Lizano
National University of Costa Rica

Increasing Empathy toward Users' Needs by using Usability Evalulations: A field-experiment.

Fulvio Lizano (corresponding author)

Informatics School, National University, Heredia, Costa Rica

flizano@una.cr (corresponding author)

Jan Stage

Department of Computer Science, Aalborg University, Aalborg, Denmark
jans@cs.aau.dk

Abstract. Although HCI techniques as usability evaluations are considered strategic in software development, there are important obstacles in the application thereof. One of the most relevant is the software developers' lack of understanding regarding usability. This problem is connected to the lack of empathy toward users' needs on the part of software developers. Empathy is strategic in order to handle this and other problems presented in both HCI and SE. This paper presents results from an empirical study in which we explored the effectiveness of some typical usability evaluation methods to improve of empathy towards users' needs. The results show that the classical, user-based thinking aloud protocol significantly increases empathy. The alternative remote synchronous method is also effective and is more efficient, considering its logistical and economic advantages. Both methods create a scenario in which users work in conditions that are more realistic and expose their implicit needs while being observed by developers. This interaction allows the contagion of users' emotions by software developers, which increases their empathy toward users' needs.

Keywords: Empathy towards users' needs; understanding regarding usability; emotional contagion; usability evaluation.

1 Introduction

There is clarity regarding the relevance of a high level of usability in software. However, many software developers reveal contradictory behavior in that do not give priority to aspects of usability. This lack of priority is rooted in the different aims and motivations of SE and HCI practitioners (Lee 2006, Sohaib and Khan 2010). It is also possible to link this issue to what some researchers have called the software developers' mindset (Bak et al. 2008, Ardito et al. 2011). In the particular case of usability evaluations, the developers' lack of priority is related to a lack of understanding when it comes to the concept of usability concept (Rosenbaum 2000). This problem has become a serious

obstacle to applying usability in the software development process (Bak et al. 2008, Ardito et al. 2011, Seffah, A., and Metzker 2004).
Recent studies have suggested that involving software developers in the conduction or observation of usability evaluations with users could increase their understanding of usability (Bruun, A., and Stage 2012, Hoegh et al. 2006, Skov and Stage 2012). In parallel with this, it is possible to identify an improvement in empathy towards users' needs (Hoegh et al. 2006).
A clear understanding of reasons why empathy increases in a usability evaluation context is not trivial. This paper presents the results of an empirical study that aimed to explore the effectiveness of several typical usability evaluation methods to improve or increase the software developers' empathy towards users' needs. In the following sections, we offer an overview of related works, the method used in our research, the results of our study, the analysis and conclussions.

2 Related work

Empathy is the capacity of an observer to observe, identify, and understand another person's feelings, producing in the observer, at least partially, a sense of sharing the same feelings as the other person (Decety and Jackson 2006, Singer and Lamm 2009). Empathy occurs in the context of an emotional contagion processes (De Vignemont 2004, Singer and Lamm 2009).
Emotional Contagion (EC) theory has been extensively used in group psychology. According to EC theory, some of the key points presented in the contagion process are the tendency to automatically mimic and synchronize expressions, vocalizations, postures and movement, converging emotionally (Hatfield et al. 1994) and creating an unconscious induction of emotional states and behavioral attitudes (Schoenewolf 1990). This process involves the perception of emotions using nonverbal signals, such as facial expressions, body language and tone rather than words (Barsade 2002).
Several behavioral processes are connected to EC theory. Normally, these processes are inherent in social influences and could occur both consciously and subconsciously (Barsade 2002). The observer assumes a submissive role in her/his interaction with the observed who, in turn, assumes a dominant role. The particular circumstances or personalities of each are decisive in establishing who assumes a particular role (Schoenewolf 1990). In this process, emotions may be passed to others in order to influence not only their emotions but also their perceptions of other aspects. For example, Pugh (2001) argues that when employees display positive empathy towards customers, the customers perceive more quality in the service obtained.
Empathy and EC theory have been considered in HCI research (Mattelmäki and Battarbee 2002; Yammiyavar 2005; Preece 2001). Empathy towards users on the part of software developers has also been part of the research agenda in HCI and SE. Gilmore and Velázquez (2000) found that developers' participation in user experiences is important in order to improve the software developer's empathy for users. Newell et al. (2006) also reported the relevance of the generation of empathy with users during the process of interface design. Furthermore, empathy is relevant in software development in general (Grudin 1991; Patton 2002; Karn et al. 2007).
Despite the aforementioned studies, there is a gap in the literature when it comes to training-based strategies to improve software developers' empathy towards users' needs. Some HCI training proposals (LeBlanc et al. 2006, Lunt et al. 2008) do not consider empathy toward users' needs. A marginal reference regarding some efforts to increase the empathy of software engineering students was provided by Shaw and Dermoudy (2005) who questioned the effectiveness of traditional approaches based on lectures to increase empathy.
An alternative approach, based on software developers participating in or observing usability evaluations, has emerged in some recent studies. Hoegh et al. (2006) found an improvement in the

lack of understanding regarding usability and a parallel increase in empathy towards users' needs by software developers who observed usability evaluations. Gilmore and Velázquez (2000) also argued the relevance of interaction with users in order to increase empathy towards them. The feasibility of conducting usability evaluations by developers was proved by Skov and Stage (2012). Bruun and Stage (2012) argued that, with only a few hours of training, software practitioners could be taught to identify usability problems.

Many of the usability evaluation methods used in these studies correspond to classical methods that are regularly used, such as conventional thinking aloud protocol (Rubin and Chisnell 2008), inspection methods like heuristic evaluation (Nielsen and Molich 1990), and methods that are more modern, like the remote synchronous testing model (Andreasen et al 2007).

Despite these research efforts, no specific studies have been conducted on exploring the effectiveness of these typical usability evaluation methods in order to improve or increase the software developers' empathy towards users' needs.

3 Method

We conducted a field experiment aimed at exploring the effectiveness of several usability evaluation methods in order to improve or increase the software developers' empathy towards users' needs.

We used a between-group design with four conditions corresponding to four types of usability evaluation:

- Classical inspection method based on heuristic methods (hereinafter referred to as Heuristic).
- Variation of the inspection method with "supervision" (hereinafter referred to as Heuristic-Supervised). We introduced this variation in order to provide additional support to evaluators and to compensate for support received by participants in other conditions.
- Classical laboratory-based think-aloud method (hereinafter referred to as Lab).
- Modern remote synchronous testing method (hereinafter referred to as Remote).

3.1 Participants

A total of 36 people participated in our study. Participants were advanced System Engineering students who were organized in 12 teams of 2-4 members. The average age was 22.2 (SD =2.17) and 11% were female. In addition to the courses taken previously, the participants had amassed nearly 18 months of real experience of practical academic activity by developing a software system in a real organization, with real users and their needs. The scope of all these software projects were carefully controlled in order to guarantee similar characteristics in quality/size/complexity.

Of the nearly 30 existing projects, 16were pre-selected as potential participants in the experiment. These pre-selected projects belonged to organizations which could have at least 3 potential users available and were willing to participate in the study. After that, we randomly selected the number of organizations and participants needed for the experiment. Finally, again by using a random distribution, we grouped the actors into the different conditions used in the experiment. In each condition, each team evaluated another team's software. The software project were developed in diverse organizations (e.g., schools, colleges, biological research organizations, municipal police stations, etc.). As an incentive for participation, the participants received extra credits.

3.2 Training

All participants received training and advice from the researchers of this study during the experiments (the Remote condition's participants were contacted in a remote way). The training included a workshop by developing experiences such as planning, conducting and reporting usability evaluations. Participants received specific instructions in order to consider three categories of the usability problems identified: critical, serious, and cosmetic (Andreasen et al. 2007). The training was based on the forms and guidelines defined for the study (Nielsen and Molich 1990, Kjeldskov et al 2004, Rubin and Chisnell 2008). The number of hours spent in training was 10.

3.3 Procedure

We conducted two data collections (1DC and 2DC). 1DC was conducted approximately one month before the study. Following this, the participants worked in teams to design and conduct the usability evaluations according to the corresponding conditions. We separated each test in two main parts. The first part, under the responsibility of the team who made the software, corresponded to the planning of the complete process. The second part was the conduction of the test itself under the responsibility of another team. In this way, we guaranteed that all the teams participated impartially enough in a comprehensive testing process. In Table 1 we present the settings and specific procedure followed in each condition. The second data collection, 2DC, was conducted immediately after the finalization of the tests. In addition, we held four focus group sessions (1 per condition) which were audio-visually recorded in order to transcribe and analyze them later.

Condition	*Settings*	*Procedure*
Heuristic	• Classical heuristic evaluation (Nielsen and Molich 1990)	• Preliminary individual assessment. • Final Result group
Heuristic-Supervised	• Similar to Heuristic condition.	• Similar to Heuristic condition. • Heuristics' interpretation support by "supervisor"
Lab	• State-of the-art usability lab. • Conventional thinking aloud protocol (Rubin and Chisnell 2008)	• 3 Users sessions/5 Tasks each • Test-Monitor sat next to user. Logger and observers took notes. • Audio-Visual recording. • Final analysis session (Kjeldskov 2004)
Remote	• Remote synchronous testing (Andreasen 2007) • All participants separated spatially.	• Similar to Lab condition.

Table 1: Settings and procedure used in the conditions

Usability evaluations was only to set an environment where the participant could interact or not, with users in a more realistic context. In this paper We do not include unnecessary details about how the usability evaluations where conducted.

3.4 Data collection and analysis

Data collection in this study was focused on the participants' opinion about their software. Our intention was to identify if such opinions were connected or not, to usability matters in order to explore the participants' level of understanding of usability. Any change in this understanding implies an improvement in empathy towards users' needs (Gilmore and Velázquez 2000, Hoegh et al. 2006).
We used two forms in each data collection. The first form had the aim of allowing the participants to express their opinions related to the main strong and weak points of the software. This form was coded as 1DC-F1 and 2DC-F1, respectively. The second form had the goal of measuring the relative importance given by the participants to certain software/usability concepts. We measured the relative importance by using several pairs of sentences that could illustrate several activities/concepts related to HCI or SE.
We conducted our analysis of the first form following a three-step procedure. First, the concepts contained in the forms were reviewed and clarified. Second, we coded the concepts related to usability matters. Finally, the remaining concepts were coded as technical aspects related to software. With this, we analyze variations in the participants' understanding regarding usability between 1DC and 2DC. In the second form we calculate the level of relative importance reported by each participant. We analized this results individually in order to triangulate with the previous concepts given on the other forms. Thus, we have identified and measured the improvement of the participants' understanding regarding usability. We also used independent-sample t tests and paired-sample t tests. Finally, the data collected during the focus group sessions were quantified by using basic principles of the grounded theory approach by Strauss and Corbin (Strauss and Corbin 1998).

4 Results

4.1 Initial empathy status: initial overall understanding of usability

When the study began, we were interested in gauging the perceptions of participants of all conditions regarding their understanding of usability. This understanding is directly related to the use of concepts connected to usability or software. In Table 2, we present the general results obtained. First, we present the results obtained when we enquired about the strong (ST) and the weak (W) points of the software (F1). We organized these opinions into two categories: strong points and weak points. In each category, we present the results obtained for the points related to usability (U) and for the points related to software (S). In both cases, these are expressed in terms of quantity and percentage (#/%). In addition, we present the relative importance that participants placed on usability/software matters (F2) in terms of percentages (%).

Condition	*F1*				*F2*	
	Strong points (#/%)		*Weak points (#/%)*		*(%)*	*(%)*
	U	*S*	*U*	*S*	*U*	*S*
Heuristic	13 38%	21 62%	3 14%	19 86%	34%	66%
Heuristic-Supervised	6 19%	26 81%	3 13%	20 87%	29%	71%
Lab	5 15%	29 85%	7 30%	19 73%	27%	73%
Remote	13 35%	24 65%	7 29%	18 71%	47%	53%

Table 2: Results at 1DC. F1: Strong / weak points related to usability (U) / software (S) matters. F2: relative importance

The results regarding relative importance confirmed the perception of the strong and the weak points of the software. Initially, the participants' opinions were mainly focused on technical software matters.

In general, usability matters were considered only marginally by participants. This situation is interesting, considering that, at the time of the study, the participants had had contact with users for almost 18 months and that they had pursued numerous computing courses for seven semesters. This last factor explains their preference for software matters.

Considering our interest in usability, the independent-sample t test suggests that there is no significant difference between strong (ST) and weak (W) points for almost all conditions, except in the cases of the strong points of Heuristic versus Lab and for the strong points of Lab versus Remote (see Table 3 for the results of these tests).

	Heuristic	*Heuristic-Supervised*	*Lab*	*Remote*
Heuristic		(ST) p=.056	(ST) (p=.048	(ST) p=1.000
Heuristic-Supervised	(W) p=1.000		(ST) p=.806	(ST) p=.056
Lab	(W) p=.129	(W) p=.129		(ST) (p=0.48)
Remote	(W) p=.129	(W) p=.129	(W) p=1.00	

Table 3: Independent-sample t test, df=16 for strong (ST) / weak (W) points related to usability, at 1DC. (p) =significant

In general, it is possible to affirm that the participants of all conditions had a similar overall understanding of usability before the usability evaluations. The results reveal that, before the evaluations, the understanding of usability was relatively the same for all participants in our study, which is logical considering that they have the same background (training, courses pursed, contact with users). Similarly, considering the relationship between understanding and empathy (see the related work section), these results suggest that at the beginning of the study the participants had a similar level of empathy towards users' needs.

4.2 Change in the empathy towards users' needs: Improvement of understanding of usability

After the usability evaluations, we identified changes in the understanding of usability. The intensity of these changes was variable in the different conditions. Table 4 presents the general results obtained in 2DC in the four conditions and the variations with respect to 1DC. In this table, we present only the results related to usability matters. First, we present the strong and weak points (F1) and the results obtained at 2DC for the strong points expressed in terms of quantity and percentage (#/%). Next, we present the variation of this aspect with respect to 1DC (expressed in terms of a percentage - %). Thereafter, we present the results obtained for the weak points, which are expressed in terms of quantity and percentage (#/%). Finally, we present the variation of the weak points with respect to 1DC (expressed in terms of a percentage - %). In addition, we present the relative importance (F2). First is the percentage obtained at 2DC and the variation with respect to 1DC (in both cases, this is expressed in terms of percentages - %).

Condition	*F1*				*F2*	
	Strong. (#/%)	*Var. (%)*	*Weak (#/%)*	*Var. (%)*	*(%)*	*Var. (%)*
Heuristic	10 35%	-3%	3 16%	+2%	44%	+10 %
Heuristic-Supervised	11 38%	+19 %	8 30%	+17 %	42%	+13 %
Lab	25 71%	+56 %	12 50%	+23 %	60%	+33 %
Remote	27 63%	+28 %	9 50%	+21 %	68%	+21 %

Table 4: Results at 2DC. F1: Strong / weak points related to usability (results and variation respect 1DC). F2: relative importance of usability (results and variation respect 1DC).

These results allow the identification of an improvement in the understanding of usability after the usability evaluations. This improvement is more evident in Lab and Remote conditions. After evaluations, the participants of such conditions had a different perception of the relative importance of usability in respect of software matters (33% and 21% of increment respectively for Lab and R). These results were confirmed by the perception regarding the strong and weak points related to usability, at 2DC. The participants of Lab condition reported 35 strong points, of which 71% were related to usability (56% more than 1DC). The same participants also increased their perception of weak points related to usability (50%, 23% more than 1DC). For Remote condition, it is possible to observe the same situation: 63% strong points (28% more than 1DC) and 50% weak points (23% more than 1DC).

In order to confirm if the improvement in the understanding of usability was only presented in Lab and Remote conditions, paired-sample t tests were made in order to identify significant differences between 1DC and 2DC, for strong and weak points in the four conditions. Significant differences were detected for strong points at Lab and Remote conditions (see Table 5).

Condition	*Strong points*	*Weak points*
Heuristic	p=.282	p=1.000
Heuristic-Supervised	p=.139	p=.051
Lab	(p=.002)	p=.247
Remote	(p=.011)	p=.512

Table 5. Paired-sample t test, df=8 for strong and weak points (1DC vs. 2DC). (p) =significant.

In addition, the independent-sample t test suggests that, at 2DC, there is a significant difference between the strong points for Lab-Remote conditions when compared to Heuristic with Heuristic-Supervised conditions. The same situation occurred with the weak points of Lab condition versus Heuristic condition. Finally, no significant differences were detected between the strong or the weak points for Lab versus Remote conditions, and neither between the strong or the weak points for Heuristic versus Heuristic-Supervised conditions (see Table 6). This absence of significant differences confirms that Lab and Remote conditions changed at the same time. Final results in these conditions (at 2DC) confirm a change of a similar magnitude. The same parallelism is present in Heuristic and Heuristic-Supervised conditions, with the difference that no significant difference at 2DC means that no changes occurred in either condition.

	Heuristic	*Heuristic-Supervised*	*Lab*	*Remote*
Heuristic		(ST) p=.807	(ST) (p=.002)	(ST) (p=.003)
Heuristic-Supervised	(W) p=.091		(ST) (p=.008)	(ST) (p=.009)
Lab	(W) (p=.026)	(W) p=.343		(ST) p=.714
Remote	(W) p=.063	(W) p=.779	(W) p=.490	

Table 6: Independent-sample t test, df=16 for strong (ST) / weak (W) points related to usability, at 2DC. (p) =significant

In summary, the change in Lab and Remote conditions allowed us to confirm that, after the usability evaluations, only the participants of such conditions have significantly increased their understanding of usability and, consequently, their empathy towards users' needs.

4.3 Nature of the improvement of understanding of usability: Confirming the increase of empathy

More references to usability matters in Lab and Remote conditions imply that participants focused more on usability after their participation in the evaluations and, consequently, there was an improvement in the understanding of usability for such conditions. At the focus group sessions held in these conditions, we identified increased concern/attention on usability; specifically, on users' roles and their needs.

Some examples of the opinions provided by participants during the focus group could better explain this situation. Most of them highlighted the importance of users in the development process, "The system should be usable enough for users. If not, what would be its purpose?" or

"I must confess that in the past, I have developed software only thinking in that final result should be a functional enough software. I had never thinking in simplicity or attractive for users".

In the focus group session for the Heuristic-Supervised condition, we also detected some concern and interest on users' issues. However, in the case of Heuristic condition, usability matters were almost entirely overlooked. Just before finishing the focus group session for Heuristic condition, we requested a final reflection on the software development process. The participants only provided concepts related to the importance of the overall process, the courses, the experience obtained, and the prospects for their future careers.

5 Discussion

5.1 Empowering empathy for users: The usability evaluation approach

The initial understanding regarding usability identified in 1DC confirms the differences of the aims and motivations of SE and HCI practitioners (Lee 2006, Sohaib and Khan 2010), the software developers' mindset (Bak et al. 2008), and the resistance to User-Centered Design or Usability (Rosenbaum et al 2000). Before the usability evaluations, the participants showed more focus on software matters. This level of understanding also reflected low levels of empathy regarding users' needs.

The level of empathy of participants was modified after the usability evaluations. We detected more identification on the part of the participants with usability concerns, something that is fully consistent with previous studies (Gilmore and Velázquez 2000, Hoegh et al. 2006). The change was more evident and significant in those evaluations conducted under Lab and Remote conditions. Because there were no significant differences at 2DC between strong points related to usability in such conditions (see Table 6), it is possible to argue that the resulting level of understanding regarding usability and empathy were similar for such conditions. We confirmed this during the focus group sessions.

The viability of conducting usability evaluations by software developers (Bruun and Stage 2012, Skov and Stage 2012) was also confirmed in our study. The predilection of participants for the strong points of their software can be explained by the developers' sense of individual ability and a personal identification with the software (Rasch, Tosi 1992; Hertel et al. 2003). In addition, this is another manifestation of the software developers’ mind-set (Bak et al. 2008, Ardito et al. 2011).

Classical usability evaluation methods based on thinking aloud protocol (usability lab and remote synchronous testing) are effective for increasing/improving empathy toward users' needs. However, considering the statistically similar results obtained in both methods, it is a fact that the remote synchronous testing method should be considered to be the best option for the software development process; logistical considerations also make this option sufficiently attractive.

Tentative increments in Heuristic and Heuristic-Supervised conditions are not significant but are always interesting. Even considering that no interaction with users occurred in these methods, results suggest that the participants' understanding and empathy were affected as a result of the assessment of other software.

We cannot contrast our results with existing literature. No previous studies have focused on proposals aimed at improving or increasing software developers’ empathy towards users’ needs. HCI training proposals (Rasch and Tosi 1992, Mattelmäki and Battarbee 2002) do not consider this issue. Alternatives used in other fields could be considered (Boker et al. 2004). However, similar to other fields (Stepien and Baernstein 2006), the diversity of definitions of empathy and the inadequacy of empathy measurement instruments could produce problems when applying this approach.

Indeed, in the context in which our study was conducted, we found some evidence that suggested the limited effectiveness of traditional approaches based on training. The initial level of the participants’ empathy was based on their previous training (courses that included some HCI topics) and their practical experience in technical matters, interaction with users, and so on. However, ultimately, usability evaluations were more effective by producing a relevant improvement in empathy. In the usability evaluation context, it is possible to potentiate reflection to and focus on users' feelings. Furthermore, improving empathy towards users is another way to improve soft skills in software developers, as such soft skills which are normally less well supported in university pedagogy (Begel and Simon 2008, Taft 2007).

5.2 Processes involved in improving/increasing empathy toward users’ needs

The statistically similar results in the increase of understanding and empathy presented in Lab and Remote conditions imply the existence of similar mechanisms behind these results, particularly those of the EC theory.

EC theory involves the automatic or unconscious contagion of users' emotions (Schoenewolf 1990, Hatfield et al. 1994). In our study, the unconscious contagion experienced by participants who saw the users working with a software system in the usability evaluation context explains the change of perspective of the participants. This was confirmed during the focus group sessions.

The absence of interaction with users in Heuristic and Heuristic-Supervised conditions explains why, in these conditions, participants did not experience similar contagion of users' emotions.

Another EC theory concept presented in our experiment is related to the definition of who will be the observer and who the observed in the contagion process. The particularity of the circumstances or personalities involved is a key factor in this regard (Schoenewolf 1990). In our experiment, participants observed users working with a software system. Not only did the participants assume the role of observers, but the users also assumed an active role by thinking aloud, expressing their thoughts, opinions and feelings. The developers passively received these emotions. Specific guidelines and protocols prevented other interactions that could have modified this scenario.

Finally, EC theory has also established that the remote contagion of emotions is possible (Hancock et al. 2008, Kramer 2012). Our study confirmed an increase in empathy not only in a usability lab, but also remotely and with significant resource savings.
Identifying EC theory's mechanisms in the context of our study is not trivial. EC theory helps to understand the reasons behind the high efficiency and effectiveness of conduction or observation of usability evaluations by software developers in order to increase their empathy toward users' needs. This is an approach that should be seriously considered seriously.

6 Conclusion

This paper provides an empirical exploration focused on the potential of usability evaluations to increase or improve empathy toward users' needs. The main contribution of this study is its empirical basis in a realistic software development context.
The usability evaluation methods that included interaction with users had remarkable results in the generation of empathy towards users' needs. Both methods used in our study (usability lab and remote synchronous testing) provided vivid interaction with users, a scenario that enabled the contagion of users' emotions by software developers. Practical and logistical considerations make the remote synchronous testing method the best option to be considered in the software development process.
Our study has a main limitation. The participants in the study were final year undergraduate students. Nevertheless, real conditions present in our study have allowed control of this bias.
Future work may be oriented towards conducting further longitudinal studies in order to explore results in various contexts and with developers who are more experienced.

Acknowledgments

The research behind this paper was partly financed by UNA-MICIT-CONICIT (Costa Rica) and the Danish Research Councils (grant number 09-065143). We are very grateful to the participants, observers, facilitators, organizations and users that helped us in this research

References

Andreasen, M. S., Nielsen, H. V., Schrøder, S. O., and Stage, J., 2007. What happened to remote usability testing?: an empirical study of three methods. *In Proc. SIGCHI*, ACM Press, 1405-1414.

Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R., Bruun, A., and Stage, J., 2011. Usability Evaluation: a survey of software development organizations. *In Proc. SEKE*, 282-287.

Bak, J.O., Nguten, K., Risgaard, P., and Stage, J., 2008. Obstacles to Usability Evaluation in Practice: A Survey of Software Development Organizations. In *Proc. NordiCHI*, ACM Press, 23-32.

Barsade, S. G., 2002. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, 47(4), 644-675

Begel, A., and Simon, B., 2008. Novice software developers, all over again. In *Proc. ICER*, ACM Press, 3-14.

Boker, J. R., Shapiro, J., and Morrison, E. H., 2004. Teaching empathy to first year medical students: evaluation of an elective literature and medicine course. *Education for Health*, 17(1), 73-84.

Brunero, S., Lamont, S., and Coates, M., 2010. A review of empathy education in nursing. *Nursing Inquiry*, 17(1), 65-74

Bruun, A., and Stage, J., 2012. Training software development practitioners in usability testing: an assessment acceptance and prioritization. *In Proc. OzCHI*, ACM Press, 52-60.

De Vignemont, F., 2004. The co-consciousness hypothesis. *Phenomenology and the Cognitive Sciences*, 3(1), 97-114.

Decety, J., and Jackson, P. L., 2006. A social-neuroscience perspective on empathy. *Current directions in psychological science*, 15(2), 54-58.

Fagerholm, F., and Münch, J., 2012. Developer Experience. *In Proc. ICSSP*, IEEE, (2012),73-77.

Gilmore, D. J., and Velázquez, V. L., 2000. Design in harmony with human life. Ext. Abstracts CHI 2000, ACM Press, 235-236.

Grudin, J., 1991. Obstacles to user involvement in software product development, with implications for CSCW. *International Journal of Man-Machine Studies*, 34(3), 435-452.

Hancock, J. T., Gee, K., Ciaccio, K., and Lin, J. M. H., 2008. I'm sad you're sad: emotional contagion in CMC. *In Proc. ACM conference on Computer supported cooperative work*, ACM Press, 295-298.

Hatfield, E., Cacioppo, J. T., and Rapson, R. L., 1994. *Emotional contagion*. Cambridge University Press

Hertel, G., Niedner, S., and Herrmann, S., 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research policy*, 32(7), 1159-1177.

Hoegh, R. T., Nielsen, C. M., Overgaard, M., Pedersen, M. B., and Stage, J., 2006. The impact of usability reports and user test observations on developers' understanding of usability data: An exploratory study. *International journal of Human-Computer Interaction*, 21(2), 173-196.

Karn, J. S., Syed-Abdullah, S., Cowling, A. J., and Holcombe, M., 2007. A study into the effects of personality type and methodology on cohesion in software engineering teams. *Behaviour & Information Technology*, 26(2), 99-111

Kjeldskov, J., Skov, M. B., and Stage, J., 2004. Instant data analysis: conducting usability evaluations in a day. *In Proc. NordiCHI*, ACM Press, 233-240.

Kramer, A. D., 2012. The spread of emotion via facebook. In Proc. SIGCHI, ACM Press, 767-770.

LeBlanc, R. J., Sobel, A., Diaz-Herrera, J. L., and Hilburn, T. B., 2006. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. IEEE Computer Society Press.

Lee, J. C., 2006. Embracing agile development of usable software systems. *Ext. Abstracts CHI 2006*, ACM Press, 1767-1770.

Lunt, B. M., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., and Reichgelt, H., 2008. *Curriculum guidelines for undergraduate degree programs in information technology.* IEEE Computer Society Press.

Mattelmäki, T., and Battarbee, K., 2002. Empathy probes. In *Proc. PDC*, 266-271.

Newell, A. F., Morgan, M. E., Gregor, P., and Carmichael, A., 2006. Theatre as an intermediary between users and CHI designers. *Ext. Abstracts CHI 2006*, ACM Press, 111-116.

Nielsen, J., and Molich, R., 1990. Heuristic evaluation of user interfaces. *In Proc. SIGCHI*, ACM Press, 249-256.

Patton, J., 2002. Hitting the target: adding interaction design to agile software development. *In OOPSLA 2002 Practitioners Reports*, ACM Press, 1-ff.

Preece, J., 2001. Sociability and usability in online communities: determining and measuring success. *Behaviour & Information Technology*, 20(5), 347-356.

Pugh, S. D., 2001. Service with a smile: Emotional contagion in the service encounter. *Academy of management journal*, 44(5), 1018-1027.

Rasch, R. H., and Tosi, H. L., 1992. Factors affecting software developers' performance: an integrated approach. *MIS quarterly*, 395-413.

Rosenbaum, S., Rohn, J. A., and Humburg, J., 2000. A toolkit for strategic usability: results from workshops, panels, and surveys. *In Proc. SIGCHI*, ACM Press, 337-344.

Rubin, J., and Chisnell, D., 2008. *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.

Schoenewolf, G., 1990. Emotional contagion: Behavioral induction in individuals and groups. *Modern Psychoanalysis*, 15(1), 49-61.

Seffah, A., and Metzker, E., 2004. The obstacles and myths of usability and software engineering. *Communications of the ACM*, 47(12), 71-76.

Shaw, K., and Dermoudy, J., 2005. Engendering an empathy for software engineering. *In Proc. of the 7th Australasian conference on Computing education*-Volume 42, 135-144.

Singer, T., and Lamm, C., 2009. The social neuroscience of empathy. *Annals of the New York Academy of Sciences*, 1156(1), 81-96.

Skov, M. B., and Stage, J., 2012. Training software developers and designers to conduct usability evaluations. *Behaviour & Information Technology*, 31(4), 425-435.

Sohaib, O., and Khan, K., 2010. Integrating usability engineering and agile software development: A literature review. *In Proc. Computer Design and Applications ICCDA 2010 International Conference on*, Vol. 2, IEEE, 32-38.

Stepien, K. A., and Baernstein, A., 2006. Educating for empathy. *Journal of general internal medicine*, 21(5), 524-530.

Strauss, A., and Corbin, J. 1998. *Basics of qualitative research. Techniques and Procedures for Developing Grounded Theory*. 2nd ed. SAGE Publications.

Taft, D. K., 2007. Programming Grads Meet a Skills Gap in the Real World. *eWeek.com*.

Yammiyavar, P., 2005. Emotion as a Construct in HCI. *In Research Symposium*, 8.