AN ABSTRACT OF THE THESIS OF

Shraddha R. Sorte for the degree of Master of Science in Computer Science presented on September 26, 2005.

Title: End User Software Engineering Features for Both Genders.

Abstract approved:

_____

Margaret M. Burnett

Previous research has revealed gender differences that impact females' willingness to adopt software features in end users' programming environments. Since these features have separately been shown to help end users problem solve, it is important to female end users' productivity that we find ways to make these features more acceptable to females. This thesis draws from our ongoing work with users to help inform our design of theory-based methods for encouraging effective feature usage by both genders. This design effort is the first to begin addressing the gender differences in the ways that people go about problem solving in end-user programming situations.

End User Software Engineering Features for Both Genders

by
Shraddha R. Sorte

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 26, 2005
Commencement June 2006

Master of Science thesis of Shraddha R. Sorte presented on September 26, 2005.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Shraddha R. Sorte, Author

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude towards my advisor Dr. Margaret Burnett for her constant support, guidance and encouragement throughout my work. She has been a great source of inspiration in my academic life. She entrusted me with many tasks throughout the year. This helped me gain a completely new experience and develop a new perspective of thinking. I must thank her for giving me this wonderful opportunity to work with her.

I would also like to thank all the members of the Forms/3 team who lend their helping hand in resolving the bugs in time for the main experiment and resolving many minor issues in implementing the prototype with special mention to Andrew Christmann and Joey Lawrence who helped me develop the prototype and take it to the final stage. Special thanks to Laura Beckwith who helped me a lot during the work and also provided guidance during early stages of the work. It was great working with her.

I thank the participants of our study and also extend this thanks to all those in the EECS department, for all their support and who made working here enjoyable. I must also thank all my friends who have directly or indirectly provided help at some of the craziest hours of work.

I am indebted to the most important people in my life: my family, especially my parents, sister and all other family members. They gave me all the support and encouragement throughout my career. Nothing that I have accomplished would have been possible without them.

This work was supported in part by Microsoft Research, by NSF grant CNS-0420533 and by the EUSES Consortium via NSF grants ITR-0325273 and CCR-0324844.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF APPENDIX FIGURES

Figure Page

End User Software Engineering Features for Both Genders

Shraddha Sorte
Oregon State University

1. Introduction

Although there have been gender studies [Camp 1997] designed to understand and ameliorate the low representation of females in the computing field, there has been little emphasis on software's *design* attributes and how these design attributes affect females' and males' performance in computing tasks. Building upon theories and research about gender differences from a number of domains [Beckwith and Burnett 2004], our research group has begun investigating whether there are features within software that interact with gender differences in the realm of end-user programming.

We used theory and previous gender difference empirical work from other domains to hypothesize gender issues and their causes that could arise in end-user programming environments and used empirical methods to investigate whether these issues do actually arise in end-user programming environments. This work concentrates on using the empirical results along with theory and qualitative empirical work involving low-cost prototyping to derive and refine approaches to address the issues.

Our group's work on the first step was presented in [Beckwith and Burnett 2004]. In that paper, our group derived a set of hypotheses from relevant research literature; the subset of those hypotheses of interest are given in Table 1. A particularly useful aspect of these hypotheses is that, because many of these hypotheses are theory-based, they tend to suggest a cause for the hypothesized effect. These causes potentially point the direction for our designs to take in addressing issues that are present.

Basis: Confidence and Risk
H1: There will be gender differences in users' interest in (initially) exploring new features in end-user programming environments.
H2: Females' high perceptions of risk will render them less likely to make (genuine) use of unfamiliar devices in end-user programming environments.

Table 1: Theory-based hypotheses about gender differences in end-user programming environments [Beckwith and Burnett 2004]

Our group's work has so far concentrated on hypotheses H1 and H2 in the table. To investigate these hypotheses, we conducted a study [Beckwith et al. 2005] in which we gave male and female spreadsheet users two spreadsheet debugging tasks in an environment containing a number of features that support such debugging tasks. The hypotheses were confirmed by our investigation:

- Females had lower self-efficacy (a form of confidence) than males did about their abilities to debug. Further, females' self-efficacy was predictive of their effectiveness at using the debugging features (which was not the case for the males).
- Females were less likely than males to accept the new debugging features. A reason females stated for this was that they thought the features would take them too long to learn. Yet, there was no real difference in the males' and females' ability to learn the new features.
- Although there was no gender difference in fixing the seeded bugs, females introduced more new bugs—which remained unfixed. This appears to be explained by their low acceptance of the debugging features which left editing formulas as their primary "debugging" device. High effective usage of the debugging features was a significant predictor of ability to fix bugs.

This thesis reports the results of the next step of our investigation: applying the above findings related to H1 and H2: to the development of potential solutions to address the issues revealed. As design progressed, we supplemented theory-derived approaches with qualitative empirical methods and low-cost prototyping to refine our proposed solutions.

This thesis also details how the prototype design evolved with the help of continuous feedback from the end users. Three think-aloud studies were conducted to evaluate the prototype design at each stage. The results seemed promising but they need to be confirmed by a follow-up summative experiment.

The contributions of this work are two-fold. First, it shows the application of these particular theories to the design of potential solutions to address gender issues in end-user programming features. Second, the potential solutions themselves are, to our knowledge, the first reported approaches to target gender issues for end-user programming environments.

2. Background

2.1 Related Work

Gender differences in attitudes toward technology: As reported in [Ray 2003] there are gender differences in attitudes toward technology. Males saw machines as a challenge, something to be mastered, overcome, and be measured against. They were the risk takers, and they demonstrated this by eagerly trying new techniques and approaches. On the other hand, females approached the machine as a tool, and attempted to work with it in a cooperative manner. So, rather than dominate the machine, females attempted to work with it to achieve their goals.

Gender differences in mathematical skills: A study [Fennema and Sherman 1977] by Fennema and Sherman attempted to show gender differences in spatial and mathematical abilities. They found that males generally outperformed females on mental rotation tasks and on problems requiring mathematical skills.

Gender differences in decision making: Altizer et al. [Altizer et al. 1996] hypothesize that given the gender differences in mathematical skills and information processing, there will be gender differences in decision making. They classified decision strategies as compensatory and non-compensatory. Compensatory decision strategies use all available information in a decision task, whereas non-compensatory strategies focus on a limited set of information available. They hypothesize that females will generally use compensatory strategies, whereas males will generally use non-compensatory strategies. They haven't reported about the results yet.

Gender differences in information processing: Meyers-Levy and Maheswaran conducted a study [Meyers-Levy and Maheswaran 1991] to explore differences in males' and females' information processing strategies. They found that females' processing often involved detailed elaboration of message content, sometimes even focusing on the particulars of message

claims. In general, these studies revealed that females' information processing strategies were more detail-oriented while males were more schema-based or theme-oriented.

Interference of stereotype threat: Quinn and Spencer in [Quinn and Spencer 2001] point out that stereotypes about academic skills are well known and according to these stereotypes, males are better at math and science domains and females are better at English and reading domains. These stereotypes are transmitted in the culture in a variety of ways, including books, media, parents, peers, and teachers. It has been observed that even the females who have achieved the most, who have the strongest math skills, underperform in comparison to their male peers. The authors considered whether there was an interaction between cultural stereotypes and test-taking situation which they termed as "stereotype threat" situation. They conducted a study to find out the relation. The study revealed that under conditions of high stereotype threat, females underperformed in comparison to males, and were less likely to be able to formulate strategies. However, when they were told that the same test was gender fair, thereby reducing the stereotype threat, males and females performed equally on the test and did not differ in the ability to formulate and use strategies. The possible explanation that the authors gave for women's difficulty in formulating strategies when stereotype threat is high is that stereotype threat may reduce the cognitive resources available to generate strategies.

Gender differences in games and software design: A study [Huff and Cooper 1987] by Huff and Cooper revealed that there was a bias in designing software for each gender. The designers were asked to design programs for boys, girls, and students in general. Both boy and student programs were game oriented (requiring more hand-eye coordination and more action on the screen) while those for girls were learning tools.

Much later, Miller conducted a small pilot study [Miller 1996] to investigate girls' preferences in computer software and future interactive software. Findings of this study are briefly summarized below:

- *Manual – last resort*: If the game or the environment was not self-revealing, girls were not motivated to pursue the manual. They would instead look around for another available computer when they were stuck as a way to exit this "stuck" status, or if none were available, they would ask the person next to them for help.
- *Non-closure/Exploration*: Girls seemed to move freely among games without seeming to need to complete or win one game or segment before switching.
- *Rich Texture*: Girls placed a high value on the quality of the visual and audio design of an environment. The richer the texture of the environment, the more it appealed to the girls.
- *Supportive over competitive environment*: Most of the girls expressed a desire that a game be challenging and include elements of problem solving, but not to the point of causing frustration. The girls wanted the activity to challenge them, but they did not view winning as a necessary objective. They placed priority on having a good experience and wanted the game to include features that preferred supportive feedback.
- *Education versus entertainment*: Younger girls, regardless of computer experience, preferred the entertainment environment, while the older girls preferred the more informational options.
- *Virtual reality*: Many girls advocated the idea of vicariously experiencing adventures or activities.
- *Career Exploration*: Providing real life simulations and role-playing associated with a variety of careers gained the girls' interest.

This study points out that using girls' imaginations and learning styles as the starting point, rather than expecting girls to be accommodated by male-produced and accepted games, is the next step in providing alternatives that may ultimately lead to re-capturing girls' interests in computing and its associated professional opportunities.

Gender differences in self-efficacy and its effect on software adoption and use: Hartzel conducted a study [Hartzel 2003] to find out if a tutorial affected self-efficacy[1] of the participants and if self-efficacy affected the likelihood of successful use and adoption of the software. The study revealed that previous experience predicted higher comfort levels. Participants with more experience using computer-based technologies had higher task-specific self-efficacy levels than those with less experience. Also this study confirmed the results of past studies that there exists a relationship between self-efficacy beliefs concerning computer use and the motivation to use those technologies. Also, self efficacy has a cumulative nature and experiences build on each other. The study also found that including a tutorial boosted the self-efficacy. This was especially true for the women.

Gender differences in computer confidence and its effect on problem solving: Computer confidence based in gaming experience can affect girls' success in problem-solving. From their observations of girls playing computer and video games, Inkpen et al. [Inkpen et al. 1994] concluded that the confidence levels of selected study participants affected their playing abilities and their willingness to solve problems through trial and error. When the girls in their study doubted their abilities, they were less likely to tackle math problems embedded in games, and they had less success in completing the games.

---

[1] Self-efficacy is the measure of one's confidence in mastering a new challenge. When self-efficacy is high, one believes there is a high probability that one will be successful, while low self-efficacy suggests a limited belief one will accomplish an objective.

Gender differences in behavior towards software and its features: Microsoft reported, in a workshop [Greenberg 1993], an unpublished study which categorized users into two Profiles, A or B, depending on their perception of software as bloated or not. Profile A users preferred software that was complete, they stayed up-to-date with upgrades, they assumed that all interface elements have some value, and they blamed themselves when something went wrong or when they couldn't figure out how to perform a specific task. Interestingly, this category was comprised of mostly females. Alternately, Profile B users preferred to pay for and used only what they needed, they were suspicious of upgrades, they wanted only the interface elements that were used, and they blamed the software and the help system when they couldn't do a task. These were mostly males.

Another study was conducted to gain better understanding of how the users actually experienced software bloat[2] (complex functionality-filled software applications) and the extent to which users experience them in similar/different ways. This study also confirmed the results of the Microsoft study [Greenberg 1993]. They found that gender was a significant factor in the perception of bloat between the two groups of users and that it was females who fell into Profile A, i.e., those wanting the most up-to-date, and complete version of the software.

Gender difference in dealing with help systems: A study [Fennema et al. 1998] of children's problem solving abilities revealed gender differences in strategy use. Girls tended to use concrete modeling (e.g.: counting on fingers) or counting strategies (i.e. following the methods they were taught), while boys tended to use more abstract strategies such as invented algorithms or derived facts. These results may imply that girls need more structured and

[2] Software bloat has been defined as the result of adding new features to a program or system to the point where the benefit of the new features is outweighed by the impact on the technical resources and the complexity of use. A bloated application is one in which there are a large number of unused features.

concrete help approaches, while boys may suffice with retrieval-based and more abstract help approaches.

Arroyo [Arroyo 2003] found that boys of high cognitive development ignored help the most, while girls of high cognitive development ignored them the least, spending more time within hints overall. Also, girls spent more time working with hints than boys, on average. In general, girls seemed more affected by the over-support and the under-support. The study [Arroyo et al. 2000] by Arroyo reported that girls performed better in subsequent problems when help was highly interactive, while boys performed better in subsequent problems when the help had low levels of interactivity. Thus girls were willing to spend more time on hints, and interaction with help eventually turned into better learning.

A second study on AnimalWatch (a mathematics intelligent tutoring system) in 1998 showed that while girls' self-confidence was positively affected by highly interactive and high amounts of help, boys' self confidence improved significantly most with a version that provided reduced help [Beck et al. 1999]. This also supports the fact that girls feel comfortable with high levels of support while boys may feel comfortable with low levels of support.

2.2 Confidence and Self-Efficacy

As seen in previous section, gender differences regarding computer confidence have been widely studied, revealing that females (both computer science majors and end users) have lower self-confidence than males in their computer-related abilities [Huff 2002].

Self-efficacy is a person's judgment about his or her ability to carry out a course of action to achieve a certain type of performance. Achieving a desired type of performance depends on two factors, the skills needed to carry out the task and the perception of efficacy that will allow the individuals to use their skills effectively. High self-efficacy is critical in problem solving because self-efficacy influences the use of cognitive strategies, the amount of effort put

forth, the level of persistence, the coping strategies adopted in the face of obstacles, and the final performance outcome.

Research has shown that low self-efficacy affects females' perceptions of a software application before actual use [Hartzel 2003], raising the possibility that females with low self-efficacy may avoid using it altogether. Through self-efficacy literature and a short survey of our own, we consider how confidence and perceived risk might be tied to feature acceptance.

2.2.1 Gender Survey

There were studies done a number of years ago that reported these results; however software has changed significantly since then. Thus, in part to confirm this phenomenon in 2004-era software, and in part to consider potential ties with feature acceptance, we ran a small survey. Our survey looked for links between respondents' software confidence and their self-reported willingness to explore new features in their real-world computer usage, with questions such as "I avoid working with new software since it requires more time to learn," "If something goes wrong with the software (like the program crashes), I believe I can fix it," and "I enjoy exploring new features provided with the software." Some of these are summarized in Table 2 and Table 2 (contd). We administered the questionnaire in a psychology class at Oregon State University in July 2004. There were 32 questions (26 agree/disagree, 5 ranking, 1 subjective) that took approximately ten minutes to complete. Questions were answered on either a five-point Likert scale (1=disagree,… 5=agree) or a ranking of choices (1=highest ranking, 2=second highest ranking, and so on). There were 21 respondents: 14 females and 7 males enrolled in an undergraduate psychology course; mostly psychology and business majors. Approximately two-thirds were psychology majors and one-third were business majors. Out of the 14 females, there were 9 Psychology, 3 Business and 2 Liberal Studies majors. Out of the 7 males, there were 5 Psychology, 1 Business and 1 Arts major.

Our survey results were extremely consistent with the findings reported in [Huff 2002]. In all ten of our questions about software confidence and respondents' acceptance of new or advanced software features, females' mean scores were lower than the males'. In fact, even with this small sample size, many of these differences were statistically significant.

In particular, Mann Whitney on the self-confidence questions revealed

Question	Mean		Mean Rank		P-value
	F	M	F	M	
I work independently on most of my computer work	4.9	4.6	12.0	9.0	0.167
Software is difficult to understand	3.0	2.1	12.3	8.5	0.174
I avoid working with new software since it requires more time to learn	3.5	2.1	13.0	6.9	**0.027**
I avoid working with new software since it requires me to think more	3.1	1.7	13.2	6.5	**0.015**
I find that most software is self explanatory	3.1	4.4	8.9	15.3	**0.019**
If something goes wrong with the software (like the program crashes), I believe I can fix it	1.8	3.9	8.1	16.9	**0.001**
I am usually confident that I understand the functionality of these features	3.8	4.3	9.9	13.3	0.192
I am comfortable changing the settings of these features	3.2	4.3	9.2	14.5	0.057
I enjoy exploring new features provided with the software	3.1	4.0	9.3	14.4	0.067

Table 2: Survey Questionnaire – Summary of questions of type Agree/Disagree (Mann Whitney Test)

Question	Mean		Mean Rank		P-value
	F	M	F	M	
Use Web Browser frequently	2.4	1.1	13.0	7.0	**0.023**
Use Email frequently	1.6	2.4	9.2	14.6	**0.043**
Use Word Processor frequently	3.1	4.9	9.4	14.1	0.094
If something goes wrong with the software, I seek help from someone to fix it	1.6	3.0	8.8	15.5	**0.010**
When I have problems using the software, I refer to a technical expert	1.7	3.1	9.2	14.6	0.053

Table 2 (Continued): Survey Questionnaire – Summary of ranking questions (Mann Whitney Test)

that females had significantly lower self-confidence than males (p-value=.0056). Also for females, there was a significant relationship between self-confidence[3] and how they rated themselves in exploring new software features (Table 4). The results of Mann Whitney on individual questions are summarized in Table 2. Cronbach Alpha Reliability[4] test run on the group of questions related to confidence gave a highly significant alpha value = .8; this lends credence to the results.

[3] Self-confidence of the subject was calculated by rating the answers to the questions listed in Table 3. Those who agreed with the first set of questions in Table 3 were given positive points for self-confidence based on their degree of agreement while those who agreed with the next set of questions were given negative points based on their degree of disagreement.

[4] Cronbach alpha test is the most common form of reliability (or consistency) coefficient. It is not a statistical test and is used to estimate the proportion of variance that is systematic or consistent in a set of scores. It can range from 0.00 (if no variance is consistent) to 1.00 (if all variance is consistent). For example, if the Cronbach alpha for a set of scores is .90, then the test is 90% reliable. By convention, alpha should be .70 or higher to retain an item in a scale.

Questions contributing to negative points
Software is difficult to understand.
I avoid working with new software since it requires more time to learn.
I avoid working with new software since it requires me to think more.
Questions contributing to positive points
Software helps me perform my task more quickly.
I find that most software is self-explanatory.
If something goes wrong with the software (like the program crashes), I believe I can fix it.

Table 3: Self-efficacy questions – Subset of questions (from the questionnaire) considered for self efficacy rating

Gender	Regression Test Results
Males (n=7)	R^2=.1405, p=.3967
Females (n=14)	R^2=.5799, p=.**0016**
All participants (n=21)	R^2=.5107, p= .**0003**

Table 4: Survey Qus results – Results of regression analysis of self confidence as a predictor of exploring new software

2.2.2 Gender differences in debugging in spreadsheet-like environment

Our study [Beckwith et al. 2005] was aimed at investigating gender-related issues within software aiming to support end-user programmers The results of this study established ties from the well known gender differences in computer-related confidence to end users' debugging behaviors. The females, whose self-efficacy was significantly lower than the males, were less willing to accept the new debugging features in the software environment—which is unfortunate, because these features, which explicitly support testing and debugging, were statistically significant predictors of debugging success.

The results also indicated that previous experience with spreadsheets has an important influence on self-efficacy. Lower self-efficacy of females for spreadsheet debugging may be remediated by greater experience. Thus, as a female gets more experience, including experience with end-user debugging features, her self-efficacy can be expected to rise, with corresponding increases in effective usage of features that increase performance.

However, there is a circular dependency here—a female may never gain the experience needed to raise her self-efficacy and performance capabilities if she has already concluded that it is too risky or costly due to her perceived capabilities being too low. In this situation, time itself is not enough to produce the needed experience to raise self-efficacy. Consequently, looking to other, more aggressive, methods seems warranted.

Females' perceptions of their inability to learn new features were not borne out by their actual learning of these features. This suggests that females' low self-efficacy were a self-fulfilling prophecy: their low expectations about their ability to learn new features prevented them from achieving the benefits the new features might have brought them.

This also suggests that a partial solution may lie in the content of communication that helps users to assess both the worth and risks of using the features. Such communication may need to convince users not only of the features' ease of use, but also of the accuracy risks they are taking by not using the features.

This study was a starting point that led us to address gender issues in our debugging environment by taking into consideration the behaviors of the females. The rest of the thesis is an attempt to address these gender issues to ameliorate these gender differences in our environment.

3. Prototype Design

For some time, our group has been working on a concept we term "end-user software engineering" [Burnett et al 2004]. The essence of the end-user software engineering concept is to tightly intertwine into end-user programming environments features that aid end users in guarding against errors in the "programs" they create (spreadsheets in our case). This section describes the end-user software engineering features as they existed in our prototype at the time of the empirical study that investigated H1 and H2.

As the results of H1 and H2 showed, the environment was not as effective for females as it was for males. As one specific example, females' self-efficacy was a significant predictor of their effectiveness testing spreadsheet formulas. For the males, however, this was not the case. In short, for females, low self-efficacy was tied to low usage of useful features, creating a barrier to effectively testing and debugging their spreadsheet formulas.

3.1 Forms/3 – End user Software Engineering Environment

Forms/3 is the research spreadsheet environment in which we are prototyping our work. Forms/3 is a declarative spreadsheet language, although it varies from traditional spreadsheet languages. One of the most visible variations is the lack of a predefined grid layout that cells must belong to; cells can be placed anywhere within the form (see Figure 1). Although cells can be placed anywhere within the spreadsheet, there is also support for more structure in grids. In Forms/3 grids, rows and columns are determined by user-specified formulas.

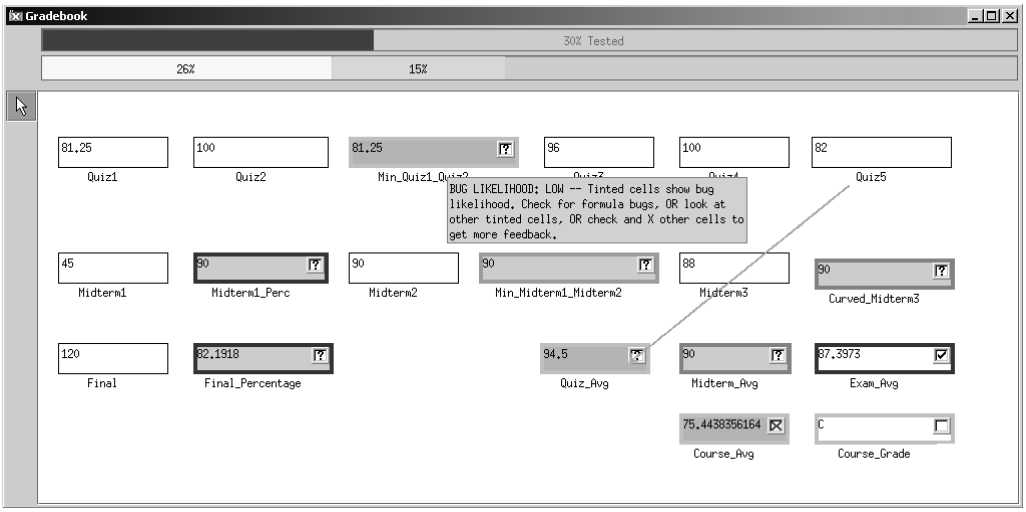

Figure 1: WYSIWYT with fault localization as prototyped in Forms/3 [Burnett et al. 2001]. The user notices an incorrect value in Course_Avg and places an X-mark in the cell. As a result of this X and the checkmark in Exam_Avg, eight cells are identified as being possible reasons for the incorrect value, with some deemed more likely than others.

3.1.1 The Features: WYSIWYT with Fault Localization

WYSIWYT ("What You See Is What You Test") is a collection of two end-user software engineering features – testing and debugging features that allow users to incrementally "check off" ("√" in Figure 1) or "X out" ("X" in Figure 1) values that are correct or incorrect, respectively [Rothermel et al. 2001; Ruthruff et al. 2005]. Besides the checkmarks and X-marks, there are optional dataflow arrows for making relationships among the cells and sub expressions explicit.

The effects of these features are that marking values correct and incorrect allows the system to track the "testedness" and estimate the fault likelihood of all the cells contributing to those correct and incorrect values. The underlying assumption behind WYSIWYT is that, as a user incrementally develops a spreadsheet, he or she can also be testing incrementally. Figure 1 shows an example of WYSIWYT in Forms/3 [Burnett et al. 2001]. Untested cells start with red borders. Whenever users notice a correct value, they can

place a checkmark (√) in the decision box at the corner of the cell they observe to be correct: this communicates a successful test. Behind the scenes, checkmarks increase the "testedness" of a cell according to a test adequacy criterion based on formula expression coverage (described in [Rothermel et al. 2001]), and this is depicted by the cell's border becoming more blue. Instead of noticing that a cell's value is correct, the user might notice that the value is incorrect. In this case, instead of checking off the value, the user can put an X-mark in the cell's decision box. X-marks trigger fault likelihood calculations for each cell that might have contributed to the incorrect value [Ruthruff et al. 2003]. Cells that are likely to contain faults are colored in shades of yellow-orange with darker shades (more orange) indicating higher fault likelihood. The goal of these features is to encourage the users to test the spreadsheet thoroughly and correct errors.

In Figure 1, the user has popped up Quiz5's arrow, which shows both that Quiz5 is referenced in Quiz_Avg's formula and that this relationship is not yet tested. The arrows also reflect WYSIWYT "testedness" status at a finer level of detail. (The user can turn these arrows on/off at will.) These features were present when the above empirical results were obtained. Also visible in Figure 1 are the progress bar (top) which reflects the testedness of the entire spreadsheet and the fault likelihood bar (below the testedness bar) which reflects the likelihood of faults in the tinted cells in the spreadsheet.

3.1.2 Surprise-Explain-Reward

The way these features are supported is via the Surprise-Explain-Reward strategy [Robertson et al. 2004; Ruthruff et al. 2004; Wilson et al. 2003]. If a user is surprised by or becomes curious about any of the feedback of the debugging features, such as cell border color or interior cell coloring, he or she can seek an explanation, available via tool tips (as in Figure 1). If the user follows up as advised in the explanation, rewards potentially ensue.

The aim of the strategy is that, if the user follows up as advised in the explanation, rewards will ensue [Ruthruff et al. 2004]. Some of the potential rewards are functional—such as being led directly to a bug—and some are affective—such as increased progress in the progress bar. One aspect of interest is whether, if gender differences in confidence were present, they might impact Surprise-Explain-Reward's success in encouraging users to approach and adopt new features.

Empirical results with end-user software engineering as supported by Surprise-Explain-Reward have been encouraging [Burnett et al. 2004; Wilson et al. 2003]. Still, the results of our investigation into H1 and H2 [Beckwith et al. 2005] suggest that the Surprise-Explain-Reward strategy was not as effective at enticing females as it was for males to use the features. This was the case not only for seriously adopting and using the features, but even for approaching the features to try them out. The theory-based hypotheses H1 and H2 mentioned earlier suggest that females' lower confidence and higher perception of risk may well be causes. The next section considers specific barriers that may be contributing to these results, and how to remove them.

3.2 Known Barriers

We drew from a combination of existing empirical results, theory, and human-computer interaction (HCI) design techniques. Following Ko et al.'s example [Ko et al. 2004], we use the concept of "barriers" to help organize the problem space. Table 5 lists known barriers. Our empirical results on H1 and H2 were the sources of the barriers.

Regarding *Barrier 1,* as our earlier work pointed out [Beckwith et al. 2005], low confidence in females in computer-related tasks was one of the barriers in approaching or adopting features in the environment. A potential solution could be to increase their experience to help increase confidence but this does not seem very useful by itself—seeming to come down to "the best way to increase feature usage is to increase feature usage"—but it could

1.	Low computer-related confidence in females (as measured in [Beckwith et al. 2005] and numerous other sources)
2.	Low feature usage by females [Beckwith et al. 2005]
3.	Perception that it will take too long to learn the X-mark feature (reported by females in [Beckwith et al. 2005])
4.	Not able to understand fault localization feedback (observations of our subjects' behavior)

Table 5: Barriers females faced related to the findings of H1 and H2

magnify the effects of other solutions that encourage users to get at least a little experience in the course of trying out the features.

According to the attention investment model [Blackwell 2002], users will take an action if they believe that the action's benefits are greater than their perceived costs and are likely to materialize given the perceived risks. This implies that a potential solution to Barrier 1 should emphasize the low risk nature of checkmarks and X-marks. Taking this into account in conjunction with females' low confidence led to two low-risk, low-confidence design ideas, in which users need not be 100% certain of the correctness of their judgments in order to make these marks.

Barrier 2, low feature usage by females, is not independent of the other barriers, but is present in the table because it encourages thinking directly about usage, rather than concentrating only on underlying causes, as in the other barriers.

Barrier 3, females' perceptions that it takes too long to learn the X-mark feature has several possible solutions. The first is ensuring the usefulness of the feature is clearly stated. The attention investment model's benefits component suggests that, if benefits of placing X-marks are not obvious to

users, they are not likely to see learning the feature as a good use of their time, especially if they expect that amount of time to be large. A potential solution could be to observe peers accomplishing the task, which is an important source of self-efficacy. This would mean that a low self-efficacy female should observe another female peer. Our collaborators at Drexel University are further investigating this.

It is also possible that the feedback about the results of X-marks led to *Barrier 4*. If so, then enhancing the feedback would help reduce the barrier. Arroyo [Arroyo 2003] and Beck et al. [Beck et al. 1999] support interactivity in learning to understand tasks, and both studies revealed useful information about gender. Arroyo's study suggested that concrete and interactive hints helped females to perform better and learn more. Beck et al.'s study further indicated that highly interactive hints helped increase females' confidence.

3.3 Are there any other potential barriers?

In addition to the known barriers of the previous section, another table Table 6 of potential barriers and items to be studied was created. Some of these "to study" items were uncovered in the think aloud studies that are discussed in Sections 4, 5 and 6. Note that the barriers in Table 5 were confirmed barriers for females [Beckwith et al. 2005]. Some research points out gender differences in many other aspects which can also be mapped in our environment. We believe that they are relevant and hence need to be considered. However, they were not confirmed in our earlier studies, so whether these are indeed barriers in our environment for the females needs further investigation. We discuss them here and in Table 6 as "potential barriers".

Potential Barriers 1 and 4: Is the feedback of fault localization (involving too many colors) a punishment to the females? These barriers are interrelated and were derived from [Ray 2003], which states that in games for females, the player should not be punished for a wrong action by having to

restart the game again. Instead there should be ways to block the player's progress for a wrong action. They point out that there should be an "element of forgiveness" in the game. This can be mapped to a problem-solving environment like Forms/3 such that the users are not punished for their wrong actions. It is not known if the colors used in the fault localization or WYSIWYT feedback is a punishment to the low confidence users. The low confidence users might be overwhelmed by too many color shades used in the feedback. A potential solution could be to not have as many colors in the Fault localization or WYSIWYT feedback.

Potential Barrier 2 was derived from Arroyo's study [Arroyo 2003], according to which male and female students performed better with different versions of hints/help system. Females were more sensitive to the amounts of help fitting their needs than to the level of abstraction while males were affected by the level of abstraction and ignored help more. As described in section 3.1.2, Forms/3 uses the Surprise-Explain-Reward strategy. The explanation component of this strategy is supported via tooltips in the system. Every feature in the system has a tooltip associated with it. The three main components of explanations include: the semantics of the object, suggested action(s) if any, and the reward; these are described in detail in [Wilson et al. 2003]. These explanations might not be serving the purpose of females.

Potential Barrier 3 was also derived from [Ray 2003] which states that "machine as a foe" became a barrier to the females' enjoyment. The game mechanics should be intuitive and easy to learn. The fundamentals of the game should not be "hidden" within the technology as this requires the player to "fight" the technology in order to enjoy the game. Similarly the working of various features in a problem-solving environment should be intuitive to the users and should not be hidden. Features in the Forms/3 environment may not seem intuitive to the females.

Potential Barrier 5: Girls described audio and visual support in the environment as important as [Miller 1996]. Perhaps providing audio clues in our environment could help females.

Potential Barrier 6: Are the females driven away by the colors used in the Forms/3 environment, leading them to not use certain features? Males and females have different preferences towards colors [Radeloff 1990; Green 1995]. Our fault localization feedback involved coloring the interior of the cells with shades of orange on a continuum from yellow to darker shades of orange. Using colors less jarring to females might affect their usage of these features that use colors. This is an item that needs to be studied and further investigated.

Potential Barrier 7: This was derived from our own observations from the previous qualitative studies. Some users perceive changed cell color as having done something wrong rather than following the feedback the system is trying to give.

Potential Barrier 8: Researchers found that boys and girls prefer to work through games in different ways. Rather than working in a linear fashion through the game girls prefer to explore and move freely about a game. (These findings are summarized in [Gorriz and Medina 2000].) If in our environment we include more than one way of doing a particular action, then this might provide support to the females' problem-solving style.

Potential Barrier 9 draws from Potential Barriers 1, 4 and 8. From Potential Barriers 1 and 4, it is implied that the environment should not punish the users for wrong actions with a violent action; instead have an element of forgiveness. Potential Barrier 8 relates to females working in a non-linear fashion, meaning having different alternative solutions for a given task [Gorriz and Medina 2000]. These potential barriers give rise to the need for a safe environment where the users can undo their actions. In our case, these actions

include making decisions about a cell by placing a checkmark or X-mark and editing a cell's formula.

	Potential Barriers	Items to study
1	Overwhelmed by too many colors of Fault Localization feedback? [Ray 2003]	Will fewer colors in FL feedback help?
2	Tool tips – Explanation not serving the purpose	Need to consider the level of abstraction (Reduced help / Abstract / Formal / Concrete) [Arroyo 2003]
3	Environment (working of Fault Localization) not intuitive for females [Ray 2003]	Fundamental working of the system should not be hidden. Interface needs to be extremely intuitive.
4	Punishment – Wrong decision made while placing a checkmark or an X-mark on a cell[Ray 2003]	Robust algorithm to handle mistakes. Result of bad decision – include "element of forgiveness" letting the user to continue with delayed progress.
5	No audio/visual cues [Miller 1996]	Add audio/visual cues to the environment – to explain certain features and while giving feedback
6	Disliking the colors used in Forms/3 environment (specifically WYSIWYT and Fault Localization) [Radeloff 1990; Green 1995]	If there is any preference of females for particular colors or color combinations, maybe use them instead of the ones that the system already has

Table 6: Additional potential barriers and items to study

	Potential Barriers	Items to study
7	Changed cell color might be seen as a result of having done something wrong. Low-confident females might blame themselves for the errors and not look for explanations, which might help lead to solutions, from tool tips.	Prompt an explanation why the color of the cell changed
8	Females prefer to consider more than one solutions to a problem [Gorriz and Medina 2000]	Allow more than one ways to do the same task
9	Safer environment for exploring alternative solutions.	Females not able to restore original formula, after changing it once, thus leaving bugs introduced. Provide an UNDO action in the environment

Table 6 (Continued): Additional potential barriers and items to study

4. From Problem to Solution 1: "No Confidence Required"

From a high-level design perspective, we are dealing with in an "ill-structured" problem. In such problems, formulating the problem and the solution are not entirely separate issues, because each attempt to solve the problem changes the researchers' understanding of the problem. The potential solutions are not well-defined, theory is incomplete, and information upon which a solution can be based is also incomplete.

We performed a claims analysis for each solution in Table 5. Claims analysis [Carroll and Rosson 1992] is a technique for evaluating design solutions where consequences of each solution are identified with respect to the intended users, labeling each consequence as positive or negative. The claims analyses done by our group was instrumental in helping us to choose which solutions to implement first.

One of the barriers in the way of females was low feature usage (Barrier 2 in Table 5 which is also related to Barrier 1). We believed that addressing this barrier first was important since feature usage was tied to effectiveness in debugging which was the main goal of their task and also our claims analysis revealed that this should be the first solution to follow. So we decided to target this barrier first. One possible reason for low feature usage (specifically the usage of checkmarks and X-marks) by the females could have been that they did not place these marks unless they were completely sure about their decision.

Our approach towards the solution to this barrier (which we will call Solution 1) was to provide a way to make decisions about a cell's value even if the users are not completely sure about their decision by expressing their confidence level.

4.1 Prototype design ideas for Solution 1

Solution 1's goal was to communicate to users that they did not have to be confident to be "worthy" of judging the correctness or incorrectness of

values. This involved changes not just to our input device but also to the output device to reflect appropriate feedback based on the input.

4.1.1 Input Device

Our system's input device – a decision box at the top right hand corner of each cell – is a means to make decisions about the cell's value by placing a checkmark or an X-mark. We wanted to re-design it so that the user could express confidence while making decision about the cell's value. The rationale behind the design was to not increase the cost in making decisions by having the users enter the confidence each time they make a decision; the base cost being left click for a checkmark and right click for an X-mark. There were four proposed design ideas for the input device, depicted in Figure 2, Figure 3, Figure 4, and Figure 5.

4.1.1.1 Input Device 1

Figure 2 has a confidence setting at the top of each spreadsheet, allowing the user to set his/her overall confidence in making decisions about any cell's value in the spreadsheet or the confidence of the selected cell's value. If the user selects a particular cell and sets the confidence level, that confidence level applies to that particular cell. If none of the cells are selected then the confidence setting applies to all the cells in the spreadsheet. Initially, every cell in the spreadsheet is associated with the same default confidence value which is high. The possible confidence settings are range of values from 0% to 100% on a numeric scale.

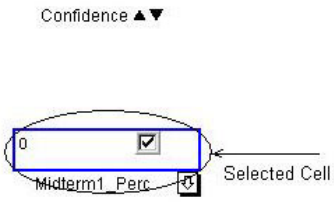

Figure 2: Input Device Design 1

Advantages

- Inputting confidence is optional, (reduced screen real estate) a single confidence widget is added to the top of the spreadsheet while providing the functionality of setting different confidence values for different cells.
- The feature is not hidden and is visible at all the times.
- Flexibility to set the confidence value at any time of making decision (user can set the confidence value either before or after making the decision).

Disadvantages

- Increased cost in setting confidence value for a single cell (need to select the cell in order to set its confidence value).
- The user cannot see the confidence level associated with a particular cell.
- Increased cost of increasing/decreasing confidence if there are too many levels allowed.
- The user may leave a cell selected while wanting to set the confidence of some other cell, thus associating a wrong value of confidence with the cell.

4.1.1.2 Input Device 2

Figure 3 depicts an alternative way of expressing confidence in making decisions about a cell's value. Each cell is associated with a tiny widget just next to its decision box that stands for the confidence level in making decision about that cell's value. The user can increase or decrease the confidence level

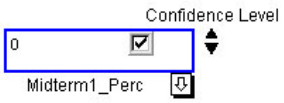

Figure 3: Input Device Design 2

at any point of time. As in the previous design FigureInputDevice1, each cell will have a default confidence value which will be the same for all.

Advantages

- Inputting confidence is optional.
- The feature is not hidden and is visible at all times.
- Flexibility to set the confidence value at any time of making decision (user can set the confidence value either before or after making the decision).

Disadvantages

- Increased screen real estate over present and over Input Device 1.
- The user cannot see the confidence level associated with a particular cell.
- Increased cost of increasing/decreasing confidence if there are too many levels allowed.

4.1.1.3 Input Device 3

Figure 4 provides another way of expressing confidence with a slightly modified widget adjacent to the cell's decision box as compared to FigureInputDevice2. The widget is in the form of a slider bar with 3 levels of confidence (High, Medium and Low). Initially every cell has a default confidence level of "Medium" as shown in the figure.

Advantages

- Inputting confidence is optional.
- The user can clearly see the confidence level of each cell at any point of time.

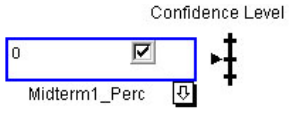

Figure 4: Input Device Design 3

- Flexibility to set the confidence value at any time of making decision (user can set the confidence value either before or after making the decision).

Disadvantages

- Perhaps a difficult design to implement in Forms/3.
- Associating a slider bar with each cell is costly in terms of screen real estate.
- Sliding is a costlier user choice than mere clicking because it involves more physical motion with the user's hand.

4.1.1.4 Input Device 4

Figure 5 combines confidence levels with placing the mark. Specifically, clicking on a cell's decision box pops up a slider with 2 levels of confidence for each of the marks (checkmark and X-mark) – completely sure and not completely sure. The checkmark is associated with "It's right" for a completely confident decision and "Seems right maybe" for a not very confident decision in placing the checkmark, while the X-mark is associated with "It's wrong" for a completely confident decision and "Seems wrong maybe" for a not very confident decision in placing the X-mark.

Advantages

- User can clearly see the confidence level associated with each cell.

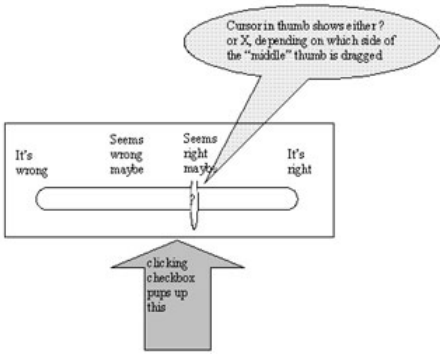

Figure 5: Input Device Design 4

- Reduced screen real estate cost; it is hidden at all the times except when the user clicks on the decision box when this confidence widget pops up.
- Only two levels of confidence associated with each of the marks (LOW and HIGH) reducing the number of choices to make decision from.

Disadvantages

- Increased cost in decision making. The user now has to select one amongst the four available choices as compared to two choices in our original prototype.
- This costs an extra click over the original prototype.
- Perhaps a complex design to implement in Forms/3.

4.1.2 Output Device

The output device was a variation of our previous one. The original design used cell border color to reflect testedness. The border color ranges from red to blue where more blue indicates more testedness of the cell. Cell interior color was used to reflect the fault localization feedback (bug likelihood of the cell). Cell interior color ranges from yellow to orange where more orange means greater likelihood of bugs in the cell. With the new input device, low confidence would result in lower saturation of these colors (50% less saturated than the higher saturation) while high confidence results in higher saturation of the respective colors. For example, low confidence in placing a checkmark results in lower saturation of the cell border color while low confidence in placing an X-mark results in lower saturation of cell interior color.

4.2 Solution 1's Prototype

After brainstorming the above ideas on the input and output devices, a design emerged that drew from the above ideas and was feasible to implement

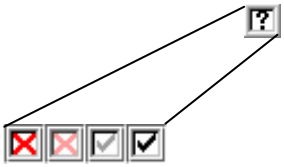

Figure 6: Input Device – Clicking on the checkbox turns it into the four choices. The tool tips over the choices, starting with the left-most X, are "it's wrong," "seems wrong maybe," "seems right maybe," "it's right."

Figure 7: Output Device – Saturation of border color (top) and interior color (bottom) reflect confidence of user judgments of values being correct or incorrect.

and consistent with all other features in the environment. The criteria behind the design were:

- Inputting confidence should be a part of the decision step.
- The user should be able to clearly see the inputted confidence level.
- Low cost in inputting the confidence.
- Low screen real estate – not hidden from the user, should be visible when the user wants to change its value.

Thus, in our prototype, instead of having only two possible actions—checking off or X'ing out values—there are now four possible actions: the original two ("it's right" and "it's wrong") plus "seems right *maybe*" checkmarks and "seems wrong *maybe*" X-marks. See Figure 6. The low saturated marks are for lower confidence judgments, as their tool tips explain.

One small but important detail: another way this change differs from the previous prototype is that in the previous version, the checkmark was done with a left click and the X-mark with a right click. Removing the need for a

right click, which we have observed is not often used by less experienced users, may make X-marks more accessible to those with less experience.

The lower confidence marks result in feedback at lower saturations. That is, a lower confidence checkmark produces lower saturations of border colors reflecting the affected cells' "testedness." Similarly, a lower confidence X-mark produces lower saturations of interior colors reflecting the affected cells' fault likelihood. See Figure 7. Like the increases/decreases in testedness and fault likelihood that arise from the correctness judgments communicated through checkmarks and X-marks, the confidence of these judgments are also propagated to all affected cells.

4.3 Feedback from Users

As the prototype evolved, we brought in eight participants, one at a time, (two males and six females) to use our evolving prototype, in order to inform our design of the prototype changes. Each participant was asked to "think aloud" while working on the same tasks as in [Beckwith et al. 2005]. After they were finished, we interviewed the participants.

Only three participants used the low-confidence marks, but in general the participants did seem to be more willing to make judgments than they had been in previous studies. This change seemed especially apparent with the X-marks. Thus, the changes may have indeed succeeded in communicating the low risk and acceptability of low confidence. However, without a statistical study, we cannot be sure that such a change occurred.

For example, one female (S4) used the approach exactly as we had hoped. Here is what she said while contemplating a cell's value:

> S4 (thinking aloud): "I am not sure if this cell's value is right so maybe I'll mark it gray and come back to it later."

The same female, when asked about the "maybe" marks post-session said that her general tendency was not to mark a cell unless she was completely sure about her decision. She was one of the low confidence females

which we had in mind while addressing this barrier about low feature usage (Barrier 2 in Table 5)

> S4 (interview): "I kinda thought it was right but then I was like... wait a minute I don't have the exact math. I didn't want it to be wrong but I knew that something further along the line was wrong, so I didn't want to put yes for sure even though I thought it was right."
> S4 (interview): "I feel like I shouldn't check them or I shouldn't check anything unless I really thought one way or the other."

The same female when asked if the "maybe" marks would be useful in a complex spreadsheet:

> S4 (interview):"I think the "may be" marks would be useful in a complex spreadsheet. I used it although I wasn't a huge fan of them.. In a huge problem with a lot of aspects, it would make sense to put the "may be" marks"

However S3, a female who did not use the low-confidence marks, later told us she did not see any reward in using them:

> S3 (interview): "I didn't use the "maybe" marks because I thought that they might not help me any more than the other ones in my task."

Some participants used X-marks to keep track of cells that they needed to revisit later. In fact, they may have been even less sure of the values' correctness than we had expected, simply marking the cells whose correctness they wanted to reconsider later.

> S6 (interview): "[X-marks] were a progress marker; just to say that's not right."

S3 also made some revealing comments relating to Barrier 3:

> S3 (interview): "I didn't know what was wrong when it seemed correct to me ...why it showed 50 and not 100 [% tested]."
> Interviewer: "Weren't the tool tips helpful?"

S3 (interview): "Yeah, they were good but sometimes I didn't find the answer that I wanted …I needed more answers than were present."

Comments such as this one pointed us toward the path to Solution 2.

5. Solution 2: Explanations

The addition of low-confidence marks may have helped with the usage of marks, but the evidence is not overwhelming. We decided that, whether or not the low confidence marks were helping, they were probably not helping enough. To strengthen our approach, we decided to tackle Barrier 3 (Table 5) which is also interrelated to Potential Barrier 2 (Table 6), perceived difficulty of learning, via the learning vehicle in the system, explanations.

As pointed out in Section 3.1.2, explanations are a critical part of the Surprise-Explain-Reward strategy [Wilson et al. 2003]. They connect surprises with rewards by providing users with a low-cost mechanism (tool tips) to explore objects that arouse their curiosity. Users can seek explanations for an object by viewing its explanation, on demand, in a low-cost way via tool tips.

Until the work we report here, explanations were as follows: each explanation described the semantics, the action users should try, and a potential reward. They were designed with the goal of encouraging users to learn by doing and to stay connected to the task they were working on when they sought the explanations. Therefore, we kept the explanations short—typically one to three very short lines.

There is literature that says males benefit more from explanations that are fast to check and go through while females prefer to go through any kind of explanation and perform better with those that are highly structured and interactive [Arroyo et. al. 2001] Also females have a positive attitude towards help and towards learning with the system more often than males [Arroyo et. al. 2004]. So while making changes to the prototype to include these explanations, we made sure that these explanations did not get in the way of the users who would not prefer to read explanations. So these explanations should appear only when the user wishes to see them and moreover the user has a choice to select which one they want to seek. The contents of these explanations are small, unlike the conventional help system which details

every minute detail. Instead they are short and limited to just a few lines of text. We made sure that they were short while still covering all the aspects mentioned in sections 5.1 and 5.2.

5.1 Requirements on Types of Explanation Content

We used theory to help develop requirements on the solutions for both Solution 1 and Solution 2. For example, one important influence on the redesign of our explanations' content was the evidence suggesting that the current short explanations may not be well suited to females. According to research in information processing and in education, short explanations such as these are closer matches to the type of information processing and learning environments in which males, not females thrive. [Arroyo 2003; Beck et al. 1999].

As described in detail in [Beckwith et al. VL2005], Anson's essay on minimalist learning theory was a second important influence on Solution 2, [Anson 1998], in which content is described using the terms *conceptual*, *procedural*, and *problem solving*. These terms provide a useful framework for organizing requirements on explanations' content types. We used the term "conceptual" to stand for content relating to concepts and semantics, "procedural" for content about how to perform actions, and "problem solving" for higher-level strategies directed toward "big picture" goals. Together, these terms form completeness requirements for our content *types*; that is, we require explanations to be available with conceptual, procedural, and problem-solving content.

A third influence on Solution 2 was Ko et al.'s work on learning barriers [Ko et al. 2004]. We used these learning barriers to cross-check our list of content type requirements for completeness and to solidify each requirements' aim. We also cross-checked the type requirements' completeness against observation-based user scenarios. The scenarios were observation-based in that they were inspired by user behaviors we have

observed (of both the participants described in Section 4.3 and those in previous studies), in which users were unable to make progress due to barriers they encountered as they were problem solving, such as misunderstanding the system's feedback.

A final influence came from research on learning [Gorriz and Medina 2000]. This work found that females' styles tend to be non-linear (not necessarily sequential in nature), whereas males' tend to be linear (sequential). As a result, we required that our redesigned explanations support both linear and non-linear styles.

5.2 Applying the Requirements

The content type requirements of Section 5.1 led initially to three additional components in the explanations: a "what" component to fulfill the conceptual requirement, a "how should..." component, to fulfill the procedural requirement, and an "advice" component to fulfill the problem-solving

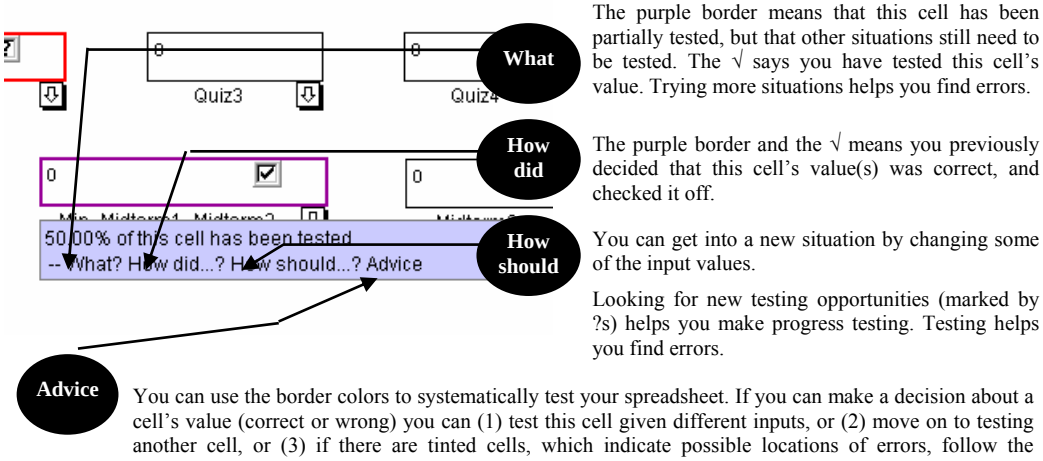

Figure 8: ToolTip Explanations – The top line of the tool tip contains a very short explanation. The expansion components will be clickable via the "What?", "How did...?", "How should...?", and "Advice" labels

requirement. Eventually, we subdivided the conceptual component for clarity of labeling: a "what" component with declarative information and a "how did..." component that explains how the current state came about (emphasizing system responses to user actions). Figure 8 shows an example of a short explanation ("50% of this cell has been tested") and the additional components. The contents of each of these components are derived from theory which is described in more detail in [Beckwith et al. VL2005]. We brought in another stream of users to evaluate this prototype.

5.2.1 Conceptual: The "What" Component

> S7 (thinking aloud): "I don't understand why this [cell] is not 100% tested when it appears to have the right value."

The goal of the "what" component is to communicate the semantics of the object in more detail than the short explanation:

> *The purple border means that this cell has been partially tested, but that other situations still need to be tested. The √ says you have tested this cell's value.*
>
> It explains in details what the present state of the feature meant.

5.2.2 Conceptual: The "How did…" Component

> S8 (thinking aloud): "...how did I do that?"

The "how did" component explains what steps the system or user took to get the object to its current state:

> *The purple border and the √ means you previously decided that this cell's value(s) was correct, and checked it off.*
>
> It explains how the user reached this state in the environment.

For S8, who proceeded to open this component in order to answer her question above, the "how did…" content provided her with the information she needed:

> S8 (thinking aloud): "Oh yeah, I should test it more."

5.2.3 Procedural: The "How should…" Component

S8 (thinking aloud): "How should I test it more?"

The "how should…" component suggests action(s) users can take to make progress on their task:

> *You can get into a new situation by changing some of the input values. Looking for new testing opportunities (marked by ?s) helps you make progress testing.*

It explains how the user should proceed from the present state in order to make progress.

When the participants were using the prototype one of the participants (following up from the previous dialog on the "how did…") realized she needed to do more testing from understanding what the purple border meant. She then went on to read the "how should…" explanation.

5.2.4 Problem Solving: The "Advice" Component

The "advice" component provides ideas about higher-level strategies to achieve the "big picture" goals. One of the purposes is to help orient the user to this feature within the context of their overall task.

> *You can use the border colors to systematically test your spreadsheet. If you can make a decision about a cell's value (correct or wrong) you can (1) test this cell given different inputs, or (2) move on to testing another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the system's guidance (cells with darkest tints) to find the causes(s).*
>
> *Border colors reflect the number of √s on this or related cells, and tints on the entire cell reflect the number of Xs (in relation to the number of √s) on this or related cells.*

It suggests additional advice on how to proceed explaining a few strategies in more details. We will see in Section 5.4 below how one of the female subjects asked for the "Advice" explanation when she was stuck.

5.3 Solution 2's Prototype

Users of our low-cost prototype experienced the new components primarily in the form of paper augmentations to our executable prototype, as shown in Figure 9. As mentioned in Section 3, each feature in the Forms/3 environment is associated with a tooltip. With our new explanations, each tooltip had our four explanation components (What?, How did…?, How should…?, Advice) just below the main tooltip contents as shown in Figure 8. The keys F1, F2, F3 and F4 were associated with What, How did, How should

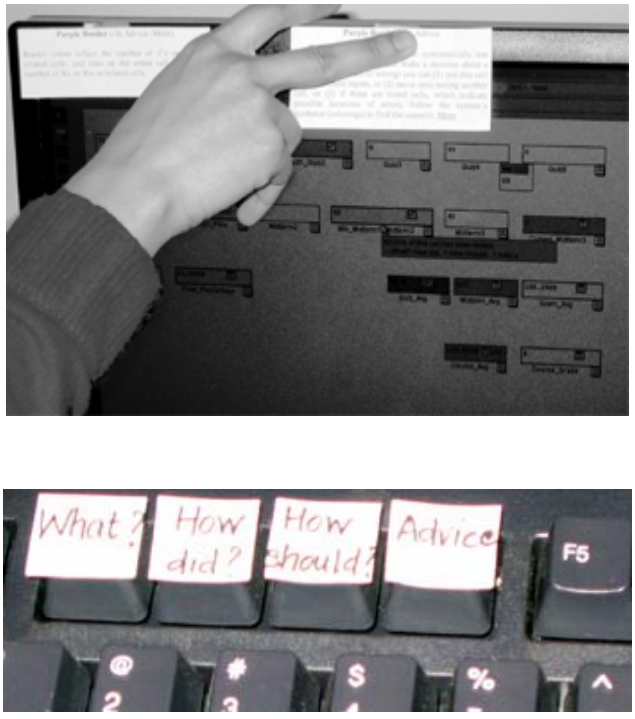

Figure 9: Low cost-prototype with paper augmentations – In our low-cost prototype, the user's request for an additional explanation component (bottom) caused the examiner to add it to the screen (top). Note the support for non-linear approaches—a user can view many unrelated components simultaneously.

and Advice explanation components respectively.

Paper labels were glued to these keys to remind the user what each of them meant. Whenever the user wanted more explanation on any of the components, he/she would press the respective key and the respective paper augmentation of that explanation component was put on top of the screen as shown in Figure 9. The explanation remained there as long as the user wanted them to be there and were removed when they no longer needed it. They could ask for more than one explanation at a time, so that there could be multiple paper augmentations on top of the screen at a time.

5.4 Feedback from Users

The low confidence female quoted in Section 5.2.1 who did not understand purple border color:

> S7 (thinking aloud): ...still there is some problem. I don't understand what to do!"

She then referred to more explanation on "What?", "How should…?" and "Advice".

A high confidence female, after asking why she did not use the explanations, reported that she would prefer to explore things on her own and would refrain from asking for help unless she was completely stuck and could not progress.

The high confidence female quoted in Section 5.2.2 who did not understand what the purple border colors meant:

> S8 (thinking aloud): "Why is it still purple? ...is something missing?"

She used the explanation "What" on purple border colors. It reminded her about what the purple border color meant, then she asked for "How did…?" which explained her how she reached that state. After reading this explanation:

> S8 (thinking aloud): "Oh ya, I should test it more... how should I test it more?"

She then asked for more explanation on “How should...?” which explained her how she can make progress by changing the input values.

The explanations might have helped the females in making progress, but we cannot be very sure of this. However, females considered seeking more explanations when stuck more often than the males.

Users who understood the system very well did not need explanations to help them make progress, so they never used the explanations unless they were stuck completely and could not make progress. These were mostly the ones with high confidence. However, users who did not understand some aspect of the system referred to these explanations often. These were mostly the ones with low confidence.

6. Think-aloud Analysis and Final Implementation

6.1 Quick and dirty evaluation

After the low cost prototype evaluation with paper augmentations, another intermediate evaluation was done with a slight modification in the prototype replacing the lightweight tooltips with tooltip explanations in the form of internal frames as shown in Figure 10. We invited a male and a female

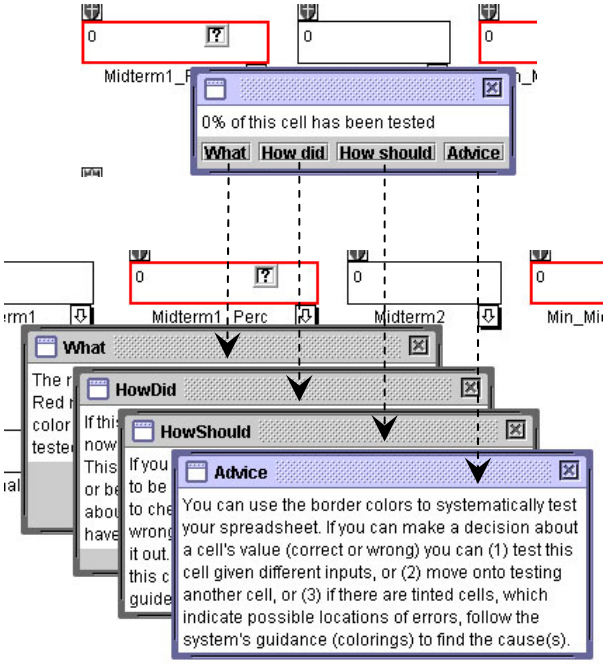

Figure 10: Internal frame explanations implemented for think-aloud

subject to get feedback on whether the explanations were serving the purpose. Both of them were led through the same tutorial as mentioned in section 5 and were given the same spreadsheet task to work and were asked to test the given spreadsheet and find and fix any errors. The tasks were the same one as given in the earlier think aloud studies.

One of the subjects mentioned in post-session interview:

S9: “They [explanations] could be helpful if the wordings were in layman’s terms”

The other subject mentioned that she was looking for more explicit explanations:

S10: “These [explanations] don’t really help..., they don’t really tell you what to do...” “...something like Cell B is causing Cell A to be purple...would have helped”

As seen in Figure 10, the additional explanation on the tooltip can be expanded or hidden as per the user’s choice. The user may choose not to expand this part at all. So it is completely optional to refer to them. The explanations remain on the screen as long as the user wishes to keep them. The user can discard them by simply closing the internal frame.

The feedback from the users revealed that the contents of the tooltip explanations were not really helping. So we decided in the absence of useful user feedback about how to proceed with four different components for the tooltip explanations, we will start simple, with just one additional line of text with the heading “Tips” which gave them additional tips on how to make progress.

A modification to the earlier design was made with the help of other team members[5]. The tooltip explanations in this case were implemented as a part of Java tooltips Figure 11 in order to reduce the “weight” (cost and screen real estate) of the tooltips. These tooltips had an expander named “Tips” (implemented as a Java Tree) which could be expanded by clicking it. The tooltip could be discarded by again clicking on this expander. On expanding the “Tips” portion of the tooltip, the additional tooltip explanation was appended to the main tooltip as shown in figure Figure 11.

---

[5] Java ToolTip Explanations implemented with the help of Joseph Lawrence, assisted by Andrew Stucky and Marc Fisher.

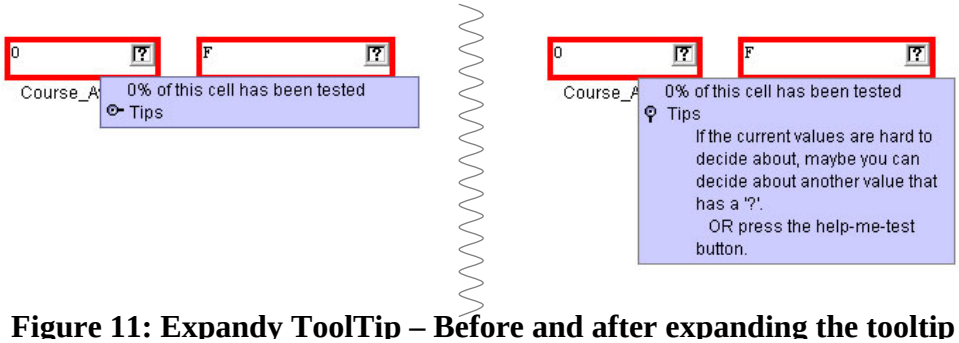

Figure 11: Expandy ToolTip – Before and after expanding the tooltip

6.2 Some interesting observations

The three think aloud evaluation studies conducted revealed some interesting observations which are discussed in this section.

6.2.1 Unintended usage

Most of the females seemed to use X marks for keeping track of the cells.

> S6 (interview): "[X-marks] were a progress marker; just to say that's not right."

Also, they used the low confidence marks for keeping track while their ultimate goal was to achieve the darker color of testedness.

> S4 (thinking aloud): "I am not sure if this cell's value is right so maybe I'll mark it gray and come back to it later."

Another interesting observation on how some users made decisions was, instead of placing a mark based upon a cell's value, they used formulas to base their decision in placing a mark. This behavior is also noted down in [Phalgune et al. 2005].

> S10: "The formula is correct, so I'll mark it right"

6.2.2 Pattern of debugging

Females almost always used a systematic way of testing. Their testing approach was to follow Western reading order, proceeding from one cell to the

other from the top left cell towards the right, row after row, and not just randomly select cells to test.

These females used the Western reading order for the task, and so the interior cell colors did not help them much in debugging. Perhaps if they had used a sink to source order (proceeding from the bottom cells to the top cells), they would have found the interior colors useful which was a means of feedback for fault localization. It would be interesting to see what approach males follow in their debugging task.

6.2.3 More observations

Based on the observations of the behaviors of our subjects, there were two categories of users depending on how they tested the cells.

- Category 1 users checked the cell's values first and placed the marks without looking at the formula. These were more likely to place X-marks along with the checkmarks.
- Category 2 users made formula changes without bothering to place the marks. These were unlikely to place X-marks. However they placed check marks after the formula change.

Most of the subjects seemed to use X-marks more often than they did in our previous studies. Of course, these were just a handful of them but we expect to see similar results in our main experiment. Again this needs to be verified by the main experiment. Possible reasons for this behavior could be:

- They no longer needed to right click in the modified prototype version in order to place an X-mark.
- They had both the options (checkmark and X-mark) clearly visible while making the decision.

It seemed that the users would not use the low confidence marks unless they got any reward. So the idea that females would use more marks if they found a way to express their confidence level did not seem effective as least in a direct way. Although there was a female, S4 who used the low confidence

marks in exactly the way we had predicted. We could tie some kind of reward for using the low confidence marks, which gave them more support than if they used the high confidence marks.

Some subjects were overwhelmed by the complex formulas in the payroll spreadsheet task. This is especially true of the users with low mathematical ability. They tried to run away from formula edits / changes. One such female subject S11 with poor mathematical ability tried to evade formula changes. She was a Category 1 user as described above, who used all the other features except for formula edits to get the spreadsheet tested. She did not want to play with formulas.

S11 (interview): "Such big formulas/equations blew me away!"

Many users feared that if they changed a formula they might not be able to get it back to the original form, so they did not edit many formulas unless they were completely sure. While all the other features (checkmarks, X-marks, arrows) can be undone formula edits cannot be undone. So the users do not feel safe to edit formulas. This lack of "undo" could be impacting our experiments' results; spending the resources to add an undo should be considered.

Most of the users got carried away by the percent testedness of the spreadsheet, so the main task may not have been finding and fixing bugs accurately but instead getting all the cells borders to blue so that the spreadsheet is 100% tested and the testedness progress bar shows 100%.

When stuck or when not able to make progress, females sought help of explanations. However, the textual contents of these explanations did not seem to be of much help to them and hence they later stopped looking at explanations.

Quite a few of our subjects fixed all the bugs. However, they could not get the spreadsheet 100% tested and they did not know how to get it 100%

tested, and that's where they were stuck most of the time. Most of these were stuck with purple border color.

Most of the users tried to change the formula and expected some kind of color changes indicating positive progress after they were done with the formula change. They expected the system to give some feedback in terms of color changes after they hit the "Apply" button in the formula for a cell. This could be because they were confused or because they did not understand the system thoroughly. And if the system did not give them feedback after a formula change, they thought that perhaps the change was not right.

A check mark placed before formula change was often not quite so confident while a check mark placed after a formula change was always confident.

Some users assumed that whatever the system does is always right. This was true of most of the low confident users. So if they expected color changes after a formula edit and the system did not change, they thought that their change was not right. There were certain users like S4 who thought that all the features present in the system serve some purpose and hence are built into the system, otherwise, they wouldn't have been present.

S4 (interview): "They [confidence marks] are there because they make some sense."

Research in gaming and software design [Miller 1996] indicates that females preferred moving freely among environments without completing or winning one. This was particularly observed of a female subject S4. Although she started looking at cells systematically, she did not want to be stuck at any one particular cell (described in Section 6.2.1), so she wanted to go to some other cell and marked the earlier cell so that she can get back to it later.

S4 (thinking aloud): "I am not sure if this cell's value is right so maybe I'll mark it gray and come back to it later."

7. Conclusion

This thesis described our investigation into gender issues in end-user software engineering environments. We used theory and previous empirical work to derive specific hypotheses related to gender issues in such environments, and to investigate whether these hypothesized issues really do arise in end-user software engineering. The empirical result of the previous step was confirmation that two hypothesized gender issues: 1. There will be gender differences in users' interest in (initially) exploring new features in end-user programming environments and 2. Females' high perceptions of risk will render them less likely to make (genuine) use of unfamiliar devices in end-user programming environments, indeed exist in end-user software engineering. The next step, reported in this thesis, was to develop solutions to address these issues.

Our work resulted in two complementary solutions: a single-mouse-button "no confidence required" device to elicit inputs from low-confidence users that were then reflected in the feedback devices, and changes to our explanation system to support user-driven, non-linear exploration of the end-user software engineering devices in the system.

Our procedure for developing these solutions used theory, low-cost prototyping, and qualitative empirical work. Specifically, we showed how theories such as self-efficacy theory, attention investment, etc. can be used to help understand barriers, derive requirements, and ultimately derive design ideas to address gender issues in end-user software engineering. Using the theory-derived design ideas, coupled with design techniques originally developed in HCI, we then designed the potential specifics of our solutions, evaluated them analytically and through rapid prototyping, and informed our emerging approaches with a small stream of users. The solutions that resulted are the first to begin addressing gender differences through the design of features in end-user software engineering environments. As discussed earlier,

we were dealing with an “ill-structured” problem where it is not possible to formulate the problem and solution independently. We used the best HCI techniques with a combination of Claims Analysis and low-cost prototyping to design our potential solution.

BIBLIOGRAPHY

[Altizer et al. 1996] Altizer, M., Sen T. K., and Tegarden, D., Gender Differences in Decision Making, *Association for Information Systems Conference Proceedings*, Phoenix, 1996.

[Anson 1998] Anson, P., Exploring minimalistic technical documentation design today: a view from the practitioner's window, In J. M. Carroll (Ed.), Minimalism Beyond the Nurnberg Funnel, Cambridge, MA: MIT Press, 1998, 91-117.

[Arroyo 2003] Arroyo, I., Quantitative evaluation of gender differences, cognitive development differences and software effectiveness for an elementary mathematics intelligent tutoring system, PhD Thesis, Univ. Mass. Amherst 2003, http://ccbit.cs.umass.edu/people/ivon/Dissertation80.pdf

[Arroyo et al. 2001] Arroyo, I., Beck, J. E., Beal, C. R., Rachel E., Wing, and Woolf, B. P., Analyzing students' response to help provision in an elementary mathematics Intelligent Tutoring System. Help Provision and Help Seeking in Interactive Learning Environments, *Workshop at the Tenth International Conference on Artificial Intelligence in Education*, San Antonio, TX, May 2001.

[Arroyo et al. 2000] Arroyo, I., Beck, J. E., Woolf, B. P., Beal, C., Schultz, K., Macroadapting AnimalWatch to gender and cognitive differences with respect to hint interactivity and symbolism, *Fifth International Conference on Intelligent Tutoring Systems*, 2000.

[Arroyo et al. 2004] Arroyo, I., Murray, T., Woolf, B. P., Beal, C. R., Inferring Unobservable Learning Variables from Students Help Seeking Behavior. James C. Lester, Rosa Maria Vicari, Fábio Paraguaçu (Eds.): Intelligent Tutoring Systems, *Seventh International Conference*, ITS 2004, Maceiò, Alagoas, Brazil.

[Beck et al. 1999] Beck, J. E., Arroyo, I., Woolf, B. P., and Beal, C., An ablative evaluation, *Ninth International Conference on Artificial Intelligence in Education*, 1999, 611-613.

[Beckwith and Burnett 2004] Beckwith, L. and Burnett, M., Gender: An important factor in end-user programming environments? *IEEE Symposium on Visual Languages and Human-Centric Computing*, 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 software features: Are there gender issues? *ACM Conference on Human Factors in Computing Systems*, Portland, Oregon, April 2005, 869-878.

[Beckwith et al. VL2005] Beckwith L., Sorte S., Burnett M., Wiedenbeck S., Chintakovid T., and Cook C., Designing Features for Both Genders in End-User Software Engineering Environments, *IEEE Symposium on Visual Languages and Human-Centric Computing*, Dallas, TX, to appear, September 2005.

[Blackwell 2002] Blackwell, A., First steps in programming: a rationale for Attention Investment models, *IEEE Symposium on Human-Centric Computing Languages and Environments*, 2002, 2-10.

[Burnett et al. 2001] Burnett, M., Atwood, J., Djang, R., Gottfried, H., Reichwein, J., and Yang, S., Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm, *Journal of Functional Programming*, 11(2), 2001, 155-206.

[Burnett et al. 2004] Burnett, M., Cook, C. and Rothermel G., End-user software engineering, *Communications of ACM*, 47(9), 2004, 53-58.

[Camp 1997] Camp, T., The incredible shrinking pipeline, *Communications of ACM*, 40(10), 1997, 103-110.

[Carroll and Rosson 1992] Carroll, J. M. and Rosson, M. B., Getting around the task-artifact cycle: how to make claims and design by scenarios, *ACM Transactions on Information Systems*, 10(2), 1992, 181-212.

[Fennema and Sherman 1977] Fennema, E., and Sherman, J., Sex-related differences in mathematics achievement, spatial visualization, and affective factors, *American Educational Research Journal*, 14, 1977, 51-71.

[Fennema et al. 1998] Fennema, E., Carpenter, T., Jacobs, V., Franke, M., Levi, L., A Longitudinal Study of Gender Differences in Young Children's Mathematical Thinking, *Educational Researcher*, 27(5), 1998, 6-11.

[Gorriz and Medina 2000] Gorriz, C., and Medina, C., Engaging girls with computers through software games, *Communications of ACM*, 43(1), 2000, 42-49.

[Green 1995] Green, K. S., Blue versus periwinkle: Color identification and gender, *Perceptual and Motor Skills*, 80(1), 1995, 21-32.

[Greenberg 1993] Greenberg, S., The Computer User as Toolsmith: The Use, Reuse, and Organization of Computer-based Tools, New York: Cambridge University Press, 1993.

[Hartzel 2003] Hartzel, K., How self-efficacy and gender issues affect software adoption and use, *Communications of ACM*, 46(9), 2003, 167-171.

[Huff 2002] Huff, C., Gender, software design, and occupational equity, *ACM SIGCSE Bulletin*, 34 (2), 2002, 112-115.

[Huff and Cooper 1987] Huff, C., and Cooper, J., Sex bias in educational software: The effect of designers' stereotypes on the software they design, *Journal of Applied Social Psychology*, 17(6), 1987, 519-532.

[Inkpen et al. 1994] Inkpen, K., Klawe, M., Lawry, J., Sedighian, K., Leroux, S., and Hsu, D., We have never-forgetful flowers in our garden: Girls' responses to electronic games, *Journal of Computers in Mathematics and Science Teaching*, 13, 1994, 383-403.

[Ko et al. 2004] Ko, A. J., Myers, B. A., and Aung, H. H., Six learning barriers in end-user programming systems, *IEEE Symposium on Visual Languages and Human-Centric Computing*, 2004, 199-206.

[Meyers-Levy and Maheswaran 1991] Meyers-Levy, J., and Maheswaran, D., Exploring differences in males' and females' processing strategies, *Journal of Consumer Research*, 18, 1991, 63-70.

[Miller 1996] Miller L., Girls' preferences in software design: Insights from a focus group, *Interpersonal Computing and Technology: An Electronic Journal for the 21st Century*, 4(2), 1996, 27-36

[Phalgune et al. 2005] Phalgune A., Kissinger C., Burnett M., Cook C., Beckwith L., and Ruthruff J., Garbage In, Garbage Out? An Empirical Look at Oracle Mistakes by End-User Programmers, *IEEE Symposium on Visual Languages and Human-Centric Computing*, Dallas, TX, September 2005.

[Quinn and Spencer 2001] Quinn, D., and Spencer, S., The Interference of Stereotype Threat with Women's Generation of Mathematical Problem-solving strategies, *Journal of Social Issues*, 57(1), 2001, 55-71.

[Radeloff 1990] Radeloff, D. J., Role of color in perception of attractiveness, *Perceptual and Motor Skills*, 71, 1990, 151-160.

[Ray 2003] Ray, S., Females and Machines, In Gender Inclusive Game Design: Expanding the Market, Charles River Media, September 2003.

[Robertson et al. 2004] Robertson, T. J., Prabhakararao, S., Burnett, M., Cook, C., Ruthruff, J., Beckwith, L. and Phalgune, A., Impact of interruption style on end-user debugging, *ACM Conference on Human Factors in Computing Systems*, 2004, 287-294.

[Rothermel et al. 2001] Rothermel, G., Burnett, M., Li, L., Dupuis, C., and Sheretov, A., A methodology for testing spreadsheets, *ACM Transactions on Software Engineering and Methodology*, 10(1), 2001, 110-147.

[Ruthruff et al. 2003] Ruthruff J., Creswick E., Burnett M., Cook C., Prabhakararao S., Fisher II M., and Main M., End-User Software Visualizations For Fault Localization, *ACM Symposium on Software Visualization*, 2003, 123-132.

[Ruthruff et al. 2004] Ruthruff, J., Phalgune, A., Beckwith, L., Burnett, M. and Cook, C. Rewarding 'good' behavior: End-user debugging and rewards, *IEEE Visual Languages and Human-Centric Computing*, 2004, 115-122.

[Ruthruff et al. 2005] Ruthruff, J., Prabhakararao, S., Reichwein, J., Cook, C., Creswick, E., and Burnett, M. Interactive, visual fault localization support for

end-user programmers, *Journal of Visual Languages and Computing*, 16(1-2), 2005, 3-40.

[Wilson et al. 2003] Wilson, A., Burnett, M., Beckwith, L., Granatir, O., Casburn, L., Cook, C., Durham, M., and Rothermel, G., Harnessing curiosity to increase correctness in end-user programming, *ACM Conference on Human Factors in Computing Systems*, 2003, 305-312.

APPENDICES

APPENDIX A

Survey Questionnaire

1. What is your age? ☐ <16 ☐ 16-18 ☐ 19-21 ☐ 22-24 ☐ 25-27 ☐ >27

2. What is your gender? ☐ Female ☐ Male

3. What is your Major? _____ ☐ Undeclared

4. How often do you use computers in your activities?
☐ Less often than once a month ☐ Once a month ☐ Once a week ☐ Once a day ☐ Multiple times a day

NOTE: While ranking options, rate them in descending order of priority leaving those blank that you would not consider at all. (1- Most prioritized, 2- second most prioritized …)

5. What types of software do you use frequently? (Rank in order of your frequency of use)
___Web Browser (Internet Explorer … etc.)
___Email (AOL, Hotmail, Yahoo … etc.)
___Chat (MSN, Yahoo …etc.)
___Search Engines (Google … etc.)
___News services (CNN … etc.)
___Games
___Word Processor (MS Word …etc.)
___Spreadsheet (Excel …etc.)
___Programming environments (Visual Basic, C, C++, Java …etc.)
___Others _____

6. What features do you like in software? (Rank in order of priority – leaving those blank that you do not consider at all)
___Testing/Debugging features (that allow testing/tracing errors)
___Programming features (macros, email rules … etc.)
___Communication features (video conferencing, discussion board, chat … etc.)
___Features to customize games
___Searching features
___Language assistance features (spell checking, grammar checking, thesaurus/dictionary… etc.)
___Multimedia views of information (animation, pictures/images, audio/video content)
___Help features (manuals, tutorials, wizards, tool tips… etc.)

7. Did you use Powerpoint for your presentation? Why did/didn't you use it?

_____

_____

_____

_____

Figure 1: Survey Questionnaire

How much do you agree with the following?

8. I mainly use computers when I am in a group (friends …etc.)
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

9. I work independently on most of my computer work.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

10. Software helps me perform my task more quickly.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

11. Software is difficult to understand.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

12. I avoid working with new software since it requires more time to learn.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

13. I avoid working with new software since it requires me to think more.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

14. I find that most software is self-explanatory.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

15. If something goes wrong with the software (like the program crashes), I believe I can fix it.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

16. I wish there was more graphical content (animation, pictures, multimedia content) in (non-website) software.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

17. I wish there was less graphical content (animation, pictures, multimedia content) in (non-website) software.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

18. I prefer software that provides more than one way of achieving the same result.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

19. I prefer playing games that can tell me how I am ranked amongst other people.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

20. If something goes wrong with the software: (Rank in order of priority – leaving those blank that you do not consider at all)
___I try to fix it
___I ignore it
___I stop using the software
___I seek help from someone to fix it

21. When I have problems using software, I refer to: (Rank in order of priority – leaving those blank that you do not consider at all)
___A manual/tutorial/online help
___A technical expert
___A friend
___My own ability to figure things out

Figure 1 (Continued): Survey Questionnaire

All software comes with a set of features that help you accomplish your task effectively. (E.g. Email rules, advanced search, spell check, etc.) How much do you agree with the following regarding these features?

22. I am often not sure how these features work.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

23. I am usually confident that I understand the functionality of these features.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

24. I don't use these features much and ignore them if prompted.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

25. I am comfortable changing the settings of these features.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

26. I often change their settings as per my work requirements.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

27. They simplify my task, making work easier.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

28. I can get my work done faster using these features.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

29. These features help me overcome my weaknesses.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

30. I explore these features primarily when I am in a group of other people (e.g. friends …etc.)
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

31. I enjoy exploring new features provided with the software.
☐ Disagree ☐ Somewhat Disagree ☐ Neither Agree nor Disagree ☐ Somewhat Agree ☐ Agree

32. I started using these features when: (Rank in order of priority – leaving those blank that you do not consider at all)
___I saw someone using it
___A friend told me about them
___A friend showed me how to use them
___Playing around/accidentally ran into them
___I got frustrated not having them & went looking for them
___I read the manual/tutorial

Figure 1 (Continued): Survey Questionnaire

APPENDIX B

Tutorial for Think-aloud 2

Introduction

Hi, my name is Shraddha Sorte, and I will be leading you through today's study.

The other people involved in this study are Dr. Margaret Burnett, Laura Beckwith, and Dr. Curtis Cook.

Just so you know, I'll be reading through this script so that I am consistent in the information I provide you and the other people taking part in this study, for scientific purposes.

The aim of our research is to help people create correct spreadsheets Past studies indicate that spreadsheets contain several errors like incorrectly entered input values and formulas. Our research is aimed at helping users find and correct these errors.

For today's experiment, I'll lead you through a brief tutorial of Forms/3, and then you will have a few experimental tasks to work on.

But first, I am required by Oregon State University to read aloud the text of the "Informed Consent Form" that you currently have in front of you:
(Read form).

Please do NOT discuss this study with anyone. We are doing later sessions and would prefer the students coming in not to have any advance knowledge.

Questions?

Contact:

- Dr. Margaret Burnett burnett@cs.orst.edu
- Dr. Curtis Cook cook@cs.orst.edu

Any other questions may be directed to IRB Coordinator, Sponsored Programs Office, OSU Research Office, (541) 737-8008

Background Questionnaire *(hand it out, have them fill it out)*
(please do NOT turn to any other pages until you are asked to do so)

Tutorial

Before we begin, I'd like to ask if you are colorblind. We will be working with something that requires the ability to distinguish between certain colors, and so we would need to give you a version that does not use color.

Think Aloud Practices:

In this experiment we are interested in what you say to yourself as you perform some tasks that we give you. In order to do this we will ask you to TALK ALOUD CONSTANTLY as you work on the problems. What I mean by talk aloud is that I want you to say aloud EVERYTHING that you say to yourself such as what you are thinking. Just act as if you are alone in this room speaking to yourself. If you are silent for any length of time, I will remind you to keep talking aloud. It is most important that you keep talking. Do you understand what I want you to do?

Good. Before we turn to the real experiment and the tutorial, we will start with a couple of practice questions to get you used to with talking aloud. I want you to talk aloud while you answer the question.

How many windows are there in your parent's house?

Another practice question for you to talk aloud.

Name the states that begin with the letter "A" which you can ski in.

In this experiment, you will be working with the spreadsheet language Forms/3. To get you familiarized with the features of Forms/3, we're going to start with a short tutorial in which we'll work through a sample spreadsheet problem. After the tutorial, you will be given a spreadsheet; asked to test it, and correct any errors you find in it.

As we go through this tutorial, I want you to ACTUALLY PERFORM the steps I'm describing. When I say, "click", I'll always mean click the left mouse button once unless I specify otherwise. I will be very clear regarding what actions I want you to perform. Please pay attention to your computer screen while you do the steps.
If you have any questions, please don't hesitate to ask me to explain.
For that spreadsheet that we will be working with, you will have a sheet of paper describing what the spreadsheet is supposed to do.

(Hand out PurchaseBudget Description)

Read the description of the "PurchaseBudget" spreadsheet now.

(Wait for them to read)

Now open the PurchaseBudget spreadsheet by selecting the bar labeled PurchaseBudget at the bottom of the screen with your left mouse button.

This is a Forms/3 spreadsheet. There are a few ways that Forms/3 spreadsheets look different than the spreadsheets you may be familiar with:
Forms/3 spreadsheets don't have cells in a grid layout. We can put cells anywhere *(select and move a cell around a bit)*. However, just like with any other spreadsheet, you can see a value associated with each cell.

We can give the cells useful names like PenTotalCost *(point to the cell on the spreadsheet)*.
You can also see that some cells have red borders.

Let's find out what the red color around the border means. Rest your mouse on top of the border of the PenTotalCost cell *(wave the mouse around the cell and then rest mouse on border)*. Note that a message will pop up that tells us what this color means. Can you tell me what the message says? *(PAUSE, look for a hand.)* Yes, it means that the cell has not been tested. You can also get more information, such as: "What does this mean?", "How did it happen?", "How should I proceed?", and "Advice". Try clicking on one of these.

You might be wondering what does testing have to do with spreadsheets? Well, it is possible for errors to exist in spreadsheets, but what usually happens is that they tend to go unnoticed. It is in our best interest to find and weed out the bugs or errors in our spreadsheets so that we can be confident that they are correct.

So, the red border around the cells tells us that the cell has not been tested. It is up to us to make a decision about the correctness of the cell's value based on how we know the spreadsheet should work. In our case, we have the spreadsheet description that tells us how it should work.

Observe that the Pens and Paper cells have a black border color *(wave mouse around cells)*. Such cells with black borders are like this because their formulas are constant values.

Let's test our first cell. To do this, we'll examine the TotalCost cell. Is the cell's value of zero correct? *(PAUSE for a second)*. Well, let's look at our

spreadsheet description. Look at the Total Cost section of the spreadsheet. It says, “The total cost is the combined cost of pens and paper.” Well, both PenTotalCost and PaperTotalCost are zero, so TotalCost appears to have the correct value.

Now drag your mouse over the small box with a question mark in the upper-right-hand corner of the cell. Can you tell me what the popup message says? *(PAUSE, wait for answer.)* Yes, it says that if you can decide if this value is correct or wrong, click. It also tells us that these decisions help test and find errors.

Click the question mark in this decision box for TotalCost. Hey, there are 4 choices here – 2 X marks and 2 check marks. Can you read aloud the popup messages on each of the check mark boxes and tell me what they say (starting from the left)? (*Pause*) Yes, starting from the left the popup messages say, “It’s wrong”, “Seems wrong maybe”, “Seems right maybe” and “It’s right”. Now, we know that the value in this cell is right, so we will focus on the checkmarks. Click on the rightmost check mark and see what changes happen. Three things changed. A checkmark replaced the question mark in the decision box *(wave mouse)*. The border colors of some cells changed—three cells have blue borders instead of red, and the percent testedness indicator changed to 28% *(point to it)*. Forms/3 lets us know what percent of the spreadsheet is tested through the percent testedness indicator. It is telling us that we have tested 28% of this spreadsheet.

What about that other checkmark that we saw? We’ll try that one, click on the check mark to UNDO the changes and bring the question mark back. Now click on the question mark to bring the other choices back again. Now click on the other check mark (the left one) and see what happens. ***(Pause)*** Now if you

accidentally place a checkmark in the decision box, if the value in the cell was really wrong, or if you haven't seen the changes that occurred, you can "uncheck" the decision about TotalCost by clicking on that checkmark in TotalCost's decision box. ***(Try it, and Pause)*** Everything went back to how it was. The cells' borders turned back to red, the % testedness indicator dropped back to 0% and a question mark reappeared in the decision box.

Since we've already decided the value in the TotalCost cell is correct, we want to retell Forms/3 that this value is correct for the inputs. So click in the decision box for TotalCost to put either of the 2 check marks back.

You may have noticed that the border colors of the PenTotalCost and PaperTotalCost cells are both blue. Now let's find out what the blue border indicates by holding the mouse over the PenTotalCost cell's border in the same way as before. The message tells us that the cell is fully tested. *(PAUSE)* Also notice the blank decision box in the PenTotalCost and PaperTotalCost cells. What does that mean? Position your mouse on top of the box to find out why it is blank. A message pops up that says we have already made a decision about this cell. But wait, I don't remember us making any decisions about PenTotalCost or PaperTotalCost. How did that happen?

Let's find out. Position your mouse to the TotalCost cell and click the middle mouse button. Notice that colored arrows appear. Click the middle mouse button again on any one of these arrows—it disappears. *(PAUSE)* Now, click the middle mouse button again on TotalCost cell—all the other arrows disappear. Now bring the arrows back again by re-clicking the middle mouse button on TotalCost.

Move your mouse over to the top blue arrow and hold it there until a message appears. It explains that the arrow is showing a relationship that exists between TotalCost and PenTotalCost. The answer for PenTotalCost goes into or contributes to the answer for TotalCost. *(PAUSE)*

Oh, ok, so does that explain why the arrow is pointed in the direction of TotalCost? Yes it is, and it also explains why the cell borders of PenTotalCost and PaperTotalCost turned blue. Again, if you mark one cell as being correct and there were other cells contributing to it, then those cells will also be marked correct. *(PAUSE)* We don't need those arrows on TotalCost anymore, so hide them by middle-clicking on the TotalCost cell.

Now, let's test the BudgetOk cell by making a decision whether or not the value is correct for the inputs. What does the spreadsheet description say about our budget? Let me go back and read…oh yeah, "You cannot exceed a budget of $2000".

This time, let's use the example correct spreadsheet from our spreadsheet description to help us out. Let's set the input cells of this sheet identical to the values of our example correct spreadsheet in the spreadsheet description. The Pens cell is already zero. But we need to change the value of the Paper cell to 400 so it matches the example spreadsheet in the description. How do I do this? Move your mouse to the Paper cell and rest the mouse cursor over the little button with an arrow on the bottom-right-hand side of the cell. It says "Click here to show formula." Let's do that by clicking on this arrow button. A formula box popped up. Change the 0 to a 400, and click the Apply button. I think I'm done with this formula, so hide it by clicking on the "Hide" button. Moving on, in this example correct spreadsheet, PensOnHand is 25, and

PaperOnHand is 21. (*Wave paper around*) Oh good, the spreadsheet already has these values, so we don't have to change anything.

Now, according to this example correct spreadsheet, BudgetOk should have the value "Budget Ok". But it doesn't; my spreadsheet says "Over Budget". So the value of my BudgetOK? cell is wrong. What should we do?

Remember, anytime you have a question about an item of the Forms/3 environment, you can place your mouse over that item, and wait for the popup message. To remind us what the question mark means, move your mouse to the BudgetOk decision box. The popup message tells us that if you can decide if this value is correct or wrong, click and also that these decisions help you test and find errors. Well, this value is wrong, so go ahead and click on the question mark. But wait, there are 2 X marks. Can you read aloud the popup messages on each of the X mark boxes and tell me what they say? (*Pause*)Yes, the leftmost message says, "It's wrong" and the other message says "Seems wrong, maybe". Now, click on the leftmost X mark and see what changes happen. Then, click on the X mark to UNDO the changes and bring back the question mark. Now click on the question mark to bring the other choices back again. Now click on the right X mark and see what changes happen. Again, click on the X mark to remove it. Since we have decided that this value is wrong, go ahead and click on any of the 2 X marks.

As you probably noticed when you placed the first X, things have changed! Why don't you take a few seconds to explore the things that have changed by moving your mouse over the items and viewing the popup messages?

Now let's make a decision about TotalCost's value. For the current set of inputs, TotalCost should be 1600. But our TotalCost cell says 2800. That

means the value associated with the TotalCost cell is “Wrong”. Click on the question mark in the decision box to place an X-mark. Take a few seconds to explore anything that might have changed by moving your mouse over the items and viewing the popup messages.

Finally, I notice that, according to the example spreadsheet in the description, PaperTotalCost should be 1600. But our value is 2800, and that is wrong. Place an X-mark on this cell as well.

There is at least one bug in a formula somewhere that is causing these three cells to have incorrect values. I’m going to start looking for this bug by examining the PaperTotalCost cell. Let’s open PaperTotalCost’s formula. PaperTotalCost is taking the value of the Paper cell and multiplying it by 7. Let me go back and read my spreadsheet description. I’m going to read from the “Costs of Pen and Paper” section. *(read the section)* So the cost of paper is four dollars, but this cell is using a cost of seven. This is wrong. So change the 7 in this formula to a 4, and click the Apply button to finalize your changes.

Hey wait, the total spreadsheet testedness at the top of the window went down to 0%! What happened? Well, since we corrected the formula, Forms/3 had to discard some of our previous testing. After all, those tests were for the old formula. We have a new formula in this cell, so those tests are no longer valid. But, never fear, we can still retest these cells.

For example, the value of this PaperTotalCost cell is 1600, which matches the example spreadsheet in my description. Since this cell is correct, let’s click to place a checkmark in the decision box for PaperTotalCost. Oh good, the percent testedness of my spreadsheet went up to 7%; We got some of my testedness back.

Let's work on getting another cell fully tested. Look at the value of the PaperQCheck cell. Is this value correct? Let's read the second paragraph at the top of the spreadsheet description. *(read it)* With a value of 400 in the Paper cell, and a value of 21 in the PaperOnHand cell, we have 421 sheets of paper, which is enough to fill our shelves. Since the PaperQCheck cell says "paper quantity ok", its value is correct. Click in the decision box of this cell to place a checkmark.

But wait! The border of this cell is only purple. Rest your mouse over this cell border to see why. The popup message says that this cell is only 50 percent tested.

Let's middle-click on this cell to bring up the cell's arrows. Hey, the arrows are both purple too. Rest your mouse over the top arrow that is coming from the Paper cell. Ah ha, the relationship between Paper and PaperQCheck is only 50% tested! So there is some other situation we haven't tested yet. Change the value of the Paper cell to see if we can find this other situation. Click on the little button with an arrow on the bottom-right-hand side of the cell. Let's try changing the value to 380, and click the Apply button.

Now look at the decision box of the PaperQCheck cell. It is blank. I don't remember what that means, so rest your mouse over the decision box of this PaperQCheck cell. Oh yeah, it means we've already made a decision for a situation like this one. Okay, let's try another value for the Paper cell. I'm going to try a really small value. Move your mouse back to the formula box for the Paper cell, change its value to 10, and click the Apply button. Now push the Hide button on this formula box.

Now look at the PaperQCheck cell. There we go! The decision box for the cell now has a question mark, meaning that if we make a testing decision on this cell, we will make some progress. Let's look at the cell's value. Well, with 10 in the Paper cell and 21 in the PaperOnHand cell, we have 31 papers on stock. Is this enough paper? The spreadsheet description says we need 400 reams of paper, but we only have 31. So this is not enough paper. And the PaperQCheck cell says "not enough paper". Well, this is correct, so let's click on the PaperQCheck cell's decision box. Alright! The border changed to blue, and even more, the spreadsheet is now 35% tested.We don't need those arrows on PaperQCheck anymore, so hide them by middle-clicking on the PaperQCheck cell.

Why did it take two checkmarks to fully test the PaperQCheck cell? Let's open the cell's formula to find out (*open the formula*). See that this formula has an if-then-else statement. It says that **if** the sum of Paper and PaperOnHand is less than 400, **then** the cell should display "not enough paper". **Else or otherwise**, it should display "paper quantity ok". In other words, for PaperQCheck, if Paper plus PaperOnHand is less than 400, then "not enough paper" should appear in the cell, and if Paper plus PaperOnHand is greater than or equal to 400, "paper quantity ok" should appear in the cell. Push the Hide button on the formula box of the PaperQCheck cell.

Now let's look at the PenQCheck cell. This cell is displaying "pen quantity ok". Is this correct? Our spreadsheet description says you must keep more than 68 boxes of pens on hand. But we only have 25 boxes of pens on hand, because the Pens cell is 0 and the PensOnHand cell is 25. So even though we don't have enough pens, the PenQCheck cell is displaying "pen quantity ok". This value is not correct, so click on the question mark in PenQCheck's decision box to place an X-mark.

I'll give you a couple minutes to try to fix the bug that caused PenQCheck to have this wrong value. After a couple minutes, we'll fix the bug together to make sure that you've found it.
(*wait exactly two minutes*)

Okay, let's start by looking at PenQCheck's formula. Unless you have changed this cell's formula, it says that if the sum of the Pens and PensOnHand cells is greater than 68, then the cell should contain "not enough pens", and otherwise it should contain "pen quantity ok". But let's go back and look at our spreadsheet description and read that second paragraph again. It says that we only need to keep 68 or more boxes of pens in stock. So, based on the description PenQCheck should really print "pen quantity ok" if Pens plus PensOnHand is greater than 68, and otherwise it should print "not enough pens". So let's change this formula accordingly and push the "Apply" button when you are done. (*wait a second*). Note that PenQCheck now displays the correct value. So go ahead and put a checkmark in this cell by clicking on the question mark.

Look at the bottom of the description. It says, "Test the spreadsheet to see if it works correctly, and correct any errors you find." Remember, if you are curious about any aspect of the system, you can hover your mouse over the item and read the popup and also get more information / explanation like "What does this mean?", "How did it happen?", "How should I proceed?", and "Advice". Also, you might find those checkmarks and X-marks to be useful. Starting now, you'll have a few minutes to test and explore the rest of this spreadsheet, and to fix any bugs you find. Remember, your task is at the bottom of your spreadsheet description.

Gradebook.frm

Here is a Gradebook spreadsheet problem. Let's read the second paragraph at the top of the description:

"Your task is to test the updated spreadsheet to see if it works correctly and to correct any errors you find."

The front side of this description describes how the spreadsheet should work.

Also, if you turn to the backside of this sheet (*turn over your description*), you'll see that two correct sample report cards are provided to you. You can use these to help you in your task.

Remember, your task is to test the spreadsheet, and correct any bugs you find. To help you do this, use the checkmarks and X marks by clicking cell decision boxes.

Start your task now, and I'll tell you when time is up.

(*Task is 22 minutes*)

Payroll.frm

Here is a payroll spreadsheet problem. Let's read the second paragraph at the top of the description:

"Your task is to test the updated spreadsheet to see if it works correctly and to correct any errors you find."

The front side of this description describes how the spreadsheet should work.

Also, if you turn to the backside of this sheet (*turn over your description*), you'll see that two correct sample payroll stubs are provided to you. You can use these to help you in your task.

Remember, your task is to test the spreadsheet, and correct any bugs you find. To help you do this, use the checkmarks and X marks by clicking cell decision boxes.

Start your task now, and I'll tell you when time is up.

(*Task is 35 minutes*)

APPENDIX C

Purchase Budget

You are in charge of ordering office supplies for the office you work at. You must order enough pens and paper to have on hand, but you cannot spend more than your allotted budget for office supplies.

You must keep more than 68 boxes of pens and 400 reams of paper on hand and you cannot exceed a budget of $2000.

Pen and Paper
The quantity of pens and paper that you are ordering and the quantity you have on hand.

Costs of Pen and Paper
The cost of pens is $2 per box, and the cost of paper is twice that, $4.

Pen and Paper Check
These cells are used to check to ensure you are ordering enough pens and paper to restock the shelves.

Total Cost
The total cost is the combined cost of pens and paper. The BudgetOK cell determines if you went over your allotted budget.

Example data for correct spreadsheet

Pens	0
Paper	400
PensOnHand	25
PaperOnHand	21
PenTotalCost	0
PaperTotalCost	1600
PenQCheck	not enough pens
PaperQCheck	paper quantity ok
TotalCost	1600
BudgetOK?	Budget ok

Task: Test the spreadsheet to see if it works correctly and correct any errors you find.

Figure 2: Purchase Budget Task – Spreadsheet Description

GRADEBOOK SPREADSHEET PROBLEM

Another teacher has updated a spreadsheet program that computes the course grade of a student. Two correct sample report cards and information about the class' grading policy are provided.

Your task is to test the updated spreadsheet to see if it works correctly and to correct any errors you find.

Quizzes

There are five quizzes. The lower of the first two quiz scores is dropped. The average quiz score is then the average of the highest four quiz scores.

Midterm Exams

There are three midterms. The first midterm has 50 possible points; however, it must be adjusted to a "0-100" percentage scale. The third midterm score is curved; students receive a two-point bonus if their score is not zero.

The lower of the first two midterm scores is dropped. The average midterm score is then the average of the third midterm and the higher of the first two midterm scores.

Final Exam

There is one final exam. It has 146 possible points. It must be adjusted to a "0-100" percentage scale.

Exam Average

The exam average is the average of three scores: the two highest midterm scores and the final exam score.

Course Average

Quizzes are worth 40% of a student's grade. Midterms are worth 40% of a student's grade. The final exam is worth 20% of a student's grade.

Course Grade

A student's course grade is determined by their course average, in accordance with the following scale:

90 and up : A	70 - 79 : C
80 - 89 : B	60 - 69 : D
	Below 60 : F

Figure 3: Gradebook Task – Spreadsheet Description

	Example Correct Gradebook Report Cards
John Doe	**Report Card**
Quiz1	81.25
Quiz2	100
Quiz3	100
Quiz4	96
Quiz5	100
Quiz_Average	99
Midterm1 (Original)	45
Midterm2	96
Midterm3 (Original)	80
Midterm_Average	89
Final	129
Final_Percentage	88.36
Course_Avg	92.87
Course_Grade	A
Mary Smith	**Report Card**
Quiz1	0
Quiz2	88.24
Quiz3	85
Quiz4	87
Quiz5	100
Quiz_Average	90.06
Midterm1 (Original)	24
Midterm2	61
Midterm3 (Original)	66
Midterm_Average	64.5
Final	106
Final_Percentage	72.6
Course_Avg	76.34
Course_Grade	C

Figure 3 (Continued): Gradebook Task – Spreadsheet Description

PAYROLL SPREADSHEET PROBLEM

- **A spreadsheet program that computes the net pay of an employee has been updated by one of your co-workers.**
- **Below is a description about how to compute the answers.**
- **On the backside of this sheet are two correct examples, which you can compare with the values on screen.**

Your task is to test the updated spreadsheet to see if it works correctly and to correct any errors you find.

FEDERAL INCOME TAX WITHHOLDING

To determine the federal income tax withholding:

1. From the monthly adjusted gross pay subtract the allowance amount (number of allowances claimed multiplied by $250). Call this amount the adjusted wage.
2. Calculate the withholding tax on adjusted wage using the formulas below:
   a. If Single and adjusted wage is not greater than $119, the withholding tax is $0; otherwise the withholding amount is 10% of (adjusted wage – $119).
   b. If Married and adjusted wage is not greater than $248, the withholding tax is $0; otherwise the withholding amount is 10% of (adjusted wage – $248).

SOCIAL SECURITY AND MEDICARE

Social Security and Medicare is withheld at a combined rate of 7.65% of Gross Pay. The Social Security portion (6.20%) will be withheld on the first $87,000 of Gross Pay, but there is no cap on the 1.45% withheld for Medicare.

INSURANCE COSTS

The monthly health insurance premium is $480 for Married and $390 for Single. Monthly dental insurance premium is $39 for Married and $18 for Single. Life insurance premium rate is $5 per $10,000 of insurance. The monthly employer insurance contribution is $520 for Married and $300 for Single.

ADJUSTED GROSS PAY

Pretax deductions (such as child care and employee insurance expense above the employer's insurance contribution) are subtracted from Gross Pay to obtain Adjusted Gross Pay.

Figure 4: Payroll Task – Spreadsheet Description

	Example Correct Payroll Stubs		
John Doe		**Month**	**Year-To-Date**
Marital Status – Single			
Allowances	1		
Gross Pay		6,000.00	54,000.00
Pre-Tax Child Care		0.00	
Life Insurance Policy Amount		10,000	
Health Insurance Premium	390.00		
Dental Insurance Premium	18.00		
Life Insurance Premium	5.00		
Employee Insurance Cost	413.00		
Employer Insurance Contribution	300.00		
Net Insurance Cost		113.00	
Adjusted Gross Pay		5,887.00	
Federal Income Tax Withheld	551.80		
Social Security Tax	372.00		
Medicare Tax	87.00		
Total Employee Taxes		1,010.80	
Net Pay		4,876.20	
Mary Smith		**Month**	**Year-To-Date**
Marital Status – Married			
Allowances	5		
Gross Pay		8,000.00	72,000.00
Pre-Tax Child Care		400.00	
Life Insurance Policy Amount		50,000	
Health Insurance Premium	480.00		
Dental Insurance Premium	39.00		
Life Insurance Premium	25.00		
Employee Insurance Cost	544.00		
Employer Insurance Contribution	520.00		
Net Insurance Cost		24.00	
Adjusted Gross Pay		7,576.00	
Federal Income Tax Withheld	607.80		
Social Security Tax	496.00		
Medicare Tax	116.00		
Total Employee Taxes		1,219.80	
Net Pay		6,356.20	

Figure 4 (Continued): Payroll Task – Spreadsheet Description

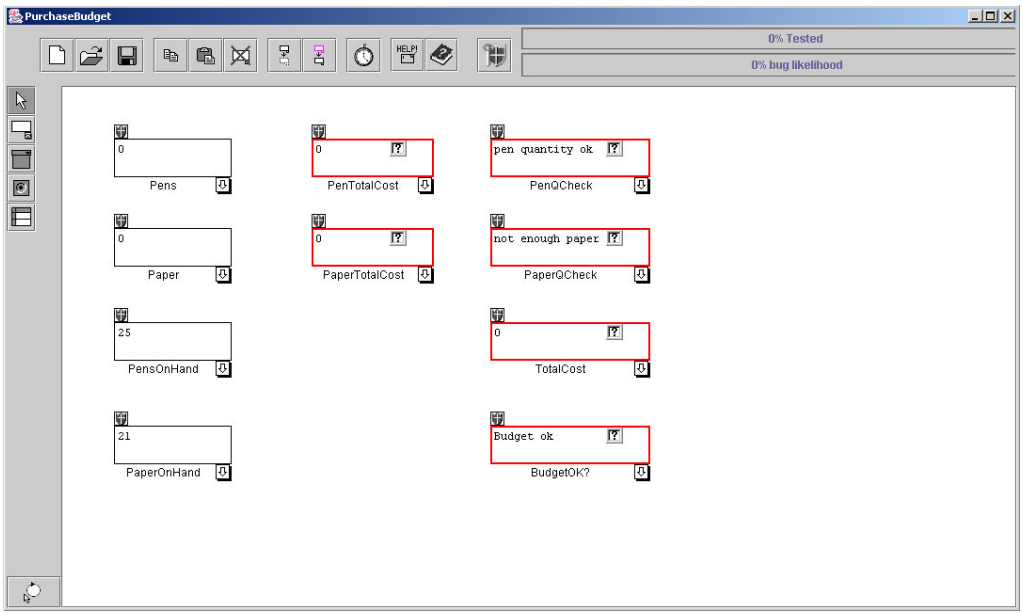

Figure 5: PurchaseBudget Spreadsheet (PuchaseBudget.frm)

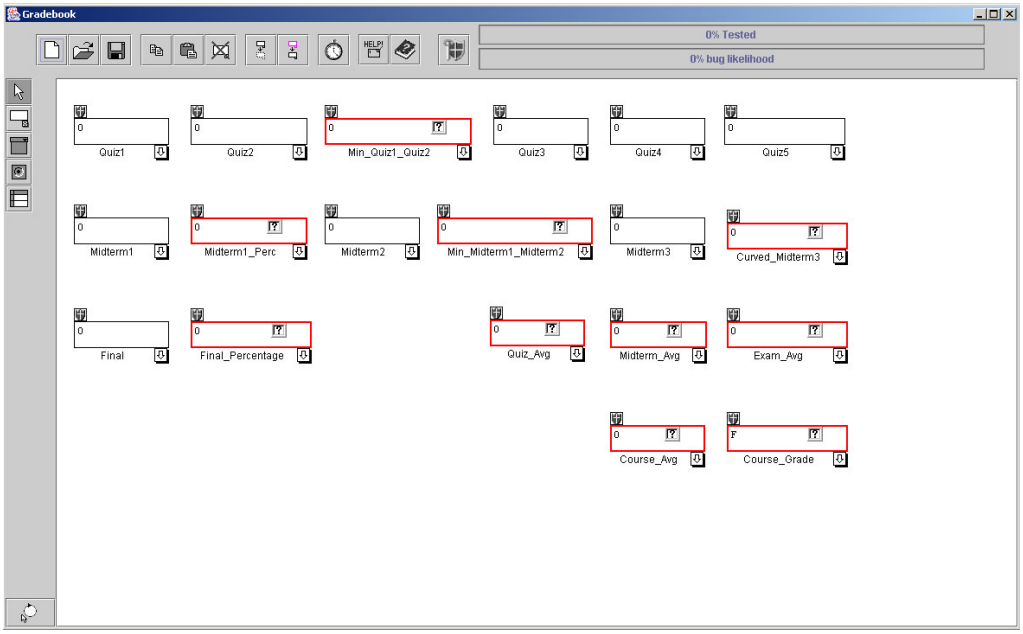

Figure 6: Gradebook Spreadsheet (Gradebook.frm)

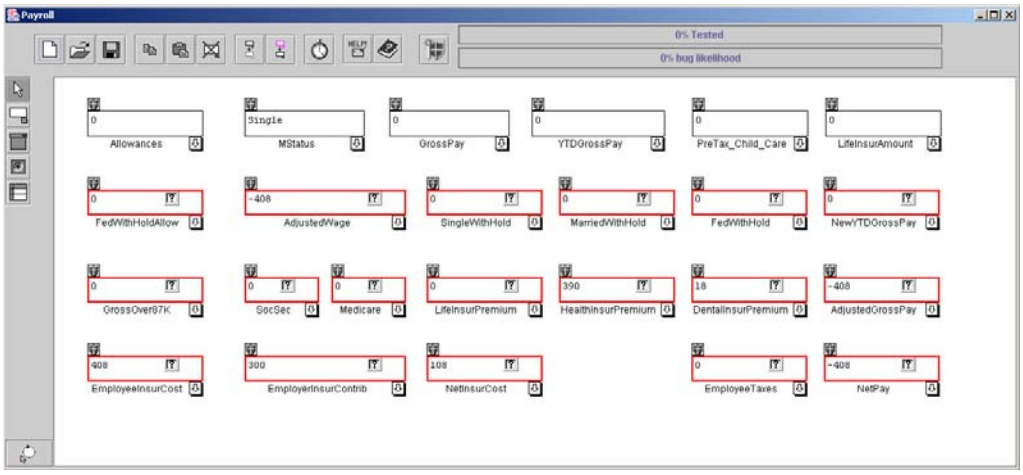

Figure 7: Payroll Spreadsheet (Payroll.frm)

APPENDIX D

Explanations Used for Think-aloud 2

A. Border Colors

Redborder:

TT: 0% of this cell has been tested

What: The red border means you have not tested this cell. Red means untested, blue means tested, and any color in between (i.e., purples) means partially tested. Testing helps you find errors.

How did: If this cell border was purple or blue before and is now red, this means the cell is no longer tested. This is either because you edited a related formula, or because you removed a √. Making a decision about this cell's value helps you find out if formulas have errors.

How should: If you can decide that the value in this cell appears to be correct given its input value(s), click on the "?" to check (√) it off. If you can decide that the value is wrong given those input value(s), click on the "?" to X it out. Checking it off will increase the testedness of this cell. Xing it out will cause the system to help guide you to the cause of the bad value.

Advice: You can use the border colors to systematically test your spreadsheet. If you can make a decision about a cell's value (correct or wrong) you can (1) test this cell given different inputs, or (2) move onto testing another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the system's guidance (colorings) to find the cause(s).

Border colors reflect the number of √s on this or related cells, and interior tinting on the cells reflects the number of Xs on this or related cells.

Blue Border

TT: 100% of this cell has been tested

What: The blue border indicates that you have tested this cell. Red means untested, blue means tested, and any color in between (i.e., purples) means partially tested. Testing helps you find errors.

How did: If this cell border was purple or red before and is now blue, it means that the cell was not completely tested before you √ed this cell or a cell that this cell affects.

How should: This cell border is blue, but you could try out more values on it, which could still reveal new errors. (More testing never hurts.) OR, you can proceed to other less tested cells (purple or red). Testing helps you find errors.

Advice: You can use the border colors to systematically test your spreadsheet. If you can make a decision about a cell's value (correct or wrong) you can (1) test this cell given different inputs, or (2) move onto testing another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the system's guidance (colorings) to find the cause(s).

Border colors reflect the number of √'s on this or related cells, and tints on the entire cell reflect the number of Xs on this or related cells.

Purple Border:

NOTE: Same for all

TT: X% of this cell has been successfully tested.

What: The purple border means that this cell has been partially tested, but that other situations still need to be tested. The "?" says you have a new opportunity to test. Trying more situations helps you find errors.

Advice: You can use the border colors to systematically test your spreadsheet. If you can make a decision about a cell's value (correct or wrong) you can (1) test this cell given different inputs, or (2) move onto testing another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the system's guidance (colorings) to find the cause(s).

Border colors reflect the number of √'s on this or related cells, and tints on the entire cell reflect the number of Xs on this or related cells.

CASE 1: Question Mark (?)

How did: The purple border and "?" mean you previously decided that this cell's value (or one that it affects) was correct, and checked it off. Since then, some values have changed, so this cell is in a new situation.

How should: If you can decide that the value in this cell appears to be correct given its input value(s), click on the "?" to check (√) it off. If you can decide that the value is wrong given its input value(s), click on the "?" to X it out. OR You can get into a new situation by changing some of the input values, by editing some of the values or by pushing the Help-Me-Test button (**draw it here**). **hide the last part of above sentence** Looking for new testing opportunities (marked by "?"s) helps you make progress testing. Testing helps you find errors.

CASE 2: Checkmark (√)

How did: The purple border and the √ mean you previously decided that this cell's value was correct, and checked it off.

How should: You can get into a new situation by changing some of the input values, by editing some of the values or by pushing the Help-Me-Test button (**draw it here**). **hide the last part of above sentence** Looking for new testing opportunities (marked by ?s) helps you make progress testing. Testing helps you find errors. Testing helps you find errors.

CASE 3: Blank checkbox ()

How did: The purple border means you previously decided that this cell's value (or one that it affects) was correct, and checked it off.

How should: Even though this situation has been tested you could try out more values on it, which could still reveal new errors. (More testing never hurts.) OR You can get into a new situation by changing some of the input values, by editing some of the values or by pushing the Help-Me-Test button (**draw it here**). **hide the last part of above sentence** Looking for new testing opportunities (marked by ?s) helps you make progress testing. Testing helps you find errors. Testing helps you find errors.

CASE 4: X-mark (X)

How did: The purple border means you previously decided that this cell's value (or one that it affects) was correct, and checked it off. You've also spotted a wrong value in this cell, and X'd it out.

How should: You can get into a new situation by changing some of the input values, by editing some of the values or by pushing the Help-Me-Test button (**draw it here**). **hide the last part of above sentence** Looking for new testing opportunities (marked by ?s) helps you make progress testing. Testing helps you find errors. Testing helps you find errors.

B. Decision Box

Question Mark (?):

TT: If you can decide that the cell's value is correct or wrong, click. These decisions help to test and find errors.

What: A "?" appears when you can make testing progress by making a decision about this cell's value. Each value is a "test case". Testing helps you find errors.

How did: A "?" is shown in the decision box whenever a situation is not tested. A "?" can reappear in a decision box for a few reasons: (1) a formula may have been changed somewhere in the spreadsheet which requires this cell be retested, or (2) some input values were changed and cover some previously untested situation.

How should: If you can decide that the value in this cell is correct given its inputs, check (√) it off. If you can decide that the value is wrong X it out. Checking it off will increase the testedness of this cell. Xing it out will cause the system to help guide you to the cause of the bad value.

Advice: You can use the decision boxes to record decisions about the cells' values. If you have made a decision about a cell's value and √ed it off or Xed it out, you can then: (1) move onto another cell to continue testing, (2) finish testing one cell completely then move onto another cell, or (3) if there are

tinted cells, which indicate possible locations of errors, follow the system's guidance (colorings) to find the cause(s).

Border colors reflect the number of √'s on this or related cells, and tints on the entire cell reflect the number of Xs on this or related cells.

A √ is a way to say the value is correct, an X is a way to say a value is wrong. A "?" shows that there is an opportunity to make this cell more tested if you can decide about its current value. (If the decision box is empty, you are still allowed to √ it off or X it out.)

Checkmark (√):

TT: You have decided this cell's value is correct.

What: A √ appears when you decide this cell's value is correct. Each value is a "test case". Testing helps you find errors.

How did: You √ed this cell's decision box. Usually this causes the border color to become more blue than it was before, reflecting the fact that the cell is more tested than it was before.

How should: If you think you shouldn't have √ed off this value, you can click the √ again to remove it. If you want to make more decisions about this cell, you can change some input values (with or without the help of the Help-Me-Test button **hide) that affect this cell and make new decisions about the results.

X-mark (X):

What: An X appears when you decide this cell's value is wrong. Each value is a "test case". Testing helps you find errors.

How did: You Xed this cell's decision box. Usually this causes the interior color to become more orange than it was before, reflecting the fact that the cell has higher bug likelihood.

How should: If you think you shouldn't have Xed off this value, you can click the X again to remove it. If you want to make more decisions about this cell, you can change some input values (with or without the help of the Help-Me-

Test button **hide) that affect this cell and make new decisions about the results.

No mark (blank decision box):

What: The decision box is blank when this cell's value does not cover a new situation. Each value is a "test case". Testing helps you find errors.

How did: You either √ed a cell affected by this cell, or if this cell's decision box previously had a √ then inputs values changed, but these new values are not a new situation.

How should: If you can decide that the value in this cell is correct given its inputs, check (√) it off. If you can decide that the value is wrong X it out. Xing it out will cause the system to help guide you to the cause of the bad value. Testing helps you find errors.

C. Arrows

TT: Relationship between X and Y is Z% tested.

What: Arrows show (1) that one cell contributes to another cell's value, and (2) the color show how much of this relationship has been tested.

How did: Arrows were turned on by middle clicking on a cell. You can turn the arrows off by (1) middle clicking on the same cell or (2) middle clicking on the arrow.

How should: If you are using arrows to aid in testing and to find new situations open the formulas to see more specifically which input cells need to change to find new situations (pay attention to red and purple arrows for situations that need testing).

Advice: Arrows can be used to help you find situations that have not been tested. If an arrow between two cells is purple or red you can open up the formula(s) to see which part of the situation has not yet been tested. Then change input values or use HMT to help you generate new inputs.

Arrows can also help you with a big picture feel for the relationships among the spreadsheet cells.

D. Interior cell colors

TT: BUG LIKELIHOOD: VERY LOW / LOW / MEDIUM / HIGH / VERY HIGH

NOTE: This applies to all cases:

Advice: The tinted cell(s) are likely to have bug(s): the darker the color the more likely there is to be a bug in that cell's formula. The more decisions you are able to make about values (correct or wrong) the more accurate the feedback can be.

CASE 1: Question Mark (?)

What: An orange or yellow cell interior means there might be a bug in this cell's formula.

How did: Although you have not explicitly made a decision about this cell's value, another cell affected by this cell was X'ed out. Since this cell affects the cell with the wrong value it is possible the problem is with this cell's formula.

How should: Check for formula bugs, OR look at other tinted cells, OR √ off and X out other cells' values to get more feedback. If you can decide the correctness of this cell's value √ it off (if the value is correct) or X it out (if it's wrong) – this will help you narrow your search for the bug. Using the interior color feedback you may be able to locate the bug.

CASE 2: Check Mark (√)

What: An orange or yellow cell interior means there might be a bug in this cell's formula

How did: Although you have decided that this cell's value is correct, this cell affects a cell with a wrong value.

How should: Check for formula bugs in this cell, OR look at other tinted cells, OR √ off and X out other cells' values to get more feedback. There is a bug somewhere (either in this cell's formula or in the cells' formulas affecting this

cell). Look at these cells' formulas, or √ off or X out other cells' values to narrow your search.

CASE 3: X mark (X)

What: An orange or yellow cell interior means there might be a bug in this cell's formula.

How did: You decided that the cell's value is wrong. Previous testing decisions on this cell or other cells affect by this cell also impact the color of this cell.

How should: Check for formula bugs in this cell, OR look at other tinted cells, OR √ off and X out other cells' values to get more feedback. There is a bug somewhere (either in this cell's formula or in the cells' formulas affecting this cell). Look at these cells' formulas, or √ off or X out other cells' values to narrow your search.

E. Progress Bars

FL progress bar:

TT:

What: Your testing has narrowed down the possible sources of the bugs to most likely be in one or more of the darkest-tinted cells.

How did: This progress bar changes to reflect the current state of all the tinted cells. It changes when a testing decision is made (√ or X). The progress bar can also change when formulas are edited.

How should: You can look to the darkest cell's formula to start searching for the bug. OR To receive more feedback from the system and narrow your search for the bug, make decisions about other cell's values. This may make some cells' interiors darker indicating the likelihood of the bug being in those cells. Using the interior color feedback you may be able to locate the error.

Advice: The tinted cell(s) are likely to have error(s): the darker the color the more likely there is to be an error in that cell's formula. The more decisions

you are able to make about values (correct or wrong) the more accurate the feedback can be.

Testedness progress bar:

TT:

What: The overall testing progress of this spreadsheet. Testing helps you find errors.

How did: Each time a value is √ed off for a new situation a cell becomes more tested, that change is reflected in this overall spreadsheet testedness. Spreadsheet testedness can decrease when formulas are edited or √ is removed.

How should: If you can decide that a value in some cell appears to be correct given its inputs, click on the "?" to check (√) it off. If you can decide that the value is wrong given its inputs, click on the "?" to X it out. Checking it off will increase the testedness of the cell. Xing it out will cause the system to help guide you to the cause of the bad value.

Advice: You can use the border colors to systematically test your spreadsheet. If you can make a decision about a cell's value (correct or wrong) you can (1) test this cell given different inputs, or (2) move onto testing another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the system's guidance (colorings) to find the cause(s).

Border colors reflect the number of √'s on this or related cells, and tints on the entire cell reflect the number of Xs on this or related cells.

F. Confidence Marks

NOTE: For all cases the following are the same:

How did: You clicked on the decision box.

Advice: You can use the decision boxes to record decisions about the cells' values. Once you make a decision about a cell's value (correct or incorrect)

you can either: (1) move onto another cell to continue testing, (2) finish testing one cell completely then move onto another cell, or (3) if there are tinted cells, which indicate possible locations of errors, follow the feedback to help you locate the error.

A √ is a way to say the value is correct, an X is away to say a value is wrong. A "?" shows that there is an opportunity to make this cell more tested if you can decide about its current value. (If the decision box is empty, you are still allowed to √ it off or X it out.)

High Confidence Checkmark (√):

TT: it's right (√)

What: √ing this cell means that this cell's value is correct given its inputs. Testing helps you find errors.

How should: If you can decide that this cell's value is correct given its inputs, √ it off. If you can decide that the value is wrong given its inputs, X it out. Checking it off will increase the testedness of this cell. Xing it out will cause the system to help guide you to the cause of the bad value.

Low Confidence Checkmark (√):

TT: it's right maybe (√)

What: √ing this cell means this cell's value might be correct given its inputs. Testing helps you find errors.

How should: If you are not entirely sure but there are indications that this cell's value is correct given its inputs, √ it off. If you can decide that the value is wrong given those input value(s), X it out. Checking it off will increase the testedness of this cell. Xing it out will cause the system to help guide you to the cause of the bad value.

High Confidence X-mark (X):

TT: it's wrong (X)

What: Xing this cell means that this cell's value is wrong given its inputs. Testing helps you find errors.

How should: If you can decide that this cell's value is wrong given its inputs, X it out. Xing it out will cause the system to help guide you to the cause of the bad value.

Low Confidence X-mark (X):

TT: it's wrong maybe (X)

What: Xing this cell means this cell's value might be wrong given its inputs. Testing helps you find errors.

How should: If you are not entirely sure but there are indications that this cell's value is wrong given its inputs, X it out. Xing it out will cause the system to help guide you to the cause of the bad value.