Gender HCI Issues in End-User Software Engineering Environments

Laura Beckwith, Margaret Burnett, and Susan Wiedenbeck[α]
Oregon State University
Corvallis, OR, USA
{beckwith, burnett}@eecs.oregonstate.edu
[α]Drexel University
Philadelphia, PA, USA
susan.wiedenbeck@cis.drexel.edu

INTRODUCTION

Until recently, research has not considered whether the design of problem-solving software, such as spreadsheets, multimedia authoring languages, and CAD systems, affect males and females differently. As a result, we began investigating how the two genders are impacted by problem-solving software and whether attention to gender differences is important in the design of software. Evidence from other domains, such as psychology and marketing (see [Beckwith and Burnett 2004]), strongly suggests that females process information and problem solve in very different ways than males do. This implies that without taking these differences into account in the design of problem-solving software, the needs of half the population for whom the software is intended are potentially being ignored. In fact, some research has shown that software is unintentionally designed for males.

To consider this issue, we are empirically investigating end users engaged in end-user software engineering activities, to inform the design of software to support end-user programmers of both genders.

METHOD

Our method for conducting this investigation consists of four steps: (1) draw from theory and previous gender difference empirical work from other domains—such as computer confidence, perceived risk, information processing, computing gaming, and technology adoption models—to hypothesize gender issues and their causes that could arise from gender-based differences in the use of problem-solving software [Beckwith and Burnett 2004], (2) use empirical methods to investigate whether these issues do actually arise in problem-solving software, (3) use the results of the first two steps along with qualitative empirical work involving low-cost prototyping to derive and refine approaches to address the issues, and (4) use quantitative empirical methods to evaluate the effectiveness of the approaches.

We have conducted four studies investigating gender differences relevant to end-user software engineering environments. Three of these are summarized here; more detail on the series of studies can be found in [Beckwith et al. 2006b].

EMPIRICAL EVIDENCE – SELF-EFFICACY

Guided by literature and early exploratory analyses, we performed a quantitative investigation of the impact of self-efficacy (a form of confidence) and gender on users' use of end-user testing and debugging features while debugging a spreadsheet [Beckwith et al. 2005]. The results of that study showed how these differences in self-efficacy negatively impacted acceptance of the features by females, and showed that the reduced feature acceptance can significantly reduce females' effectiveness at problem solving. More specifically:

- Females' self-efficacy was predictive of their effectiveness at using the debugging features, which was not the case for the males. See Figure 1. Thus, the (many) low self-efficacy females were unlikely to use the features, but the (few) low self-efficacy males were as likely to use the features as the high self-efficacy males were.
- Females were less likely than males were to accept the new debugging features (unfamiliar to all participants prior to the experiment). One reason females stated for this was that they thought the features would take them too long to learn—but there was no difference in the males' and females' learning of the new features.
- Although there was no gender difference in fixing the seeded bugs, females introduced more new bugs—which remained unfixed. This is probably explained by low acceptance of the debugging features: high effective usage of the features was a significant predictor of fixing bugs.

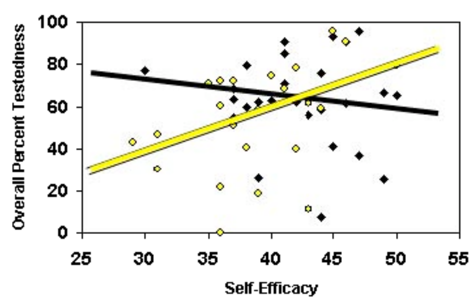

Figure 1. Females' (light) self-efficacy was a significant predictor of their effective use of the "check-off" feature, as the positively sloping line shows. For the males (dark), however, this was not the case.

Dagstuhl Seminar Proceedings 07081
End-User Software Engineering
http://drops.dagstuhl.de/opus/volltexte/2007/1076

QUALITATIVE EVIDENCE – FEATURES & MOTIVATION

A think-aloud study [Beckwith et al. 2006b] provided confirmatory evidence of females' beliefs and perceptions that seem tied to their avoidance of the debugging features. The experiment also revealed an interesting difference in the ways features were perceived by males and females. For example, female F2, in using the "guards" feature (akin to Excel's "data validation"), said:

> F2: "I don't think that you can get a -5 on the homework. No, it can't be. So 0 to 100 [is the guard I'm entering], ok. Ok, hmm… So, it doesn't like the -5 [...]. They can get a 0, which gets rid of the angry red circle."

In contrast to F2's focus on the guard feature as a way to get her spreadsheet to work correctly, the following male's initial focus was on the feature itself:

> M3: "The first thing I'm going to do is go through and check the guards for everything, just to make sure none of the entered values are above or below any of the ranges specified. So, homework 1—actually, I'm going to put guards on everything because I feel like it. I don't even know if this is really necessary, but it's fun."

Despite his initial interest in the feature for the fun of it, the male soon transitioned to its problem-solving advantages and was able to find and fix a bug with the aid of the feature. His use of guards because "it's fun" led us to the next study, investigating the role of exploratory investigation (or tinkering) as a manner of becoming comfortable with the features and environment.

EMPIRICAL EVIDENCE – TINKERING

In this quantitative study [Beckwith et al. 2006a], we originally anticipated that males' propensity to tinker (playfully experiment) would benefit their problem solving. However, we found that even small differences in the environments had big impacts on how gender and tinkering interacted and affected debugging effectiveness. More specifically:

- As in previous research, males tinkered more than females but, surprisingly, males' tinkering was often counterproductive to their effectiveness in debugging.
- One factor in the above result was the fact that the low-cost variant of the spreadsheet environment led some males to engage in unproductive, repeated tinkering, which was linked to poor understanding.
- Although they tinkered less, females' tinkering was effective: it was significantly tied to understanding and to successfully testing and debugging, regardless of environment. However, when tinkering in the more complex environment, females' tinkering was predictive of lower self-efficacy.

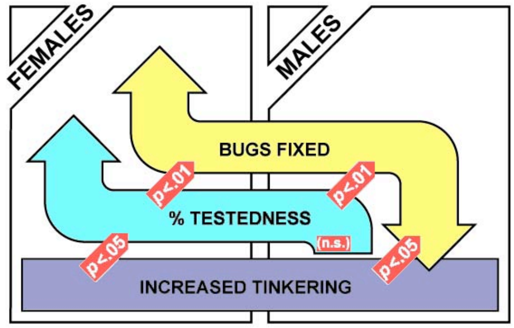

Figure 2. Males' and females' tinkering affected their debugging effectiveness, but in essentially opposite ways. Direction of stylized arrows depicts increase/decrease in a measure, and shaded arrows show significance of the regression relationships between measures.

- Tinkering with pauses allows for reflection and was helpful to everyone, but females were more likely than males to pause.

The essence of these results is depicted in Figure 2.

The implications are that designers should look for ways to encourage females' tinkering. Still, care must be taken to avoid at the same time encouraging males' tinkering further, since males' tinkering tended to be excessive and, when this was the case, was counterproductive.

SUMMARY

These results are being used to experiment with new ways software designs can counteract these effects. The outcomes of these experiments can provide the knowledge required to design future environments to better allow end-user programmers of *both* genders to succeed at end-user software engineering tasks.

REFERENCES

[Beckwith and Burnett 2004] Beckwith, L. and Burnett, M. Gender: An important factor in end-user programming environments? *IEEE Symp. Visual Languages and Human-Centric Computing Languages and Environments*, Sept. 2004, 107-114.

[Beckwith et al. 2005] Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S., and Hastings, M. Effectiveness of end-user debugging features: Are there gender issues? *ACM Conf. Human Factors in Computing Systems*, April 2005, 869-878.

[Beckwith et al. 2006a] Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A., and Cook, C. Tinkering and gender in end-user programmers' debugging. *ACM Conf. Human Factors in Computing Systems*, April 2006, 231-240.

[Beckwith et al. 2006b] Beckwith, L., Burnett, M., Wiedenbeck, S., and Grigoreanu, V. Gender HCI: What about the software? *IEEE Computer*, Nov. 2006, 83-87.