AN ABSTRACT IN THE DISSERATION OF

Laura A. Beckwith for the degree of Doctor of Philosophy in Computer Science
presented on April 23, 2007.
Title: Gender HCI Issues in End-User Programming

Abstract Approved: ____
Margaret M. Burnett

Until recently, research has not considered whether the design of end-user programming environments, such as spreadsheets, multimedia authoring languages, and CAD systems, affects males and females differently. As a result, we began investigating how the two genders are impacted by end-user programming software and whether attention to gender differences is important in the design of software. Evidence from other domains, such as psychology and marketing, strongly suggests that females process information and problem solve in very different ways than males. This implies that without taking these differences into account in the design of problem-solving software, the needs of half the population for whom the software is intended are potentially being ignored. In fact, some research has shown that software is unintentionally designed for males. Our research has uncovered several factors which affect males and females differently as they engage in end-user programming. The gender differences range from the effects of self-efficacy (a form of confidence) on engagement with environment features to how males and females use "tinkering" as part of their problem-solving strategy. We further investigate the effects of several environment changes on both males' and females' problem solving. This research is the first to both uncover what gender differences are relevant in end-user programming environments and address how to account for these gender differences in the design of such environments.

Gender HCI Issues in End-User Programming

by

Laura A. Beckwith

A DISSERTATION
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented April 23, 2007
Commencement June 2007

Doctor of Philosophy dissertation of Laura A. Beckwith presented on April 23, 2007.

APPROVED:

Major Professor, representing Computer Science

Associate Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

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

Laura A. Beckwith, Author

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support and guidance of my advisor, Margaret Burnett, who continuously showed me that my limits were expandable.

I have received invaluable feedback and support from several researchers outside of Oregon State University including: Susan Wiedenbeck, Alan Blackwell, and Mary Czerwinski. At OSU I have worked closely with members of the Gender HCI, Forms/3, and EUSES teams, many of whom have had a direct impact on the work presented in this dissertation, including Curtis Cook, Shraddha Sorte, Valentina Grigoreanu, Vaishnavi Narayanan, Kyle Rector, Derek Inman, Neeraja Subrahmaniyan, Cory Kissinger, Joseph Lawrance, Thippaya Chintakovid, and Robin Abraham for words of wisdom.

This work was supported by NSF CNS-0420533, by the EUSES Consortium via NSF ITR-0325273, and by Microsoft Research.

Finally, I owe many thanks to the friends and family that have been involved in my PhD work. In particular I would like to thank my mother, father, and Melissa for being there at a moments notice, always with encouraging words. My husband, Ulf, picked up his own life in Denmark to move here so we could be together; his unending support and faith in me was contagious!

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF APPENDIX FIGURES

Gender HCI Issues in End-User Programming

1. Introduction

In high school, “Ashley’s” career plan was to become a graphic designer. However, when attempting to tackle Flash programming in the graphics design course, frustration set in. Although Flash is used by graphic designers, the original WYSIWYG programming style has largely given way to a Java-like language aimed at software developers who use Flash. Then the need to do web programming arose as well. Ashley decided that learning Flash and web programming was too great a barrier, and instead majored in art.

What were the real causes of Ashley’s difficulties? Possibly Ashley’s problem-solving style, learning style, or level of confidence made learning these software tools and end-user programming environments seem more formidable than it would to someone else. Gender differences in these and other domains, such as psychology, marketing, and neuroscience, strongly suggest that females process information and problem solve in very different ways than males (c.f. [Bandura 1977, Meyers-Levy 1989]).

Ashley is an example of an end-user programmer. End-user programmers are people who program as a means to an end, and do not aspire to become professional programmers. Often they program because it is the fastest way to get a solution. Common end-user programming environments include spreadsheets, multimedia authoring languages, and CAD systems.

Despite substantial human-centric research relevant to end-user programming (e.g., [Blackwell 2002, Green and Petre 1996, Ko et al. 2006, Nardi 1993, Pane et al. 2001]), few researchers have considered potential *gender HCI issues*: gender differences that may need to be accounted for when designing end-user programming environments. Although, in the area of hardware design, there is a notable exception:

Czerwinski's pioneering research on the support of both genders in navigating through 3-D environments [Czerwinski et al. 2002, Tan et al. 2003].

Even though individual differences, such as in learning styles or spatial abilities, are known to have greater effects on an individual's performance than any group-based influences, such as gender or ethnicity, studying group-based differences, such as gender differences, has revealed useful solutions (e.g., [Czerwinski et al. 2002, Margolis et al. 1999, Margolis et al. 2003, Tan et al. 2003]).

One reason it is important to consider gender HCI issues in end-user programming is simply that ignorance of these issues is risky. Ignorance of gender issues has already proven to be dangerous: today's low percentage of computer science females [Camp 1997] has been directly attributed to the past unawareness of gender issues in computer science education and in the workforce. There is a risk that if gender HCI issues of end-user programming environments are ignored, a similar phenomenon could occur with end-user programmers.

This dissertation provides an initial foundation and several experiments to address the following open question: *Is gender an important factor in end-user programming environments?*

1.1 What Could Go Wrong?

What gender differences might matter in the design of end-user programming environments? Consider the following scenario in one particular end-user programming environment. In this scenario the user is engaged in an *end-user software engineering* task. (End-user software engineering tasks encompass all parts of the software engineering life cycle, from creation to testing to modification, etc.).

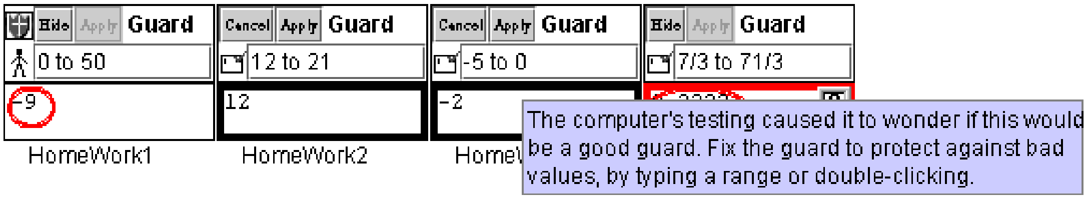
Imagine a female teacher engaged in preparing a spreadsheet to track her students' scores and to calculate their final grades. Part of her process of preparing her spreadsheet is to test the spreadsheet (to ensure her formulas work as she expects).

While she is engaged in testing, the system surprises her by decorating some of the spreadsheet cells with "assertions." (See Figure 1.)

The surprises were intentionally placed into the software by the designers relying on a strategy for end-user programming environments called *Surprise-Explain-Reward* [Wilson et al. 2003]. The surprise, which was intended to capture the teacher's attention and arouse her curiosity, reveals the presence of an "information gap" [Lowenstein 1994]. In this case the system is using the surprise to interest her in assertions [Burnett et al. 2003], which she can use to guard against future errors by specifying, for example, that the value of a cell calculating a grade average should always fall between 0 and 100.

What could go wrong in surprising the user? According to Lowenstein's information gap theory, a user needs to have a certain level of confidence in order to reach a useful level of curiosity [Lowenstein 1994]. However, given documented gender differences in computer confidence, the teacher's level of computer confidence could interfere with the surprise's ability to capture her interest.

Returning to our scenario, suppose for this particular user, the surprise is effective at arousing her curiosity; she looks to the object that surprised her (the assertion) for an explanation. The explanation, viewed through a tooltip, includes the semantics,



The computer's testing caused it to wonder if this would be a good guard. Fix the guard to protect against bad values, by typing a range or double-clicking.

Figure 1. Spreadsheet with Assertions: A spreadsheet calculating the average of three homework scores. Assertions about the ranges and values are shown above each cells' value. For example, on HomeWork1 there is a user-entered assertion (noted by the stick figure) of 0 to 50. The other three cells have assertions "guessed" by the Surprise-Explain-Reward strategy. Since the value in HomeWork1 is outside of the range of the assertion, a red circle notifies the user of the violation. A "tool tip" (lower right) shows the explanation for one of the guessed assertions.

possible actions she can take (regarding the assertion), and the future reward(s) of taking the action. (See Figure 1.)

What could go wrong with the explanation? According to one theory, males and females process information differently [Meyers-Levy 1989], and thus both the presentation and the content of the explanation may impact its effectiveness for males versus females. If the information needed by the user is not effectively communicated, the user's ability to problem solve is likely to be reduced.

Another role of the explanation is to help users make a reasonably accurate assessment of the risk in taking some action – but since males and females differ in their perceptions of risk, the explanation may need to serve these two populations differently in this respect as well. (An example of risk may be the fear that a user will lose his/her work if he/she tries a certain feature.) If one gender perceives an explanation of a feature as communicating higher levels of risk than another, the users with higher risk perceptions may avoid supposedly "risky" features that may be important to overall effectiveness.

Perhaps the most important role of explanations is to make clear the rewards of using particular end-user programming features. Providing information about rewards in the explanation is consistent with the implications of the Model of Attention Investment [Blackwell 2002], an analytic model of user problem-solving behavior that models the costs, benefits, and risks users weigh in deciding how to complete a task. An implication of this model is that if the system provides the user an idea of future benefits, users can better assess if the cost of using a feature (here assertions) is worth their time. The reward aspect of the strategy refers to rewards such as the automatic detection of errors, which is depicted by the red circle around HomeWork1's erroneous value in Figure 1.

What could go wrong with rewards? Since males and females are often motivated by different factors, there may be gender differences in what actually is perceived as a "reward." If the rewards are tailored to only one gender's perceptions of rewards, the other gender may not be motivated to use the devices that will help them be effective.

In this end-user programming scenario, potential problems arose *that may be addressable within the end-user programming software itself*. Four issues that arose here and potentially interact with gender differences were (1) software features whose effects on the user depend upon users' computer confidence, (2) the software's ability to communicate effectively with users, (3) the possibility of a user's perception of risk interfering with the user choosing to use appropriate features, and (4) possible differences between a user's actual motivations and the software's attempt to "reward" the user for using particular features.

1.2 Methodology

Our method for addressing the research question of whether gender differences exist in relation to end-user programming environments and how the design of these environments can better support such gender differences is as follows (summarized in Figure 2).

Using (1) theories from various fields relating to both end-user programming and gender differences, (2) we derived hypotheses with relation to gender differences to test in relation to end-user programming. Relying on these hypotheses and (3) qualitative evidence, such as data from think-aloud studies, or re-analysis of data from previous quantitative studies, (4) we derived specific research questions. (5) To test research questions, we designed and carried out quantitative empirical studies. (6) Using the results of the quantitative investigation along with the originating theories, we made changes to the prototype (the environment used in the quantitative study). (7) As we engaged in making changes to the prototype we simultaneously used qualitative methods to inform the design. (8) We followed up on the prototype changes using quantitative methods.

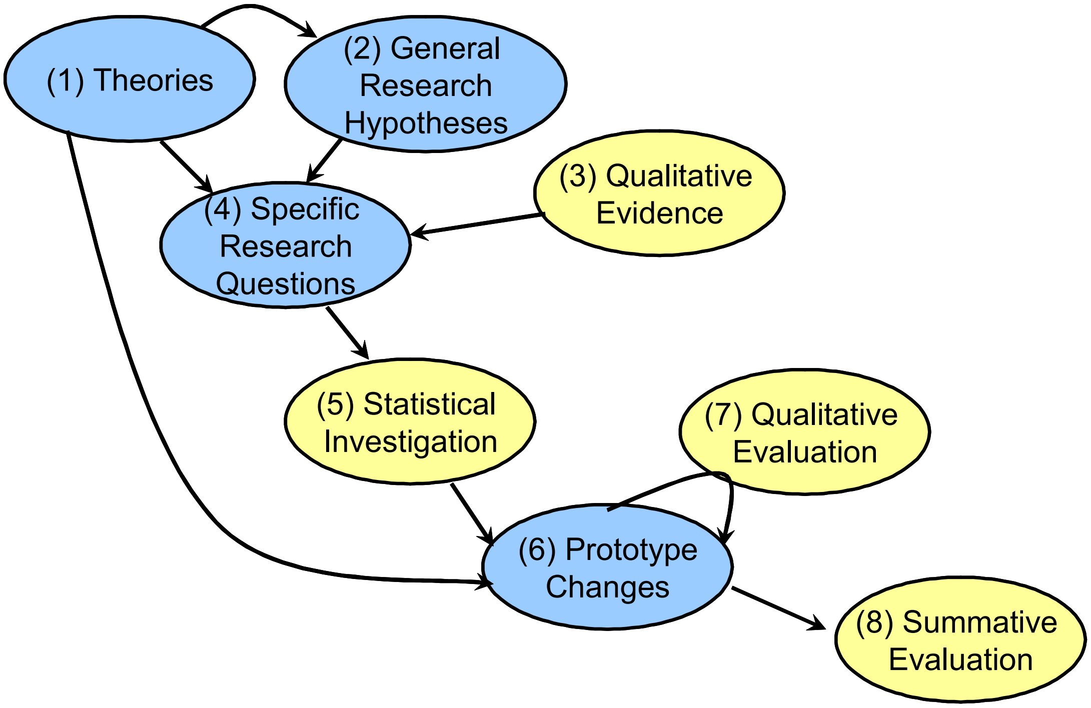

(1) Theories
(2) General Research Hypotheses
(3) Qualitative Evidence
(4) Specific Research Questions
(5) Statistical Investigation
(6) Prototype Changes
(7) Qualitative Evaluation
(8) Summative Evaluation

Figure 2. Methodology: The general steps for investigating gender differences in the design of programming environments. The empirical evaluations, whether qualitative or quantitative in nature, are differentiated from the other steps through their yellow color.

2. Hypotheses' Foundations[1]

Prior to the start of this research there was little research regarding gender and software design. The bulk of the research on gender relating to computer science concerned recruiting and retaining females in computer science and the information technology workforce. It expressly ignored the interactions of gender and software.

We began by assembling a body of relevant literature. It includes research about "computer science females," but primarily the relevant literature comes from domains outside of computer science. Using this literature we derived hypotheses applying these documented gender differences to end-user problem-solving software.

The focus of the hypotheses presented in this chapter is on *end-user problem solving*, rather than the more specific area of end-user programming, since a majority of the theories and research we draw from are not specific to end-user programming.

This chapter describes five theories: self-efficacy, selectivity hypothesis, attention investment, technology acceptance, and the information gap perspective on curiosity. It also touches on other literature with roots in gender differences or end-user problem-solving (with ties to gender). It then presents hypotheses testable in end-user problem-solving environments.

2.1 Self-Efficacy

Bandura's self-efficacy theory [Bandura 1977, Bandura 1986] defines self-efficacy as a person's belief in his/her ability to do a specific task. Presuming ability to complete a task, self-efficacy distinguishes how individuals will approach and perform the task. In particular, the impact of self-efficacy on performance occurs as a task becomes challenging; self-efficacy predicts reaction and behavior in challenging situations [Bandura 1986]. People with high self-efficacy are more likely to put in more effort, persist through challenging tasks for longer, and take more genuine interest in the task.

[1] The contents of this chapter are based on [Beckwith and Burnett 2004].

Not only does self-efficacy predict task behavior, but it ultimately affects performance outcomes; influencing whether or not an individual succeeds at the task.

Individuals with high self-efficacy for a specific task have several characteristics that aid their success in these tasks, characteristics self-doubters lack [Bandura 1986]. One of these characteristics is generating and testing alternative task strategies when their current strategy is failing. Another characteristic is abandoning a faulty strategy (which generally requires determining an alternative action). Self-doubters are less likely to abandon faulty strategies, and less likely to test alternative strategies in their effort to accomplish a task. Partially due to these characteristics, self-doubters are less likely to be successful in their tasks, which further increases their self-doubt that they are able to successfully accomplish the task and, in turn, reinforces their low self-efficacy beliefs.

Further aspects of self-efficacy theory will be discussed within the context of computer-related self-efficacy and gender.

2.1.1 Gender & Computer Self-Efficacy

Computer self-efficacy is defined (generally) as a person's judgment of his/her capabilities to use computers in a variety of situations [Compeau and Higgens 1995].

Low computer self-efficacy among females is a prevalent research result in the literature. This includes gender differences in self-efficacy among U.S. males and females, as well as males and females in many other countries, among both computer science majors and general computer users [Beyer et al. 2003, Busch 1995, Busch 1996, Colley and Comber 2003, Corston and Colman 1996, Durndell and Haag 2002, Durndell et al. 2000, Fallows 2005, Hargittai and Shafer 2006, Margolis and Fisher 2003, McCoy et al. 2001, McIlroy et al. 2001 Shashaani and Khalili 2001, Teasdale and Lupart 2001, Zeldin & Pajares 2000]. Although gender differences in self-efficacy are common, there are also studies in which no gender differences in computer self-efficacy were found (e.g. [Brosnan and Lee 1998, Miller and Crouch 2001, Rowell et al. 2003]).

Given the low female enrollment in computer science [Camp 1997], one assumption might be that those females who do choose to major in computer science must be especially confident. In fact, this is quite consistently not the case, as Margolis and Fisher highlight in numerous ways [Margolis and Fisher 2003]; even the females majoring in computer science at one of the top computer science programs within the USA suffer from low confidence:

> "I'm actually kind of discouraged now. Like I said before, [there are] so many people who know so much more than me, and they're not even in computer science. Like I was talking to this one kid, and …oh my God! He knew more than I do. It was so… humiliating kind of, you know? So I get discouraged by things like that—I don't know what I think I need to know. And that inhibits my willingness to continue (laughs) … if you can understand that. It shouldn't. It should like make me want to learn even more. But I feel like I'll always be behind, and it's discouraging." [Margolis and Fisher 2003]

In a study conducted by Beyer et al., which measured the self-efficacy of both computer science majors and non-majors, they found that even after accounting for quantitative ability, female computer science majors had lower computer confidence than the males *not* majoring in computer science [Beyer et al. 2003].

However, the population we are interested in is not the aspiring computer scientists, but rather, the end users. Despite researchers studying end users and measuring their self-efficacy through surveys, most stop without tying self-efficacy to computer behavior.

There are, however, a few notable exceptions. In one study of internet usage, experimenters had participants search the internet for information pertaining to several specific topics. Women perceived their ability in this task to be lower than the males did, but in fact in actual performance there was no difference [Hargittai and Shafer 2006].

Two other studies found similar results regarding use of technology, both looking at self-reported use and attitudes. In one study researchers surveyed students on a "laptop required" campus and found no difference in use (except in the area of entertainment, in which males used the computer more often). However, despite virtually no difference in actual use, the females reported that they considered themselves far less expert than the males [McCoy et al. 2001]. Likewise, Spotts et al. [1997] surveyed university professors regarding classroom technology use, and found no gender differences in use of technology in the classroom. Again, gender differences in perceptions of ability and of reported experience/knowledge of computer-related technologies were significantly higher by the males.

The research reviewed thus far suggests that, despite females having lower computer self-efficacy, their performance is unaffected. However, this does not therefore imply that low self-efficacy is not problematic: self-efficacy theory suggests that self-efficacy matters most in times of challenge and for making future choices of whether or not to engage in a particular (computer-based) activity. Further, the theory also suggests that negative experiences for someone with low self-efficacy have a larger impact than for someone with high self-efficacy. (This will be covered in more detail in the next section.)

Measures of self-efficacy closely related to end-user computing are relatively uncommon, and most focus on general computer use rather than computer-based problem-solving activities. However, one study in particular points out gender differences in self-efficacy among business students after having had a year long course in business-relevant computer applications. Busch found that females had significantly lower self-efficacy when asked questions about complex tasks, such as complex spreadsheet tasks [Busch 1995], although there were no gender differences on other more basic tasks within the same applications.

Within our own research on end users engaged in computer-based problem-solving, we are most interested in the ties between males' and females' self-efficacy and how this affects their use of features within problem-solving environments.

The information in this section leads to the following hypotheses:

Hypothesis SE-1: Gender differences in computer self-efficacy will be evident in differences in attitudes toward and engagement with unfamiliar features within problem-solving environments.

2.1.2 Sources of Self-Efficacy

According to self-efficacy theory, there are four major self-efficacy sources: performance accomplishments, vicarious experience, verbal persuasion, and emotional arousal. Table 2-1 provides a brief description of each of these sources. The following sections focus on performance accomplishments and emotional arousal

Table 2-1. Self-Efficacy Sources: The four major sources of self-efficacy [Bandura 1977, Zeldin and Pajares 2000].

Source	**Description**
Performance Accomplishment	Based on personal mastery experiences performance accomplishments are interpreted results of one's past performance(s). High self-efficacy gained through performance accomplishments is unlikely to be affected by occasional failure and will positively generalize to other related situations.
Vicarious Experience	Based on observing someone else (successfully) perform the task, in particular if the observed person is similar to the observer. Vicarious experience has the greatest impact when the observer has little prior experience on which to base self-efficacy judgment. These self-efficacy increases are greater when individuals see more than one person succeed, and the outcomes are clearly successful.
Verbal Persuasion	Verbal and social encouragement helps an individual exert extra effort and maintain persistence. However, any disconfirming evidence plays a much stronger part in determining self-efficacy once the verbal persuasion has been suggested. Furthermore, negative verbal messages can have a negative effect on low self-efficacy individuals.
Emotional Arousal	Emotional arousal, including stress, tension, mood, etc., will impact beliefs about personal competency, and susceptibility to failure.

because of their expected ties to gender differences.

Understanding specific sources of self-efficacy is important for addressing the impacts of self-efficacy. For example, understanding the effects of performance accomplishments, and factors related to performance accomplishments, may lead researchers to suggest different design choices for problem-solving software than considering vicarious experience.

2.1.2.1 Performance Accomplishments

Performance accomplishments [Bandura 1986] are typically the strongest determining factor in an individual's assessment of self-efficacy. If previous experiences were successful self-efficacy will generally be high; if previous experiences were unsuccessful self-efficacy will be low. However, this simplified explanation misses several important factors, as is illustrated in Figure 3. Mediating factors of external task support and effort expended impact any self-efficacy change. Failure at a task has little impact on self-efficacy if the person had high self-efficacy to begin with. (Recall that self-efficacy is a task specific construct, so high self-efficacy for a specific task is not impacted by an occasional failure.) A successful task performance increases self-efficacy only if the task was not perceived as easy, no external aids were necessary (or used), and an individual's effort was also not perceived as high.

The following example illustrates how these factors can interact: If a person with low self-efficacy scored high on a test, and did not perceive high exertion, they could attribute this to their own capabilities, thereby increasing their self-efficacy. Conversely, if they attribute their success to the instructor's study guide, rather than to their own abilities, they may credit their success to the external factor of the study guide, therefore, the impact on self-efficacy is inconsequential.

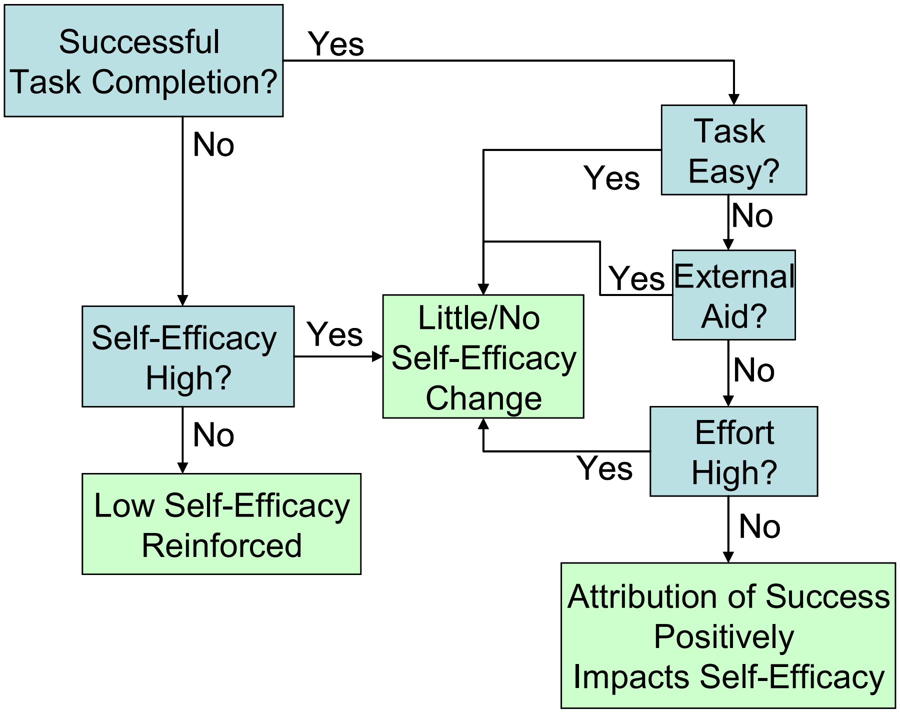

Successful Task Completion?
Yes
No
Task Easy?
Yes
No
Yes
External Aid?
Self-Efficacy High?
Yes
Little/No Self-Efficacy Change
No
Effort High?
Yes
No
Low Self-Efficacy Reinforced
No
Attribution of Success Positively Impacts Self-Efficacy

Figure 3. Self-Efficacy and Performance Accomplishments: Our representation of the self-efficacy literature related to performance accomplishments [Bandura 1977, Bandura 1986], representing the factors that affect self-efficacy due to performance on a task. Changes in self-efficacy are based upon several factors including difficulty of task, external aids, incoming self-efficacy, and effort expended in the task.

2.1.2.2 Performance Accomplishments & Gender Differences

Stereotypically, and borne out in research, females are less likely to attribute their success on a task to their own abilities [Beyer and Bowden 1997, Vermeer et al. 2000]; rather they attribute success to external or unstable factors, such as luck [Beyer and Bowden 1997]. Unlike task success, females are more likely than males to attribute their task failure to their lack of capability [Stipek and Gralinski 1991].

Females' tendency to attribute success to external factors can be problematic for low self-efficacy females, since according to self-efficacy theory, placing attributions for success on external aids (such as luck) is unlikely to increase self-efficacy. To compound this effect, Beyer and Bowden found that gender differences in self-perceptions of ability are especially apparent in “masculine” tasks [Beyer and Bowden 1997], of which computing tasks are often considered an example (as discussed in [Brosnan 1998]).

Factors of attributing success to external factors may be relevant for end-user programming environments. Numerous design choices have been made in these environments specifically to aid users in accomplishing a specific task (e.g., [Abraham and Erwig 2007, Burnett et al. 2004, Erwig et al. 2006, Ko and Myers 2004, Ko and Myers 2006, Pane et al. 2002, Ruthruff et al. 2004]). Users’ perceptions of these aids, either as external assistance or part of the task, may impact attribution of success, failure and therefore self-efficacy. This may be particularly important for females, if their self-efficacy is already lower than the males before they begin to use the environment features. Hence the following hypothesis:

> *Hypothesis SE-2: Due to gender differences in self-efficacy and attribution of success females will be more likely than males to attribute their task success to the end-user problem-solving environment features because they view the features as external help.*

If this first hypothesis is true, we further hypothesize the following:

> *Hypothesis SE-3: In end-user problem-solving environments, attributing success to the environment features, rather to oneself, will have negative consequences on self-efficacy and therefore engagement with the software.*

Further compounding females’ attribution of success to external factors is the imposter syndrome, where individuals (typically females) give little credit to their actual accomplishments [Clance and Imes 1978]. The imposter syndrome pertains mainly to successful individuals who have been successful in life, but perceive their success is not due to actual ability, and that at any point they will be uncovered as the “fraud” they really are. Although not limited to being a female phenomenon, females are most commonly afflicted with these beliefs.

One of the suggested steps to overcoming the imposter syndrome (suggested by Young on her webpage [2006]), is: “Reward yourself. Break the cycle of continually seeking - and then dismissing - validation outside of yourself by learning to pat yourself on the back.” Perhaps affective rewards are one way to encourage this type of behavior of reward. Affective rewards, in contrast to functional rewards (such as finding a formula error when working with a spreadsheet), have been found to influence people’s problem-solving effectiveness [Ruthruff et al. 2004]. In fact, one study found that affective rewards (in the form of progress bars and other slight color changes – none of which impacted the functionality of the abilities of the features to aid in the task) had a significant impact on users’ task success and their understanding of the features [Ruthruff et al. 2004].

> *Hypothesis SE-4: End-user problem-solving environments that make use of affective rewards are more likely to benefit females’ self-efficacy and engagement with the software compared with environments that do not offer these rewards.*

In an interview study Zeldin and Pajares found that for females in math and technology careers, their most often discussed self-efficacy forming memories came from verbal persuasion, specifically from support of important others close to them (such as strong parental support while growing up) [Zeldin and Pajares 2000]. The women in the study did not discuss performance accomplishments. Unfortunately for our interests, the researchers did not interview men in the same positions to assess if the men and women had different views on how they got to their careers. (For example, would the men have discussed their performance accomplishments?)

How to take advantage of self-efficacy formed through verbal persuasion in a problem solving environment is an open research question.

2.1.2.3 Emotional Arousal

Emotional arousal, including physiological states, also plays a role in determining self-efficacy. Physiological stressors (e.g., perspiring, upset/nervous stomach) impact perceptions of self-efficacy. However, the exact mechanisms by which they impact self-efficacy depends upon the perceived physiological state. Two individuals may attribute physiological arousal differently because they single out different factors of their stress and view them with different meanings [Bandura 1986].

For example, for some individuals getting nervous before going on stage, the physical symptoms associated with their anticipation (e.g., perspiring) are interpreted as distress reflecting personal failings. Yet, others may view this same physical reaction as a normal part of preparing to go on stage and attribute it to excitement, not to impending failure. Although both experience the same physiological stressors, their attribution of the stressor differs, and therefore its impact on self-efficacy also differs. An individual attributing their perspiring to their personal failings will likely have lower self-efficacy, influencing their on-stage performance.

2.1.2.4 Emotional Arousal & Gender Differences

Bandura highlights research suggesting that one way people learn to interpret physiological state and tie it to emotion is through "social labeling" (as discussed in [Bandura 1986]). Children mostly learn social labeling through their parents: parents notice a child's reaction to a particular external event and help their child label emotions surrounding the event. In this manner children learn to associate physiological symptoms with emotions.

All children are not treated equally, however, and researchers suggest that gender is a determining factor in differences between how parents act toward their children (even going as far as describing newborn babies differently depending on the baby's gender [Karraker et al. 1995]). For example, girls are more often in high-structured activities where they communicate with adults (more details provided in Section 2.2.1), whereas boys spend more time in low-structured activities with less adult interaction. (See [Meyers-Levy 1989] for a thorough discussion of this research.) Therefore, the opportunities children have to learn social labeling are likely to differ by gender.

Perhaps these gender differences in social labeling and other interactions with parents begin to influence factors related to computer phobias, sometimes referred to as technophobia [Brosnan 1998]. These emotional reactions toward computers and software have a direct impact on self-efficacy, and past research reports them to be more common among females [Brosnan 1998].

Thus, in end-user problem-solving environments emotional arousal is another potential factor affecting males and females in different manners:

> *Hypothesis SE-5: Gender differences in negative attribution of emotional arousal will impact males' and females' self-efficacy differently, further impacting the type of engagement with features during problem-solving tasks.*

2.2 Selectivity Hypothesis (Information Processing)

Meyers-Levy generated a theory called the "Selectivity Hypothesis" to bring together numerous theories of gender differences with respect to information processing [Meyers-Levy 1989]. The theory states that males tend to process information in a heuristic manner, paying particular attention to cues that are highly available and particularly salient in the focal context. Females, on the other hand, process information in a comprehensive manner, attempting to assimilate all available cues [Meyers-Levy 1989].

Males tend to focus on themselves when processing information, which helps to streamline the processing because "information pertaining to the self is represented in memory by a particularly well-developed and elaborate network of associations" [Meyers-Levy 1989]. This differs from the females who devote processing equally to information relevant both to others (external world) and themselves.

For computer-based problem-solving, gender differences in information processing may affect what environmental cues males and females process while problem-solving, and further may impact problem-solving decisions. Hypotheses relating to this general idea will be proposed throughout this section.

2.2.1 Gender Differences in Children with Respect to the Selectivity Hypothesis

The environment in which children are raised is hypothesized as a contributing factor to gender differences in information processing. One potentially relevant factor is the primary activity structure a child engaged in during childhood. Researchers have defined high- and low-structure activities. This research (summarized by [Meyers-Levy 1989]) found that girls were more likely to engage in high-structure activities, boys in low-structure activities. High-structure activities are characterized by "individual instruction, group feedback, and provision of information due to greater adult accessibility and behavior modeling" [Meyers-Levy 1989]. Low-structure activities are characterized by more "task initiations, leadership attempts, aggression, and peer commands" [Meyers-Levy 1989].

These gender differences toward high- and low-structure activities are driven by social agents (e.g., parents), who tend to encourage and keep girls in greater proximity to adults [Meyers-Levy 1989]. In fact, Margolis and Fisher cite several studies which show that girls are kept physically closer to home compared to boys, often for the non-verbally stated reason that they could be in danger [Margolis and Fisher 2003]. Boys, on the other hand, in their low-structure activities, have less contact with the same social agents, thereby receiving less feedback about their interactions with their environment. Boys' low-structure environment encourages heuristic processing of the most salient environmental cues. In contrast, the high-structure activities girls engaged in lend themselves to more comprehensive style of processing of many sources of information.

2.2.2 Gender Differences in Adults with Respect to the Selectivity Hypothesis

Meyers-Levy highlights research in several areas relating to gender differences in adults. For example, females tend to be more critical in their self-evaluations than males are, especially when clear performance feedback is unavailable. Females' evaluations are sensitive to task-relevant experiences, while males' self-evaluations remain high regardless of prior experiences [Meyers-Levy 1989]. These self-evaluations are likely to closely relate to the formation of self-efficacy as well, making it important for both genders to have adequate access to information necessary to perform self-evaluations.

Related to self-evaluations is the lack of accurate self-evaluations in the form of overconfidence. Overconfidence by humans about their own performance is a well-known and robust finding in behavioral science research; Panko's survey of a number of findings from multiple domains has helped to document its pervasiveness [Panko 1998]. In particular, overconfidence in spreadsheet correctness is common [Panko 1998]. Lunderberg et al.'s work found that, while both males and females were often overconfident, males were significantly more overconfident in their incorrect answers for math-based computational skills [Lunderberg et al.1994]. Pulford and Colman also report that males were significantly more overconfident than the females [Pulford and Colman 1997]. In research on problem-solving environments, specific features have aided users in becoming less overconfident [Burnett et al. 2003]. This result plus males' greater tendency toward overconfidence leads to the following hypothesis:

> *Hypothesis SH-1: End-user problem-solving environments providing clear feedback will lessen males' likelihood of being overconfident, and the same types of clear feedback will also aid females in being less critical in their self-evaluations.*

In another area of gender differences highlighted by Meyers-Levy, researchers in the 1960's found that males and females categorize and classify information into categories differently. Females create more categories than males, and place statements with conceptual similarity into those categories in more consistent ways than do males [Meyers-Levy 1989]. This information suggests that females see, and/or create, more subtle distinctions between groups of information than males do.

Hypothesis SH-2: End-user problem-solving environments that support the ability for flexible classification of information (for example, classifying if a spreadsheet cell's value is correct/wrong/maybe correct, etc.), may accommodate females' preferences for finer granularity more than environments supporting only broader classification groups.

2.2.3 Application of the Selectivity Theory in Practice

After developing the Selectivity Hypothesis, Meyers-Levy empirically evaluated the implications of the theory within the context of marketing. Meyers-Levy and Sternthal [1991] looked at what level of "noticeably" a cue embedded within a statement about a particular item would have on each gender's ability to notice and process that cue. For example, in the description of toothpaste they included two words that suggested the toothpaste had a bad taste. The words were either placed together or separated within the description, but always appeared somewhere in the middle of the description. Participants in the study then were asked to try the toothpaste, and following this they recalled as much of the description as possible and commented on the toothpaste. There were no gender differences in the recall of the particular information within the description. However, placing the two words together regarding the taste impacted the females' judgment of the taste; females found the taste less favorable than the males' taste perceptions. In the condition when the words were separated in the text there was no difference between the males' and females' taste perceptions (both found the taste favorable) [Meyers-Levy and Sternthal 1991]. In other words, the bad taste cues became salient for the females when they still went unnoticed by the males.

In another study reported in the same paper, Meyers-Levy and Sternthal concluded that although males and females read the same text, females' judgments made on the basis of the text reflected greater consideration of the message cues than did those of the males. This research suggested that females are more likely to elaborate on the cues within texts than males. The authors' interpretation is that this is due to females' lower elaboration threshold [Meyers-Levy and Sternthal 1991].

In end-user problem-solving environments, these slight differences in perceptions could negatively impact the males. Often errors within spreadsheet formulas can be subtle, and if the males are more likely to gloss over them, their information processing strategy could work against them:

> *Hypothesis SH-3: Gender differences in information processing may affect males and females differently in their search for errors, with males being more prone to overlook specific cues about the location of errors within problem-solving environments such as spreadsheets.*

The Selectivity Hypothesis has been studied in areas other than marketing, including auditing [O'Donnell and Johnson 2001] and web page perception [Simon 2001]. In the latter, Simon [2001] conducted a study of various culture groups looking for gender differences in perception of web sites. He found a significant effect of gender differences on the perception that a webpage was appropriate for his/her home country. He surmised that this difference was due to levels of information processing by the different genders. His interpretation was supported by the argument that more of the female respondents indicated a preference for more information on all the web pages viewed [Simon 2001]. Although perceptions of web pages are not directly linked to problem-solving, these findings suggest that selectivity-related gender differences are relevant to computers, and lead to the following hypothesis:

> *Hypothesis SH-4: Gender differences in information processing will impact the amount of information males and females desire prior to making problem-solving decisions. In particular, females may desire more information than males.*

2.2.4 Focus on Self versus Self + Others

The research of Meyers-Levy focused on how males were more likely to consider only "self" in their decision making process whereas females more often included other factors, along with self in their decision making. This difference has been found in motivations for engagement with technology as well.

Researchers have found that computer science females are motivated by how technology can help other people, whereas males tend to enjoy technology for its own sake [Margolis et al. 1999]. As an example, the following quote is from a computer science female at Carnegie Mellon, describing why she chose to major in computer science:

> "I think with all this newest technology there is so much we can do with it to connect it with the science field, and that's kind of what I want to do (study diseases) … Like use all this technology and use it to solve the problems of science, the mysteries" [Margolis et al. 1999].

Miller, a leader in women's psychology issues, highlights that, in general, women often make it their life work to serve others, while men are specifically discouraged from doing so by society [Miller 1976].

Table 2-2. Females and Males Technology Fantasies: Summarization of gender differences in ways males and females fantasized about technology [Brunner et al. 1998].

	Women …	**Men …**
1	Fantasize about it as a **medium**	Fantasize about it as a **product**
2	See it as a **tool**	See it as a **weapon**
3	Want to use it for **communication**	Want to use it for **control**
4	Are impressed with its potential for **creation**	Are impressed with its potential for **power**
5	See it as **expressive**	See it as **instrumental**
6	Ask it for **flexibility**	Ask it for **speed**
7	Are concerned with its **effectiveness**	Are concerned with its **efficiency**
8	Like its ability to facilitate **sharing**	Like its ability to facilitate **autonomy**
9	Are concerned with **integrating** into their personal lives	Are intent on **consuming** it
10	Talk about wanting to **explore** worlds	Talk about using it to **exploit** resources and potentialities
11	Are **empowered** by it	Want **Transcendence**

These differences are consistent with reports on other females who use technology, such as architects, NASA scientists, and filmmakers. In one study [Brunner et al. 1998], females and males were asked to write a science fiction story in which the perfect technological object is described. The females described objects as tools to help integrate personal and professional lives and to facilitate creativity and communication. The male's descriptions, however, used the technological device to increase command and control over nature and one another. Brunner et al. then summarized some of the distinctions between how males and females viewed technology, shown in Table 2-2. Their findings are consistent with Bennett et al.'s research on girls' and boys' gaming interests [Bennett et al. 2004].

Table 2-2 leads to the following hypotheses:

> *Hypothesis SH-5: End-user problem-solving environments that support "productizing" a user's program (for example by allowing a user to keep his "source code" private) will be perceived to be more valuable by male end-user programmers than environments that do not have these features.*
>
> *Hypothesis SH-6: End-user problem-solving environments that support communication (for example by connecting users working on similar environments though the network) will be perceived to be more valuable by female end-user programmers than environments that do not have these features.*
>
> *Hypothesis SH-7: End-user problem-solving environments that support sharing will be perceived to be more valuable by female end-user programmers than environments that do not have these features.*

2.3 Attention Investment

Unlike Self-Efficacy theory and the Selectivity Hypothesis, the theory of Attention Investment [Blackwell 2002] was proposed from the perspective of choices made during technology use. The theory proposes a model of how end users make decisions (to use particular environment features, for example) when engaged in problem-solving. Specifically, Attention Investment is an analytical model of user problem-solving behavior that allows a designer to account for the costs, benefits, and risks users weigh in deciding how to complete a task. For example, consider a programmable phone. If the ultimate goal is to make a phone call, then programming the number into the phone has a cost, benefit, and risk. The cost is figuring out how to program the phone. A benefit is the freedom to forget the phone number. The risk is that the "program" might not work as the user intended.

Notice that it is the user's *perception* that matters, not the actual cost to use a feature, nor the actual risk. Cost and benefit are measured in attention units—for example, the cost is the perceived amount of attention to do the particular task while the benefit is the saved attention for future occurrences of this task. Risk is measured as a percentage (probability of failure).

A second example of the Attention Investment Model illustrates a scenario relevant to problem-solving in the context of a spreadsheet. A female end user has a spreadsheet calculating budget expenses, and needs to submit figures from the spreadsheet to her boss in just a few hours. Beyond her time constraints she needs to be confident that her budget (calculated through multiple formulas in a spreadsheet) is correct. As she decides how to spend this time she (perhaps subconsciously) starts weighing alternatives in her head. She knows there are features within the spreadsheet environment which could help her ensure a reliable spreadsheet; however, she has never used those features before so she would have to spend some time learning them (a cost), and then use them on her spreadsheet (also a cost). If she decides to learn how to use the features now, though, she will also be able to use them in the future (benefit). She also considers the possibility that her efforts will not pay off in a spreadsheet delivered on time with an accurate budget (risk).

2.3.1 Risk Perception

Marketing research has looked closely at risk perception. One definition of risk perception is "the combination of uncertainty plus seriousness of outcome" ([Bauer 1960] as referenced in [Featherman and Fuller 2003]). Marketing researchers have discovered that if an individual feels uncertain, uncomfortable, and/or anxious toward a new service then the greatest influence on the adoption decision is the individual's risk perception (as discussed in [Featherman and Fuller 2003]). This may apply to end users deciding to use new features in problem-solving environments as well.

2.3.2 Risk Perception and Gender

Risk may play a different role in males' and females' choices to engage with features in problem-solving environments, as suggested by research pertaining to gender differences in general risk perception. Females perceive more risk from everyday life decisions and situations than do males [Barke et al. 1997, Blais and Weber 2001, Byrnes et al. 1999, Finucane et al. 2001, Gustafson 1998, Hudgens and Fatkin 2001, Jianakoplos and Bernasek 1998]. Further, Brosnan's research on females' higher computer anxiety levels [Brosnan 1998] also ties in with factors affecting perception of risk, as mentioned above.

Researchers have also found that females are more risk averse in their financial decisions than males [Jianakoplos and Bernasek 1998]. Females are also more risk averse in "informed guessing" [Byrnes et al. 1999]. Informed guessing is the willingness to make an educated guess on questions when the result of an incorrectly answered question is negative (such as losing points).

Drawing from this research we hypothesize the following:

> *Hypothesis R-1: Gender differences in perception of risk may impact females' more than males' willingness to make use of unfamiliar devices in end-user problem-solving environments.*

Gender differences in avoidance of risky behavior were also borne out in a study Hudgens and Fatkin [2001] conducted investigating risk taking in a simulated battleground simulation. Participants (from the military) were asked to make decisions about sending a tank across a mine-laden field, with mines at various densities. Although not apparent immediately, the females were more risk averse than the males; females were more reluctant to send the tank across in conditions in which the males did so more willingly [Hudgens and Fatkin 2001]. Furthermore, females and males also reported distinct strategy differences in their decision making process. The females based their decision to send the tank based on the overall density of the minefield. In contrast, the males analyzed the layout of the mines in the field to determine the number of safe paths before making their decision [Hudgens and Fatkin 2001]. This suggests that the females were more risk averse and that gender differences in strategies of analysis might have an impact on why there are differences in risk taking behavior.

Hypothesis R-2: Gender differences in risk perception may impact the strategies by which males and females engage in end-user problem-solving environments.

Other research also found differences in users' actions when they perceive higher risk; Featherman and Fuller [2003] found that individuals with a high perception of risk increased their information seeking. Information seeking closely relates to the research on the impact of the Selectivity Hypothesis. In one study on perceptions of web pages, more females reported wanting additional content on all of the web pages they viewed than males who wanted additional information [Simon 2001]. For end-user problem-solving environments, information seeking behavior may increase in situations users perceive as risky. These situations may provide end-user problem-solving environments an opportunity to provide information to help users assess the risk. This may unequally affect females due to their often higher perceptions of risk.

Hypothesis R-3: Due to females' high risk perceptions, with resulting increases in their information seeking behavior, end-user problem-solving environments that provide additional information about potentially risky aspects of the environment or common tasks may keep females more engaged during problem solving than environments that do not do so.

The impacts of gender differences in risk perception may be important for problem-solving software. For example, if women perceive the risk of taking an action regarding a particular environment feature, such as assertions, as being higher than men do, and if their perceived risk results in avoidance behavior (not using features that might help them fix and avoid errors), then the result could be a less robust program, leading to possible future errors, undetected errors, and ultimately erroneous decisions based on the erroneous output.

2.4 Models of Technology Acceptance

MIS (management information systems) researchers study how and why users adopt information technologies. This section focuses on two models of technology acceptance[2] (TAM and UTAUT), which researchers have studied in the context of introducing new technologies into workplace settings.

The theories introduced in this section are computer science related theories. They do not necessarily focus on end users as their primary audience, and the technologies studied are general software technologies, not necessarily end-user problem-solving environments. Nevertheless, we believe there are strong ties to the more specific research of end-user problem solvers.

2.4.1 TAM

The technology acceptance model (TAM) theorizes that users are most influenced by two factors in their intention to use technology: perceived usefulness and perceived ease of use [Davis 1989]. Davis defined perceived usefulness as “the extent to which a person believes that using a particular technology will enhance his/her job performance,” and perceived ease of use as “the degree to which a person believes that using a technology will be free from effort.” In later research a third variable was added, subjective norm, defined as “the degree to which an individual believes that people who are important to her/him think s/he should perform the behavior in question” [Venkatesh and Morris 2000].

Since TAM’s creation, researchers have conducted studies of TAM in numerous variations, but one type of study in particular is most relevant to our research. The relevant studies are those in the context of a workplace where a new software

[2] The two theories were chosen because of the gender differences researchers have discovered.

technology is being introduced. The general procedure followed for these studies is that participants are introduced to the new software technology through a tutorial, which is followed by a questionnaire assessing their perceived ease of use, perceived usefulness (and in later studies subjective norm), and whether they intend to use the technology for their work [Davis 1989, Venkatesh and Morris 2000, Venkatesh et al. 2003]. After several weeks or months the researchers return to assess participants' actual usage and again measure their perceived ease of use and perceived usefulness and relate these to intention to continue using the software.

In these studies researchers have consistently found that perceived usefulness and perceived ease of use are both significant predictors of intention to use new software immediately after the introduction to the software [Davis 1989, Venkatesh and Morris 2000, Venkatesh et al. 2003]. For example, in one study these factors accounted for intention to use a technology: individually, perceived usefulness and perceived ease of use accounted for 85% and 59% respectively, and together they accounted for 56% [Davis 1989]. At later measurement times, however, perceived ease of use diminished as a predictive factor of intention to use the technology, while usefulness remained predictive.

2.4.2 UTAUT

TAM was adapted from another theory (the Theory of Reasoned Action) modeling human behavior, but modified to fit the context of information technology. However, other researchers had developed other models as well. In an effort to develop a unified model of user acceptance researchers tested eight models, and used the results to develop one model [Venkatesh et al. 2003]. They approached this task by testing the eight models in a study of four companies introducing a new software technology. Their unified model, UTAUT (Unified Theory of Acceptance and Use of Technology), is similar to TAM, although it includes several more direct and indirect factors in predicting intention to use; see Figure 4. Their own study to verify the

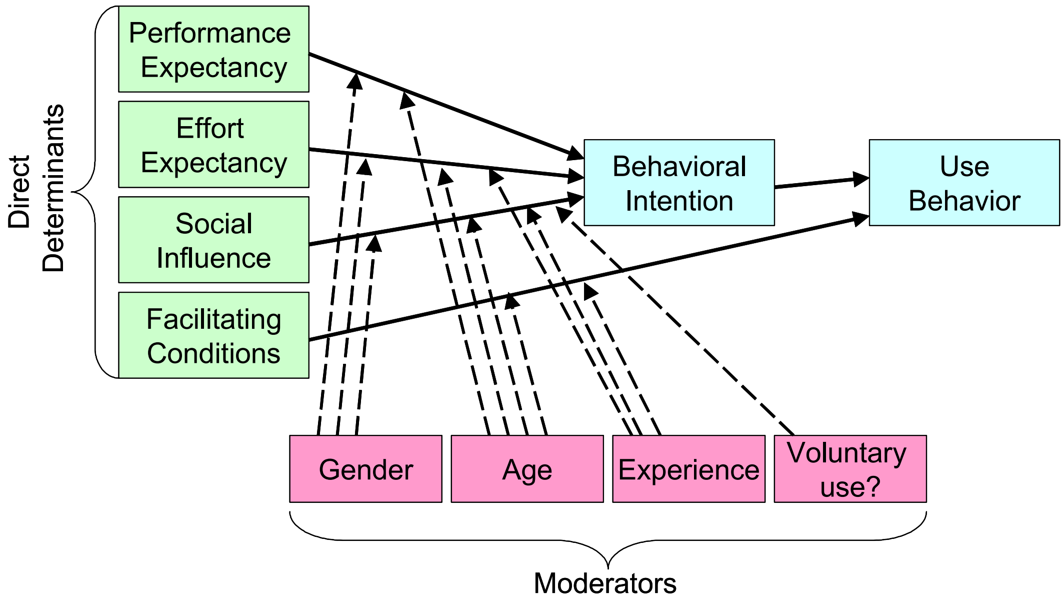


Figure 4. UTAUT Model: Researchers tested eight models of user acceptance theories to derive this model [Venkatesh et al. 2003].

predictive power of this model found that the UTAUT model accounted for 70% of participants' intention to use technology.

The direct factors are those which were found to impact individuals' intention to use a technology or actual usage. The moderating factors (including gender, age, experience with the technology, and if the technology can be used voluntarily) mediate between the direct factors and behavioral intention.

The direct determinants include:

1.) Performance Expectancy: "the degree to which an individual believes that using the system will help him or her to attain gains in job performance."

2.) Effort Expectancy: "the degree of ease associated with the use of the system."

3.) Social Influence: "the degree to which an individual perceived that important others believe he or she should use the new system."

4.) Facilitating Conditions: "the degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system."

2.4.3 User Acceptance Models for End-User Problem-Solving Environments

These models have ties to end-user problem solving because end users may be choosing to use a new software environment due to its potential for automating some otherwise repetitive task. In this case, the models of technology acceptance closely relate to end users automating tasks using, for example, a new software package. Another way the technology acceptance research may relate is when end users are already familiar with a software package (spreadsheets are an example), but there are new (unfamiliar to the user) features within the software that could aid their task. The acceptance of these unfamiliar features may relate to the research on technology acceptance.

2.4.4 Gender & User Acceptance Models

Venkatesh and Morris studied gender differences in the relative importance of the TAM's components on decision to use the introduced technologies [2000]. They gathered participants in five different organizations and took measurements at three times: after the initial day-long tutorial, and two long-term measurements at one and three months. (Measurements included perceived usefulness, perceived ease of use, subjective norm, and intention to use.)

Figure 5 shows the predictive relationship of three factors for males and females on intention to use the new software technology. In general, for the males, the only factor that was predictive of behavior intention was perceived usefulness. For the females all components (ease of use, usefulness, and subjective norm) were equally important for predicting intention to use the introduced technology. The authors concluded that for males, productivity related factors were the most important for technology adoption and use decisions, while for females productivity was just one factor in the decision to adopt and use technology [Venkatesh and Morris 2000].

Gender differences were also found in UTAUT that parallel the differences in TAM, although age moderated the relationships as well [Venkatesh et al. 2003]. For example, performance expectancy (perceived usefulness in TAM) was more influential for males' and young workers' intention to use, while effort expectancy was more important for females' and older workers' intention to use, particularly early on in the experience with the new technology [Venkatesh et al. 2003].

Both self-efficacy and perception of risk are implicated as influencing perceptions of

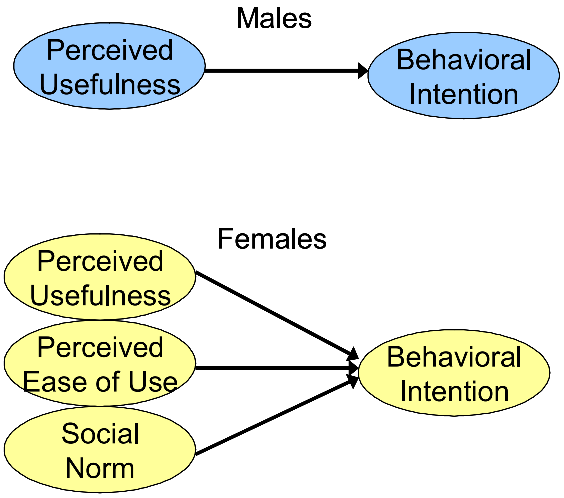


Figure 5. Technology Acceptance by Gender: Show the differences in the factors that predict intention to use newly introduced software technology between males and females. These same relationship existed immediately after training, and one month later. At three months after training social norm drops from the females' model.

ease of use and usefulness [Venkatesh and Morris 2000, Featherman and Fuller 2003]. Venkatesh and Morris [2003] suggested that perceived ease of use and self-efficacy are closely related to one another since self-efficacy affects an individual's judgment about the ease of a task. (Perception of risk also lowered the perception of ease of use in a study by [Featherman and Fuller 2003].) Therefore, lower self-efficacy is expected to result in lower perceptions of perceived ease of use. This may help to describe why ease of use was more important for the females in intention to use the technology in Venkatesh and Morris' study; perhaps the females had lower self-efficacy about their ability to use the newly introduced technology.

The gender differences in TAM and UTAUT may tie to a problem-solving environment's features, such as our surprise-explain-reward strategy[3]. For example, in our prototype, when users are introduced to a new error-guarding feature through the surprise-explain-reward strategy, the explanation can influence perceptions of both ease of use and usefulness, and therefore intention to use the new feature.

Hypothesis TA-1: An end-user problem-solving environment that emphasizes the potential usefulness of its features (such as in its on-line help content) will be perceived as being more valuable, and encourage use of these features, by male end users than an environment that does not emphasize usefulness of its features.

Hypothesis TA-2: An end-user problem-solving environment that emphasizes the ease of use and usefulness of the features, and allows for particular social norms about those features, will be seen as more attractive to the females, and will influence their choice to engage with the features, compared with environments that do not have the same balance of emphasis.

[3] The surprise-explain-reward strategy relies on users becoming curious about particular features to entice them to use a feature they may have not used previously. In general, if a user is surprised by or becomes curious about any of the feedback environment features, he or she can seek an explanation, available via tool tips. If the user follows up as advised in the explanation, rewards potentially ensue.

2.5 Curiosity: Information-Gap Theory

Loewenstein's information-gap theory draws several earlier theories on curiosity into one theory [Loewenstein 1994]. Loewenstein defines "information gap" in terms of two types of information: what ones knows, and what one wants to know. According to Loewenstein, "curiosity… arises when one's informational reference points in a particular domain become elevated above one's current level of knowledge" [Loewenstein 1994], where the "informational reference point" is what one wants to know.

The information-gap theory was the backbone in the development of the surprise-explain-reward strategy (see Footnote 3 and Chapter 3). The development and success of the strategy relies on raising a user's curiosity to an ideal level, such that he/she becomes aware of missing knowledge, but perceives it as attainable. For this reason, understanding the underlying theory of the information-gap theory is important for the design of problem-solving environments that depend on it. (Recall, that when people perceive risk they may start seeking information, as discussed in Section 2.3.2.)

Two types of information provoke curiosity: an incremental gain in knowledge versus a flood of insight that can help reveal an entire problem and solution. (For example, imagine a covered picture, in which removing one part of the covering could provide insight into the whole picture.) For incremental gains in knowledge, however, any new single piece of information is unlikely to lead to sudden solutions. According to the theory, when insights are just around the corner curiosity increases more than when an incremental gain in knowledge occurs.

In the case of incremental problems, curiosity is expected to increase as one begins to understand more about a problem or general information space. Loewenstein discusses this from the following perspective: if a person knows 3 states out of the 50 US states they are likely to focus on their knowledge of those 3 states [Loewenstein 1994]. In contrast, knowledge of 47 of the 50 states will likely put more emphasis on the three unknown states and raise an individual's curiosity about what these three states are. (Individuals, of course, need to be aware that information is missing in order for incremental problems to noticeably increase curiosity, e.g., that there are 50 states total.)

The surprise-explain-reward strategy relies on users perceiving their own missing knowledge. The following examples illustrate two ways this kind of acknowledgment of missing information can occur.

A user familiar and comfortable with several features in a problem-solving environment encounters a feature they have not used previously. The original encounter takes place, for example, because of the system bringing something to the attention to the user which makes the user aware of this new feature, or through the user's own exploration of the environment. In both cases the user has been confronted with an unfamiliar feature, and may become curious about it. According to the information gap theory this is one manner in which individuals become curious, through "directly confronting an individual with missing information," of which this new feature is an example [Lowenstein 1994].

In a second scenario, curiosity is generated through a violation of expectations. One of the "surprises" in the surprise-explain-reward strategy is to get the users' attention and increase their curiosity through generating assertions (a value range that a spreadsheet cell should fall within) that are purposely not "reasonable" numbers (for example, a range suggested by the computer might be -189 to 5,637). This type of surprise is expected to gain the attention of the users because it is a "violation of expectation." (A reaction akin to "what's this odd range?") According to Lowenstein [1994], this type of information gap "often triggers a search for an explanation" – which is where the explanation portion of the surprise-explain-reward strategy becomes important.

A critical component of curiosity comes from revealing an information gap of "just the right size." An information gap that is too small often leaves individuals feeling bored, and too large the opposite: rather than getting curious they are overwhelmed, and are unlikely to benefit from the prospect of new information. This description of curiosity is depicted as an inverse U, as shown in Figure 6.

We hypothesize that females, based on their often high computer anxiety, will more often fall into the "zone of anxiety," potentially harming their ability to get the same benefit that males get out of intentional violations of expectation within a problem-solving environment. This is summarized in the following hypothesis:

> *Hypothesis C-1: Gender differences in self-efficacy with end-user problem-solving situations could impact the effectiveness of using a "surprise" to capture the user's attention and curiosity, and instead cause the user to avoid supporting features.*

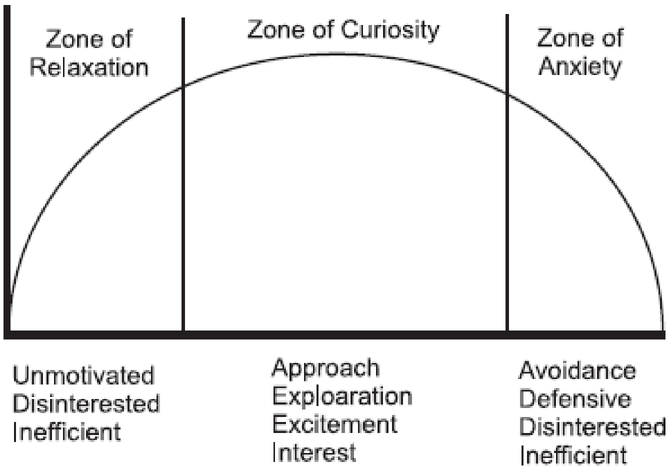


Figure 6. Inverted U of Curiosity: Curiosity with relation to the size of the information gap [Arnone and Small 1995]. Curiosity (y axis) increases as the information gap increases (x axis). However, when the information gap becomes too large, the curiosity begins to decrease again.

2.6 Summary

Drawing from others' research, we have developed a series of hypotheses related to potential gender differences in end-user problem-solving environments, specifically regarding how each gender interacts with the environment, and to specific problem-solving features within the environment. Table 2-3 lists each of the hypotheses.

A few select hypotheses are investigated in subsequent chapters.

Table 2-3. Hypotheses: Drawing from five theories and other research these hypotheses relate to gender differences that may impact end users using computers for problem solving activities.

Self-Efficacy
Hypothesis SE-1: Gender differences in computer self-efficacy will be evident in differences in attitudes toward and engagement with unfamiliar features within problem-solving environments.
Hypothesis SE-2: Due to gender differences in self-efficacy and attribution of success females will be more likely than males to attribute their task success to the end-user problem-solving environment features because they view the features as external help.
Hypothesis SE-3: In end-user problem-solving environments, attributing success to the environment features, rather to oneself, will have negative consequences on self-efficacy and therefore engagement with the software.
Hypothesis SE-4: End-user problem-solving environments that make use of affective rewards are more likely to benefit females' self-efficacy and engagement with the software compared with environments that do not offer these rewards.
Hypothesis SE-5: Gender differences in negative attribution of emotional arousal will impact males' and females' self-efficacy differently, further impacting the type of engagement with features during problem-solving tasks.
Selectivity Hypothesis
Hypothesis SH-1: End-user problem-solving environments providing clear feedback will lessen males' likelihood of being overconfident, and the same types of clear feedback will also aid females in being less critical in their self-evaluations.
Hypothesis SH-2: End-user problem-solving environments that support the ability for flexible classification of information (for example, classifying if a spreadsheet cell's value is correct/wrong/maybe correct, etc.), may accommodate females' preferences for finer granularity more than environments supporting only broader classification groups.
Hypothesis SH-3: Gender differences in information processing may affect males and females differently in their search for errors, with males being more prone to overlook specific cues about the location of errors within problem-solving environments such as spreadsheets.

Hypothesis SH-4: Gender differences in information processing will impact the amount of information males and females desire prior to making problem-solving decisions. In particular, females may desire more information than males.
Hypothesis SH-5: End-user problem-solving environments that support “productizing” a user’s program (for example by allowing a user to keep his “source code” private) will be perceived to be more valuable by male end-user programmers than environments that do not have these features.
Hypothesis SH-6: End-user problem-solving environments that support communication (for example by connecting users working on similar environments though the network) will be perceived to be more valuable by female end-user programmers than environments that do not have these features.
Hypothesis SH-7: End-user problem-solving environments that support sharing will be perceived to be more valuable by female end-user programmers than environments that do not have these features.
Attention Investment
Hypothesis R-1: Gender differences in perception of risk may impact females’ more than males’ willingness to make use of unfamiliar devices in end-user problem-solving environments.
Hypothesis R-2: Gender differences in risk perception may impact the strategies by which males and females engage in end-user problem-solving environments.
Hypothesis R-3: Due to females’ high risk perceptions, with resulting increases in their information seeking behavior, end-user problem-solving environments that provide additional information about potentially risky aspects of the environment or common tasks may keep females more engaged during problem solving than environments that do not do so.
Technology Acceptance
Hypothesis TA-1: An end-user problem-solving environment that emphasizes the potential usefulness of its features (such as in its on-line help content) will be perceived as being more valuable, and encourage use of these features, by male end users than an environment that does not emphasize usefulness of its features.
Hypothesis TA-2: An end-user problem-solving environment that emphasizes the ease of use and usefulness of the features, and allows for particular social norms about those features, will be seen as more attractive to the females, and will influence their choice to engage with the features, compared with environments that do not have the same balance of emphasis.
Curiosity
Hypothesis C-1: Gender differences in self-efficacy with end-user problem-solving situations could impact the effectiveness of using a “surprise” to capture the user’s attention and curiosity, and instead cause the user to avoid supporting features.

3. Impacts of Self-Efficacy[4]

Our first investigations considered the impacts of self-efficacy on females' and males' debugging. We focused specifically on Hypothesis SE1 (see Table 2-3 in Chapter 2), which we used to generate the following research questions:

> *RQ1: Are there gender differences in self-efficacy that impact effective end-user debugging?*
>
> *RQ2: Are there gender differences in users' likelihood of acceptance of unfamiliar features in end-user programming environments?*

The methodology we followed to refine the above questions was to qualitatively investigate three sources of data: data from a small think-aloud study, data collected in a survey administered to a small psychology class, and data collected in previous empirical studies [Robertson et al. 2004, Ruthruff et al. 2005, Ruthruff et al. 2004] in which each participant's gender was collected (but the study did not investigate questions of gender). These three sources led us to develop more specific research questions. These specific research questions were then investigated quantitatively through a controlled experiment.

3.1 Qualitative Beginnings

Our initial research looking for gender differences was through a small think-aloud study. (In the remainder of this dissertation this study is referred to as the "think-aloud experiment.") This study replicated the design of an earlier study [Wilson et al. 2003]; we recruited participants from CS101, a computer literacy course for non-majors. The study was conducted one participant at a time, in which they talked aloud as they found and fixed bugs in a spreadsheet[5]. The participants' confidence in their bug finding and fixing ability provided our first evidence in gender differences in self-

[4] The contents of this chapter are based on [Beckwith et al. 2005a].

[5] Quotes from this think-aloud are presented later in this chapter, and in future chapters.

efficacy. The females' initial self-efficacy (measured using the questionnaire discussed in Section 3.2.1) was significantly lower than the males' (mean of 33.6 for the 8 females versus 39.7 for the 7 males, t-test: t=-2.18, df=13, p<0.05). Correspondingly, the females did not fix as many bugs as the males (mean of 2.5 bugs for the females, 6.6 bugs for the males, t-test: t=-6.4, df=13, p<0.01). Even with the small number of participants we found rather large differences in self-efficacy and performance between the females and males, which led us to further develop the first research question into two more specific research questions:

RQ1a: Are there gender differences in self-efficacy in the domain of end-user debugging?

RQ1b: Is self-efficacy tied to effectiveness in end-user debugging?

The survey study was motivated by a study by Torkzadeh and Koufteros [1994], who found that females in a business computer applications course had lower self-efficacy than males on computer file and software management activities, and other research showing that low self-efficacy affects females' perceptions of a software application before actual use [Hartzel 2003]. However, these studies were done several years ago and software has changed 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[6]. 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." Questions were answered on either a five-point Likert scale or a ranking of choices. The respondents were 21 psychology and business majors: 14 females and 7 males. Our survey results were extremely consistent with the above findings: in *all ten* of our questions about software confidence and

[6] This survey was conducted by Shraddha Sorte, and written up as part of her Masters Thesis.

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.)

Our think-aloud study produced results consistent with the survey. For example, a few female participants' reactions and attitudes toward new features attracted our attention:

> F1: "What's this little arrow doing? They're everywhere! So, I need to take this—oh, my goodness. Now what's happening? … too much happening."
>
> F2: "Oh my gosh. Well it's ... I pushed help[-me-test] and all the [values] are changing and a lot of things are changing, but I don't [know] what that color means, and I have no idea how to get it back, so I have to type it all in."

These quotes strongly suggest a sense of being overwhelmed by the features, and a perception of the features as a hindrance rather than a help toward their goal.

In three studies ([Robertson et al. 2004, Ruthruff et al. 2005, Ruthruff et al. 2004]) prior to the summer of 2004, we had collected each participant's gender, but had not previously analyzed the data by gender. Thus they provided an excellent source of independent data for qualitative follow-up of the phenomenon suggested by our survey and think-aloud. We proceeded to qualitatively look for gender differences with respect to engagement with features by creating graph profiles of feature usage for each participant. We graphed participants' actions by considering 100 second long intervals and actions taken between formula edits (since formula edits are often attempts to fix spreadsheet bugs – their assigned task). Figure 7 shows examples of these graphs.

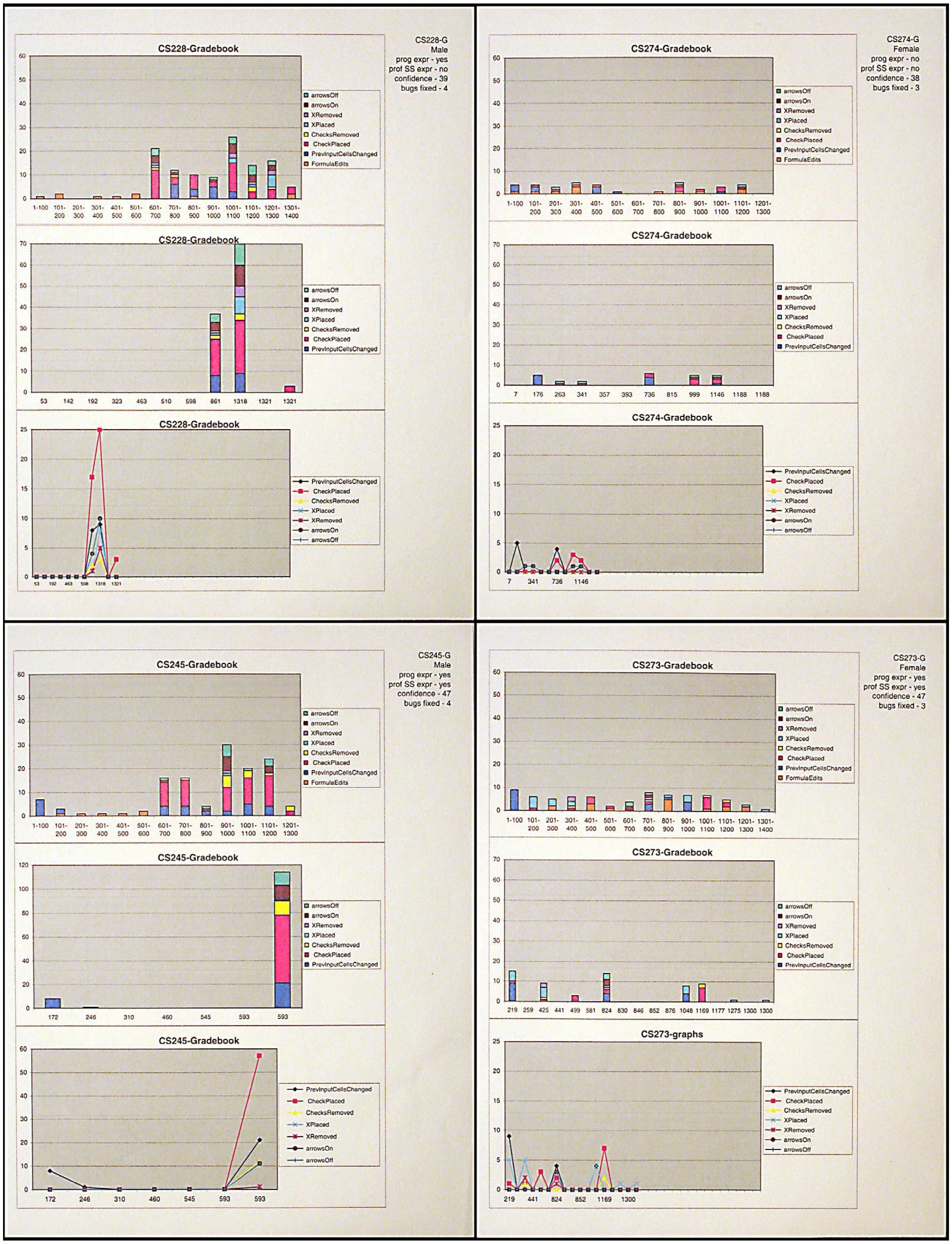


Figure 7. Feature Usage Graphs: Three graphs, representing usage of features over time, for each of two males on the left (CS228 and CS245), two females on the right (CS274 and CS273).

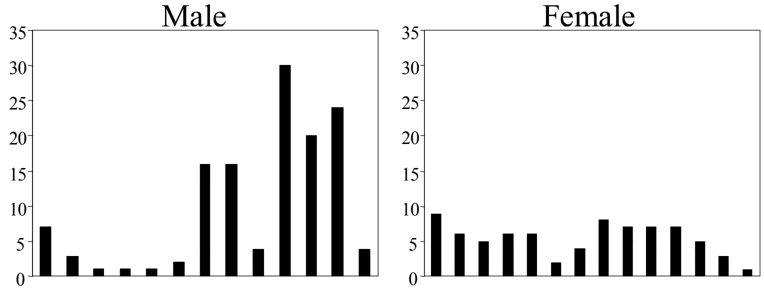


Figure 8. Activity Profiles: Activity profiles of one male (left) and one female (right). The male did more actions and used more features than the female. The horizontal positions represent points in time during the experimental task. Height represents frequency of debugging feature usage. Both participants pictured here were fairly representative of their genders.

To look for patterns and gender differences a group of five researchers gathered in a room with the graphs. Graphs were repeatedly sorted on some criterion visually represented in the graphs. (For example, one sort was done by amount of activity in each 100 second interval, in which participants were engaged with testing, visually represented by amount of red in the graph type at the top of Figure 7.) Once sorted on a particular criterion each group was recorded and analyzed for particularly intriguing gender differences. For example, one of these groupings led us to realize the pronounced differences between females and males in amount and type of activities as grouped by 100 second intervals. Figure 8 shows simplified profiles for one male and one female whose activity patterns were fairly representative of their genders.

In the design of one of these studies [Robertson et al. 2004], participants had been introduced to a feature, but not specifically taught how to use it. If they were interested in this feature they could find out more by reading explanations (tooltips). The analysis of how many tooltips were read by males and females during the two tasks showed a gender difference in when each gender became most interested in the new feature. In the first task the males read slightly more about the feature than the females, but the reverse trend was true for the second task, when, as can be seen in Figure 9, the females greatly increased their explanation reading, and the males'

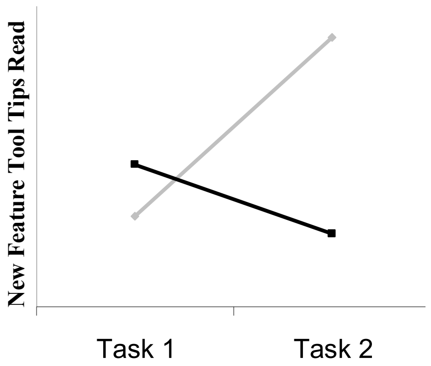


Figure 9. Explanations Read: Males' (dark line) mean interest in tool-tip accessible explanations describing new features started higher than females' (light line) and then declined, whereas females' interest increased.

reading decreased. This is further evidence for gender differences in feature usage, specifically for when new features are initially approached.

These findings of gender differences in use of features, and the timing of their interest led us to develop two specific research questions:

RQ2a: Are there gender differences in end users' willingness to approach new debugging features?

RQ2b: Are there gender differences in their willingness to then adopt these new features?

The rest of this chapter presents the design and results of a follow-up experiment we conducted to investigate all four research questions of Section 3.1.

3.2 Experiment

The experimental design is presented here. Note that the experiment's procedures, materials, tutorial, questionnaires, transcripting software, and tasks had all been used in or derived from previous studies ([Robertson et al. 2004, Ruthruff et al. 2005, Ruthruff et al. 2004, Wilson et al. 2003], except where other sources are stated below). In addition, they had been "tested" using analytical (cognitive walkthroughs) and empirical (pilot participants) methods.

3.2.1 Participants and Procedures

The 27 male and 24 female participants (mostly business students) started by filling out a pre-session questionnaire (see Appendix A for all questionnaires from this experiment) which collected participant background data and included the self-efficacy questions based on a slightly modified version of Compeau and Higgins' validated scale [Compeau and Higgins 1995]; the modifications made the questionnaire task-specific to end-user debugging. Participants were asked to answer on a five-point Likert scale their level of agreement with 10 statements. For example, "I could find and fix errors… if there was no one around to tell me what to do as I go," "…if I had seen someone else using it before trying it myself," and "…if I had a lot of time to complete the task."

The following background data were collected: gender, major, year or degree completed, GPA, programming experience (to bar participants with more programming experience than is usual for business students), spreadsheet experience, previous use of the study's prototype environment, and whether English was their primary language. We did not collect data on other factors that might seem relevant, such as mathematical ability, because the population of interest was spreadsheet users (other than trained programmers) at a roughly equivalent point in their academic careers, regardless of any other talents they may have. Statistical analysis of the background data showed that the females were academically younger than the males[7] (ANOVA: $F(1,48)=4.528$, $p<0.039$). There were no significant differences between the genders in any other background data collected.

3.2.2 Environment

The Forms/3 spreadsheet environment [Burnett et al. 2001], as described in this section, applies to each of the studies reported within this dissertation. In future

[7] Post hoc analysis using the non-parametric Kruskel-Wallis test showed that the difference in academic age was not predictive of any of the outcome measures (performance measures or behavior patterns) in this study.

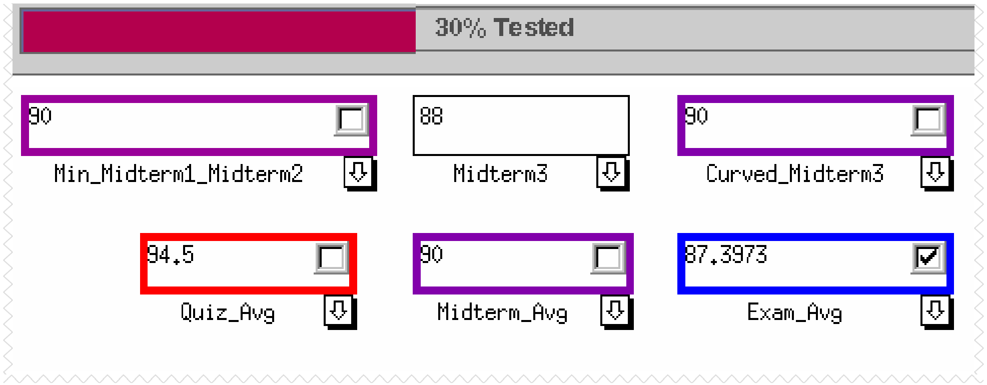


Figure 10. WYSIWYT: An example of WYSIWYT in Forms/3.

chapters, only distinctions between this version of the Forms/3 environment and other environment features will be reported.

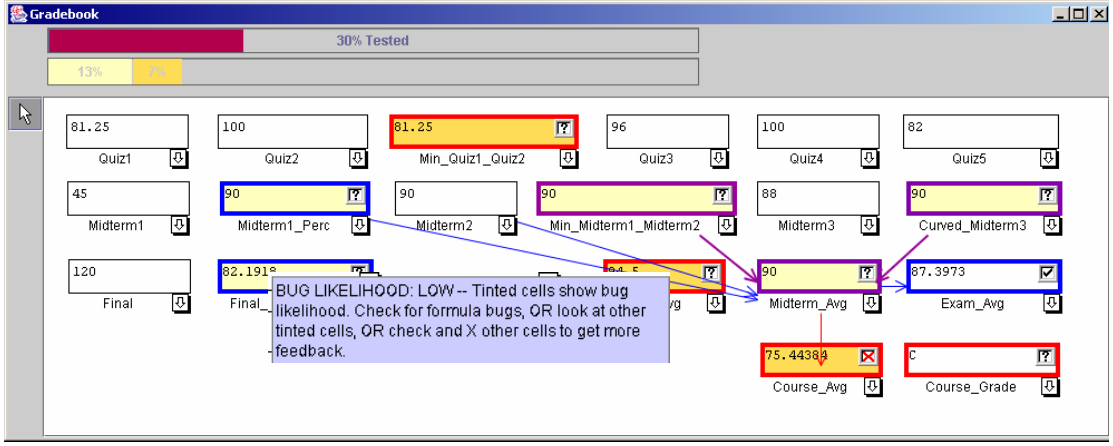
The debugging features that were present in this experiment were part of WYSIWYT ("What You See Is What You Test"). WYSIWYT is a collection of testing and debugging features that allow users to incrementally "check off" or "X out" values that are correct or incorrect, respectively [Burnett et al. 2004]. In addition, arrows that allow users to see the dataflow relationships between cells also reflect WYSIWYT "testedness" status at a finer level of detail.

The underlying assumption behind WYSIWYT is that, as a user incrementally develops a spreadsheet, he or she can also be testing incrementally. Figure 10 shows an example of WYSIWYT in Forms/3. In WYSIWYT, untested cells have red borders. Whenever users decide a cell's value is correct, 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. Also visible in the figure, the progress bar (top) reflects the testedness of the entire spreadsheet.

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, which cause interiors of cells suspected of containing faults to be colored in shades along a yellow-orange continuum, with darker orange shades given to cells with increased fault likelihood. Figure 11 shows an example of this behavior in one of the spreadsheets the participants debugged. The intent is to lead the user to the faulty cell (colored darkest orange).

The optional dataflow arrows are colored to reflect testedness of specific relationships between cells and subexpressions. (The user can turn these arrows on/off at will.) In Figure 11, the user has popped up Midterm_Avg's arrow, which shows both that Curved_Midterm3 is referenced in Midterm_Avg's formula and that this relationship is partially tested.

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



BUG LIKELIHOOD: LOW -- Tinted cells show bug likelihood. Check for formula bugs, OR look at other tinted cells, OR check and X other cells to get more feedback.

Figure 11. Gradebook Spreadsheet: The user notices an incorrect value in `Course_Avg`—the value is obviously too low—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 sources of the incorrect value, with some deemed more likely than others. In this figure, the tool-tip based explanation provides information about the interior coloring of the cell. Explanations are available on all environment features.

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 (Figure 11). 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 in this experiment was 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.

3.2.3 Tutorial

In the tutorial, participants performed actions on their own machines with guidance at each step. The tutorial motivated the necessity for testing through first having the participants notice the red borders (of the spreadsheet used during the tutorial), then inform the participants that testing is important because spreadsheet errors are common. Using an additional resource of a paper containing correct values for inputs and corresponding output the participants were encouraged to consider whether the value in a cell looked correct, and that checking it off would communicate their decision to the system. The feedback of the checkmark placement was pointed out and described (this included using arrows to help interpret and understand the feedback). After changing input values another checkmark was place, which increased the percent testedness of some cells which had only been partially tested from the first test.

In contrast to this level of detail on the checkmark and arrow features, the description of the X-mark feature was only to introduce participants to placing an X-mark if a value was wrong. After doing so, they were given about 30 seconds to explore the resulting changes in feedback. This was repeated on two other cells with additional time to explore. Finally, since the presence of a wrong value leads to a wrong

Table 3-1. Feature Categories: The classification of features within the environment according to whether they were taught or untaught during the tutorial, or participants were familiar with the feature prior to the study.

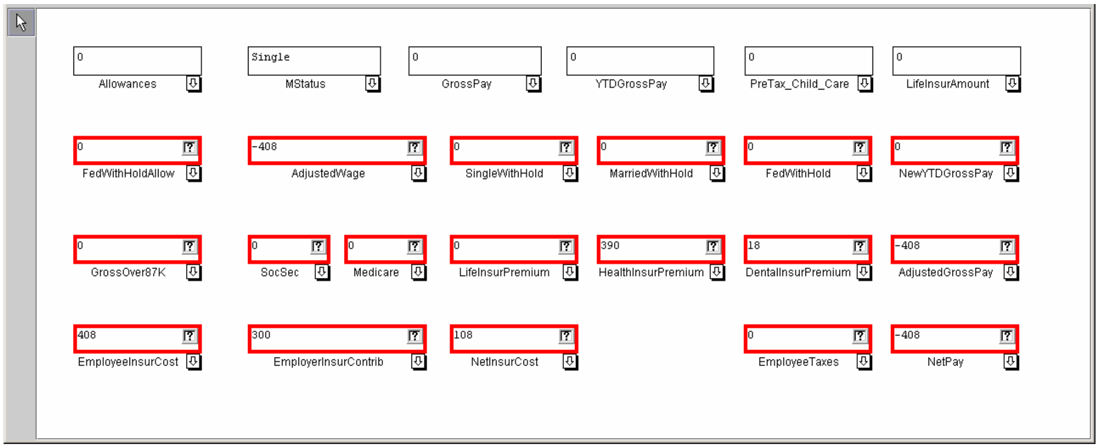
Category	**Feature(s)**
Type Familiar	Formula Edits
Type Taught	Checkmarks & Arrows
Type Untaught	X-marks

formula, participants have time to correct the formula on their own before the tutorial covers the change in detail.

This design allowed us to gather information on three types of "newness" of software features: (1) formula edits, which is a feature common to all spreadsheet environments, (2) the WYSIWYT features not previously encountered but explicitly taught (checkmarks and arrows), and (3) the fault localization (X-mark) feature, which was not taught at all. We labeled these three types of features as shown in Table 3-1.

3.2.4 Tasks

The experiment consisted of two spreadsheets, `Gradebook` and `Payroll` (Figure 4 and Figure 12). To make the spreadsheets representative of real end-user spreadsheets, Gradebook was derived from an Excel spreadsheet of an (end-user)



0 Allowances
Single MStatus
0 GrossPay
0 YTDGrossPay
0 PreTax_Child_Care
0 LifeInsurAmount

0 FedWithHoldAllow
-408 AdjustedWage
0 SingleWithHold
0 MarriedWithHold
0 FedWithHold
0 NewYTDGrossPay

0 GrossOver87K
0 SocSec
0 Medicare
0 LifeInsurPremium
390 HealthInsurPremium
18 DentalInsurPremium
-408 AdjustedGrossPay

408 EmployeeInsurCost
300 EmployerInsurContrib
108 NetInsurCost
0 EmployeeTaxes
-408 NetPay

Figure 12. The Payroll spreadsheet.

instructor, which we ported into an equivalent Forms/3 spreadsheet. Payroll was a spreadsheet designed by two Forms/3 researchers using a payroll description from a real company.

These spreadsheets were each seeded with five faults created by real end users. To obtain these faults, we provided three end users with the following: (1) a "template" spreadsheet for each task with cells and cell names, but no cell formulas; and (2) a description of how each spreadsheet should work, which included sample values and correct results for some cells. Each person was given as much time as he or she needed to design the spreadsheet using the template and the description.

From the collection of faults left in these end users' final spreadsheets, we chose five that provided coverage of the categories in Panko's classification system [Panko 1998] (based upon Allwood's classification system [Allwood 1984]). Under Panko's system, mechanical faults include simple typographical errors or wrong cell references. Logical faults are mistakes in reasoning and are more difficult to detect and correct than mechanical faults. An omission fault is information that has never been entered into a cell formula, and is the most difficult to detect [Panko 1998]. We seeded Gradebook with three of the users' mechanical faults, one logical fault, and one omission fault, and Payroll with two mechanical faults, two logical faults, and one omission fault. Payroll was intended to be the more difficult task due to its larger size, greater length of dataflow chains, intertwined dataflow relationships, and more difficult faults.

The participants were provided these Gradebook and Payroll spreadsheets and descriptions, with time limits of 22 and 35 minutes, respectively. There are two reasons for the time limits. One is related to external validity: computing tasks in the real world are governed by time constraints, and time limits provide some simulation of this fact. The other is related to internal validity: removing the time limits would have introduced possibilities that participants would spend so much time on the first task they would be unwilling to spend time on the second, would leave as soon as the participation fee was collectible, and so on, which would introduce confounds into the data. The use of two spreadsheets reduced the chances of the results being due to any one spreadsheet's particular characteristics. The experiment was counterbalanced with respect to task order so as to distribute learning effects evenly. The participants were instructed, "Test the … spreadsheet to see if it works correctly and correct any errors you find."

3.2.5 Measures

The (observed) independent variable in this study was gender. The dependent measures were self-efficacy as measured by a Likert-scale pre-session questionnaire, overall percent testedness, seeded bugs fixed, new bugs introduced, time to first use of features, feature usage, score on a post-session questionnaire's comprehension questions, and opinions given on a Likert-scale post-session questionnaire. Details of these measures are provided with the results to which they apply.

3.3 RQ1 Results: Gender Differences in Self-Efficacy and Effective Debugging

3.3.1 Gender Differences in Self-Efficacy

As discussed earlier, gender differences in computer self-efficacy have been found in several computing situations. Our analysis of the pre-session self-efficacy questionnaire revealed that these differences were also present for debugging: females had significantly lower self-efficacy than the males (Mann Whitney: U=181, tied p<0.018). See Figure 13 and Table 3-2. Cronbach's alpha for the ten-item questionnaire was .879 on 49 cases, indicating high reliability. Self-efficacy literature suggests that high self-efficacy is critical for problem-solving [Bandura 1977, Bandura 1986], which predicts that for our results, high self-efficacy will be tied with high debugging effectiveness, a point we will return to shortly.

The self-efficacy literature further suggests that previous experience is one of the factors determining self-efficacy. To consider whether this held in our domain, we examined the participants' previous spreadsheet experience as a predictor of self-

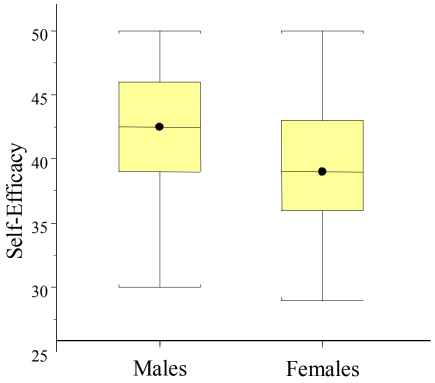


Figure 13. Self-Efficacy: Males' and females' pre-session self-efficacy. (Maximum possible self-efficacy was 50.) The center line of each box represents the median self-efficacy score. The boxes show the ranges encompassed by 50% of the scores of each gender. The whiskers extending above and below the boxes show the remaining upper and lower 25% of the scores.

Table 3-2. Self-Efficacy and Percent Testedness: Mean (standard deviation) and number of participants[8] for males' and females' self-efficacy and final percent testedness.

Gender	Self-Efficacy	Final Percent Testedness
Males	42.27 (4.69) n=26	62.85 (21.36) n=27
Females	38.96 (5.11) n=23	54.79 (25.04) n=24

efficacy. The relationships for the whole group—and for females—were significant (linear regression: all: F(1,45)=8.721, R^2=0.162, p<0.005; males: F(1,23)=4.002, R^2=0.148, p<0.057; females: F(1,20)=5.751, R^2=0.223, p<0.026). This relationship raises the possibility that, at least for females, low self-efficacy may be addressable by finding ways to increase their experience level.

[8] Note that the number of participants does not sum to 51. Some participants did not complete the questionnaire. Incomplete questionnaires were also the reason for other sample sizes in this paper that do not sum to 51.

3.3.2 Ties to Effectiveness

We first considered the relationship between self-efficacy and effective usage of WYSIWYT debugging features. We chose final percent testedness (refer back to the progress bar in Figure 10) as our measure of effective usage for two reasons. First, percent testedness can only increase through strategically checking off input/output value combinations. (Recall, input values must be chosen that actually add testing coverage of formula expressions.) Second, final percent testedness in previous experiments has been significantly tied with success in debugging [Burnett et al. 2004].

As Figure 14 shows, females' self-efficacy was indeed a significant predictor of their final percent testedness. For the males, however, self-efficacy was not a predictor of their effective usage of the debugging features[9] (linear regression: females: F(1,22)=4.52, R^2=0.177, p<0.046; males: F(1,25)=0.365, R^2=0.015, p<0.551). From this we can conclude that self-efficacy had important implications for females' problem-solving choices.

These choices of how much to use WYSIWYT testing features mattered: as in our previous studies, effective usage of the testing features (as measured through percent testedness) was predictive of the number of bugs fixed. The results of linear regression analysis of percent testedness on the number of bugs fixed were significant over all participants and also were significant for each gender (linear regression: all: F(1,49)=21.701, R^2=0.307, p<0.0001; females: F(1,22)=6.818, R^2=0.237, p<0.016; males: F(1,25)=16.60, R^2=0.399, p<0.0004).

[9] Pajares, while researching gender differences in self-regulated learning, found that males often responded to self-efficacy questionnaires with a different "mind set" than the females, the boys were being more "self-congratulatory" (likely to express confidence in skills they may not posses), where as girls were more modest [Pajares 2002]. This kind of discrepancy might also affect the males in our study.

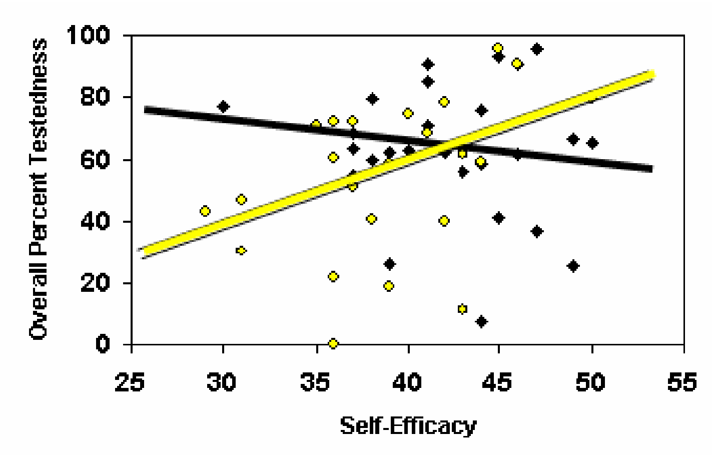


Figure 14. Self-Efficacy and Feature Effectiveness: Self-efficacy as a predictor of final spreadsheet testedness. The regression lines show the females' (yellow line) positive relationship of self-efficacy to spreadsheet testedness compared with the males' (black line) with no significant prediction between self-efficacy and spreadsheet testedness. The means are given in Table 3-2.

Finally, we considered the "bottom line" via two measures of debugging effectiveness: bugs fixed, and new bugs introduced. Recall that we had seeded each spreadsheet with five bugs. We count as "bugs fixed" those seeded bugs that were no longer present by the end of the task. "Bugs introduced" are bugs that were not seeded, but were present at the end of the task.

Although there was no significant difference between the females' and males' performance in fixing the seeded bugs (Mann Whitney: U=300.5, p<0.651), the females introduced significantly more bugs than the males did (Mann Whitney: U=227.5, p<0.011). See Table 3-3. The gender difference in bugs introduced is

Table 3-3. Fixed and Introduced Bugs: Mean (standard deviation) performance of males and females on bugs fixed and new bugs introduced that still remained at the end of the task.

Gender	Seeded Bugs Fixed (10 possible)	New Bugs Introduced
Males (n=27)	5.815 (2.167)	0.111 (0.424)
Females (n=24)	5.667 (2.014)	0.583 (0.974)

confirmed by a gender difference in participants introducing the bugs: 9 of the 24 females introduced bugs, which is significantly greater than the 2 males (out of 27 total) who introduced bugs (Fishers Exact Test: $p<0.015$). Note that these new bugs were never fixed.

3.4 RQ2 Results: Gender Differences in Acceptance of Unfamiliar Features

Was females' lower self-efficacy tied to lower acceptance of the debugging features that might have helped their effectiveness? As mentioned earlier, participants had access to three types of features: Type Familiar, Type Taught, and Type Untaught. We use these types to consider two forms of feature acceptance: willingness to initially approach a feature, and then willingness to adopt it (i.e., commit to repeated genuine usage during debugging).

3.4.1 Willingness to Approach New Features

Females were inclined to approach the Type Familiar feature earliest, using it significantly earlier than the males did (ANOVA: $F(1,49)=5.33$, $p<0.025$). In contrast to this, males approached the new features earlier than the females (Type Taught and Type Untaught): the gender difference was significant for Type Taught features and suggestive differences for Type Untaught features (ANOVA: Taught: $F(1,49)=8.694$, $p<0.005$; Untaught: $F(1,40)=3.40$, $p<0.073$; this statistic excludes the 3 males and 6 females who never placed an X-mark). Figure 15 shows the mean time of first usage for each of these feature types.

3.4.2 Willingness to Adopt New Features

Our criterion of adoption was repeated genuine usage. Measuring genuineness required somewhat different measures for each feature type. For the Type Familiar feature (formula edits), we simply used frequency of edits. This was a reasonable measure of genuine usage because editing a formula requires intellectual investment and pertains directly to debugging. However, for Type Taught features (checkmarks

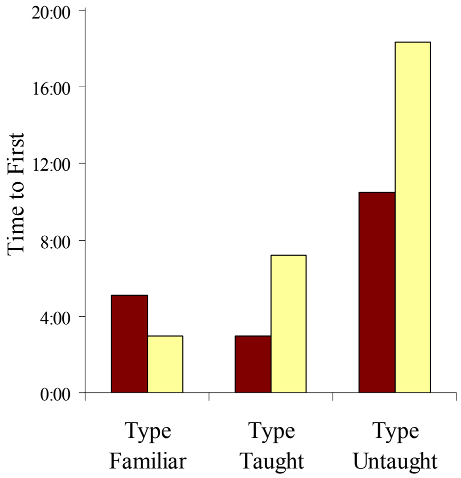


Figure 15. Mean Time to First: Males (dark bars) first used new (taught and untaught) features much earlier than females (light bars).

and arrows), the intellectual cost of usage was low, a single click. Furthermore, the effects on debugging are only indirect, because after a checkmark a formula edit was not necessarily expected (since placing a checkmark indicated belief that a cell's value was correct). For these features, it was not possible to determine presence of intellectual involvement, but there were patterns for which its absence could be inferred. We thus omitted Type Taught actions toggled again and again on the same cell by the participants after they had stopped editing formulas. After filtering these out, we then used frequency of the Type Taught actions as our measure.

For the Type Untaught feature (X-marks), intellectual cost was low, but there was a detectable route from genuine usage of the feature to debugging: following the advice of an X-mark's feedback leads eventually to formula edits on a colored cell. Thus, for Type Untaught features, a participant was counted as adopting X-marks if he or she placed more than one X-mark in at least one task, and then eventually followed up by editing a colored cell's formula. Since only about 60% of the participants exhibited this behavior and their frequency of usage according to this definition was necessarily low (1 or 2 was typical), counting participants rather than frequency was the right measure for Type Untaught feature adoption.

Table 3-4. Type Familiar: Mean (standard deviation) number of Type Familiar features.

Gender	Type Familiar Features
Males (n=27)	23.8 (9.58)
Females (n=24)	29.8 (9.66)

Table 3-5. Type Taught: Mean (standard deviation) number of actions associated with Type Taught features.

Gender	Type Taught Features
Males (n=27)	123.41 (68.27)
Females (n=24)	87.54 (47.67)

Table 3-6. Type Untaught: Number of participants who adopted Type Untaught features.

Gender	Adopted	Did Not Adopt
Males (n=27)	22	5
Females (n=24)	11	13

By these measures, the only type of feature for which females had a higher adoption rate was the Type Familiar feature of formula editing (ANOVA: F(1,49)=4.979, p<0.03). See Table 3-4. Males, however, were more willing to adopt the new features: they performed significantly more Type Taught actions than females, as Table 3-5 shows (ANOVA: F(1,49)=4.971, p<0.03). Furthermore, significantly more males used Type Untaught features than females did, as Table 3-6 shows (Fisher's Exact Test: p<0.01).

The gender difference in adoption of the Type Untaught feature may be partially explained by the answers (on a five-point Likert scale) to a statement included on the post-task questionnaire. The statement said: "... I was afraid I would take too long to learn [X-marks]." Females agreed with this statement significantly more than the males (Mann Whitney: U=157, p<0.017; not all participants answered this question therefore this statistic is reported on 22 females and 24 males).

However, despite the gender differences in expectation of their ability to learn the Type Untaught feature, there were no gender differences in actual learning of the feature—even though the males were able to practice it more through their greater adoption of it. In the post-task questionnaire, participants answered nine prediction and interpretation questions related to the Type Untaught feature. Males answered 60% of these questions correctly, and females answered 53% correctly (ANOVA: F(1,49)=0.929, p<0.34). This seems to be a case of *inappropriately* low self-efficacy of the females inhibiting their use of this feature.

3.5 Discussion

The results of this study establish ties from the well known gender differences in computer-related self-efficacy 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.

Females' low self-efficacy may be related to perceptions of risk, exacerbating the problem. Studies have documented females' high perception of risk in intellectual activities involving mathematical or spatial reasoning skills [Byrnes et al. 1999]. Applying this to our study, an individual with low beliefs in her ability to succeed at debugging may hesitate to use new debugging features because of the risk they may not pay off in better debugging performance. Further, she may believe that her cost of learning them will be high, due to her low opinion of her own capabilities. As predicted by the Attention Investment Model [Blackwell 2002] and borne out by the females' questionnaire responses and actions performed in our study, she may decide to forego the new features and use the debugging feature she already knows, formula editing.

Were the females' low self-efficacy predictions a case of realism, or of self-fulfilling prophecy? Females' perceptions of their inability to learn new features were not borne out by their actual learning of these features. This evidence suggests that females' low self-efficacy was 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.

In the present study, females spent the time they "gained" through foregoing the new features by editing more formulas. This resulted in significantly more introduced bugs, perhaps because, without the new features, they had less ammunition to use in tracking down these bugs or even realize they had introduced bugs. As several previous studies have shown, users do benefit in effectiveness from the debugging features [Burnett et al. 2004]. However, the data presented in this chapter indicate that the degree of benefit is not equal for females and males. This is a troubling result.

Our data also indicate that previous experience with spreadsheets has an important influence on self-efficacy. According to Bandura [Bandura 1977, Bandura 1986], the most important way of increasing self-efficacy is direct performance experiences. 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.

The relationship between experience and willingness to use new features suggests that a good design strategy may be to focus on how to initially attract females to try the features, thereby increasing their experience level. There are prior research results showing that the Surprise-Explain-Reward strategy effectively draws many users to new features [Robertson et al. 2004, Wilson et al. 2003], and this strategy provides a possible base for attracting females to relevant new features. However, in the underlying data there are indications of gender differences in some of the interruption-based Surprise-Explain-Reward devices. Our findings in the current study lend support to these indications. Further research into interactions between gender and interruption style in the domain of complex problem-solving tasks such as debugging may provide useful keys to how best to attract females to trying new features.

Females' perception that learning the new features would take them too long 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.

3.6 Chapter Summary

The main results of how software interacted with gender differences were:

- Females had lower self-efficacy 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 were to accept the new debugging features. One 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 is probably explained by low acceptance of the debugging features: high effective usage was a significant predictor of ability to fix bugs.

We believe these findings have implications far beyond debugging. They suggest to designers of software products for end users that, unless appropriate accommodations can be made, there are likely to be important gender differences in the users' willingness to accept new features that can benefit them.

4. Tinkering[10]

A few male and female participants in the think-aloud experiment (Chapter 3) had interesting differences in the ways they perceived features. For example, female F3, in using the new guards feature, said:

> F3: "I don't think that you can get a -5 on the homework. No, it can't be. So 0 to 100 [is the guard I'm entering], ok. Ok, hmm… So, it doesn't like the -5 [...]. They can get a 0, which gets rid of the angry red circle." [The red circle was a feedback device to call her attention to a value in violation of her guard.]

In contrast to F3's focus on the guard feature as a way to get her spreadsheet to work correctly, the following male's initial focus was on the feature itself:

> M4: "The first thing I'm going to do is go through and check the guards for everything, just to make sure none of the entered values are above or below any of the ranges specified. So, homework 1—actually, I'm going to put guards on everything because I feel like it. I don't even know if this is really necessary, but it's fun."

Despite his initial interest in the feature for the fun of it, the male soon transitioned to its problem-solving advantages to the task at hand:

> M4 (continuing): "...It looks like the guard on the sum of the first two homeworks is wrong, isn't it? Is this even necessary, should I even be doing this? Alright, what are you doing now?" [A red circle appeared because his guard did not agree with the computer generated guard.] "Ok, so it doesn't like my guard apparently. Ok, ah ha! The reason I couldn't get the guard for the sum to be correct is because the sum formula is wrong."

In fact, the above gender differences in views toward the same feature are consistent with reports of gender differences regarding motivation for using technology, for majoring in computer science, and how children talk about the use of technology [Brunner et al. 1998, Hou et al. 2006, Margolis et al. 1999]. In particular, the male participant's use of the guards "because I feel like it" is similar to oft-reported reasons males give for majoring in computer science: technology for the fun of it.

[10] The contents of this chapter are based on [Beckwith et al. 2005b] [Beckwith et al. 2006].

Males' greater engagement with features may be because they are trying out features for fun and consequently use the features for ultimately more effective problem-solving. Females, though, by not trying out the features for fun may not be gaining the benefits males are. If females spend less time exploring features they are less likely to acquire the same familiarity, understanding, and experience with the features compared to the males.

4.1 Qualitative Analysis of Self-Efficacy Data

These ideas led us to add a qualitative analysis to the data we had previously analyzed quantitatively in the features experiment (Chapter 3). The goal of this in-depth qualitative investigation of the participants' behaviors was to provide further insights into gender differences surrounding their feature usage. For example, were there behavior differences between the males and females that would lead to greater understanding of females' less engaged use of the debugging features?

For our qualitative analysis we (1) chose a subset of our original participants, (2) coded all of their actions taken during the study filling in attribute information, such as if their actions were correct or mistakes (using correctness of output values as our oracle), and (3) looked for patterns in the data with respect to the research goals.

4.1.1 Choosing the Participant Subset

Coding and analyzing data one-by-one for each of the original 51 participants would have required a huge amount of time, and did not seem likely to add valuable information beyond what we could learn from a subset of the original participants. Unlike the random selection of participants for quantitative research, when selecting participants for qualitative research, researchers often select participants with the greatest differences in specific areas of interest [Maykut and Morehouse 1994]. We therefore selected our participants based on two main characteristics: (1) checkmark and arrow usage and (2) X-mark usage. Since we were most interested in participants with extreme usage patterns in checkmarks, arrows, and X-marks, we selected participants with high and low usage in these areas, without knowledge of the gender of the selected participants. The genders of these chosen participants were then checked by another researcher not involved with applying the codings, to ensure a reasonable distribution by gender. We believed it was important that the two raters not know the gender of the participants in order to avoid bias in applying the codings and doing the early analysis of the data.

In the original statistical study that produced these data, there were 51 participants. Through the method described above we selected 22 participants. Both raters coded all the transcripts (logs of users' actions) from the 22 participants, which took approximately 160 hours in total.

4.1.2 Codes

The coding was a way of assigning each action, or set of actions, to categories that could later be used to answer questions about participants' behaviors. Developing the categories necessitated refining the research goals which we then used to determine the codes that would best allow us to address those goals. We decided to focus on how the features were used in relation to the task of finding and fixing bugs, and how features were explored.

The codes are given in Table 4-1. Two researchers applied the codes to the transcripts. For example, when a participant had placed a checkmark the raters would code whether that checkmark was correctly placed given the current value in the cell. Since the codes mainly pertained to relatively overt actions and system state, there was little disagreement. Coding discrepancies (accounting for less than 5% of the total actions coded) were then discussed. Generally coding disagreements were due to a mistake of one of raters. After discussion, of 13 transcripts where coded answers had been discussed and rated, the agreement rate was 5 disagreements out of 1234 coded actions on codes that did not include tooltips, and with tooltips included the disagreement rate was 18 out of 1437 coded actions.

Table 4-1. Codes Applied: The codes applied to each line of the participants' usage logs. Italics in the column "Details of Codes" are the specifics of what researchers indicated about participants' actions.

User Action	Details of Codes
Edit Formula	Indicate if change was: *Introduced, Fixed, Fixed Introduced, Fix Seeded,* or *Attempted Fix Seeded* Also mark if cell had *interior color* (due to previous placement of X-mark)
Value Edit	*Per Description* (value provided on handout) or *New Test Case* (not in description, and therefore their own test case)
Checkmark	Indicate if mark was: *Placed* or *Removed* Was value currently in cell correct? *Correct* or *Wrong* (Indicates if the user incorrectly marked the value of the cell as being correct.) Past status of checkbox (*?, blank, X*)
X-mark	Indicate if mark was: *Placed* or *Removed* Was value currently in cell wrong? *Correct* or *Wrong* (Indicates if the user incorrectly marked the value of the cell as being wrong.) Past status of checkbox (*?, blank, X*)
Arrows	Arrows turned *on/off* (repeating what transcript recorded, but included for completeness)
Tooltip	*Mouse resting* (would bring up many tooltips in the same second – unlikely user had time to read them) *Maybe reading* *X-placed reading x-mark tooltip* If an X-mark had just been placed, and they read about an interior we coded this action (since this was how they learned about X-marks – they were the untaught feature)

4.1.3 Results of Qualitative Analysis

Although some of the results we uncovered using this method could have been revealed using statistical methods, statistical analyses require knowing the questions to ask in order to run statistical tests. By instead systematically examining the data qualitatively, it is possible to detect unforeseen patterns that can lead to new research questions to examine statistically in later studies.

The results we highlight here were in two main areas:

1. How the features (specifically checkmarks and X-marks) were used in relation to the task of finding and fixing bugs.
2. How features were explored.

For the first result, we examined when checkmarks and X-marks were used in relation to editing formulas (presumably attempts to fix bugs). Our analysis revealed several patterns of testing in relation to finding and fixing bugs. These patterns also resembled patterns found in previous research [Krishna 2002]. These three main types of testing are incremental testing (where testing is combined with formula edits, such that a test is frequently made after a formula edit), batch testing (testing occurs all at once rather than consistently after editing a formula), and for completeness, little or no testing. In previous research, incremental testing was termed W-type testing, and batch was termed V-type testing. W-type testing was defined as a participant mixing formula modifications with testing. In comparison, in V-type testing, participants did all modifications before testing [Krishna 2002].

We coded whether or not the checkmarks and X-marks that participants' placed were correctly or incorrectly placed given the cell's value at the time of the testing decision. For participants engaged in incremental testing we separated the participants who tested immediately and primarily made correct testing decisions, from participants who tested immediately after formula edits and for whom testing decisions were incorrect (i.e. marking a cell's value correct when the value was wrong). Past research from our research group has shown that often users will make incorrect testing decisions, but that sometimes this is beneficial in terms of the feedback the system provides [Ruthruff et al. 2004]. However, they did not examine this in terms of specifically incremental testing.

Each of the participants' two tasks was analyzed separately since participants often changed their testing style from one task to another. Table 4-2 presents this data for each participant's tasks in terms of the type of testing they engaged in, and whether they were above or below the median number of bugs fixed.

Table 4-2. Testing Categories: Three main testing categories, each participant (males represented by the symbol ♂, females ♀) is represented for each task, and their testing strategy for that particular task.

		Bugs Fixed >= median	**Bugs Fixed < median**
Test after formula edit	Majority correct tests	♂♂♂♂♂♂♂♂♂♂♀♀	♂♀♀
	Majority wrong tests	♂♂♀	♂♂♂♀♀
Batch Testing		♂♂♂♂♂♀♀♀	♂♂♀♀♀
Very little / No Testing		♀	♂♀♀♀♀♀

There were two main gender differences within the categories. Males were the dominant gender engaged with incremental testing, and in particular with correct tests in incremental testing. Females, on the other hand, were the dominant gender in the very little to no testing category. The other categories were more evenly split.

The type of testing an individual engages in has an impact on the feedback they receive, and some types of testing are also related to more successful outcomes of finding and fixing spreadsheet errors, according to a number of past studies. These characteristics may have implications on design and presentation of the features. The following are some of the characteristics:

1. Incremental testing leads to immediate visual feedback in terms of a cell's and spreadsheet's testing progress, which could play a role in encouraging more of the same behavior.

2. In general, correctly testing a cell's value after a formula edit was tied to success in finding and fixing the spreadsheet errors.

3. Batch testers may view testing and fixing bugs as separate activities, whereas incremental testers may view the task as one and the same.

4. Not using the testing features was almost a guarantee of not succeeding at the task.

Our initial analysis into the question of how features were used in relation to finding and fixing bugs indicated gender differences in the type of testing in which males and females had a tendency to engage. A majority of the males engaged in incremental testing, while many more females did very little or no testing. (Note that this finding provides suggestive evidence for Hypothesis R-2: "Gender differences in risk perception may impact the strategies by which males and females engage in end-user programming environments.")

In the second result area we investigated how males and females explored the debugging features. We specifically focused on the X-mark feature since this feature had not been taught; therefore, participants had not been taught a specific usage strategy.

While observing specific instances of uses with the X-mark feature, a clear pattern emerged: some participants would place an X-mark and immediately remove it before taking any other action. Occasionally this corrected a slip, but usually there was no obvious goal-oriented explanation for placing and removing the X-mark. Having discovered this seemingly non-goal-oriented behavior, we further observed that the participants who behaved in this manner were consistently males. A quantitative analysis of these data (including all participants and not only those we investigated qualitatively) confirmed that males did significantly more X-mark tinkering than females (t=-2.2, df=49, p<0.035).

This led us to develop new research questions pertaining to the role of playful experimentation in end-user software engineering.

4.2 Playful Experimentation

Research over two decades indicates that a playful approach to learning increases motivation to learn and the corresponding ability to perform tasks effectively [Martocchio and Webster 1992, Webster and Martocchio 1993]. Similarly, learning can be enhanced through arousing curiosity, by providing change, complexity, or attention-attracting features that motivate exploration of an environment [Lepper and

Malone 1987, Malone and Lepper 1987, Wilson et al. 2003]. Tinkering and curiosity are related because tinkering, as an informal, unguided exploration of features visible in the environment, is one way to satisfy one's curiosity.

Curiosity-based exploration is a familiar phenomenon in science and technology education. Research emphasizes the potential value of open-ended exploration in learning [Rowe 1978]. Other educational research has identified gender differences in exploratory behaviors. Among primary school students, studies in mathematics, geography, and gaming indicate that boys tend to tinker and to use tools in exploratory, innovative ways. Girls are less likely to tinker, preferring to follow instructions step-by-step [Jones et al. 2000, Martinson 2005, Van Den Heuvel-Panheizen 1999]. Similar tinkering findings are also true of males majoring in computer science [Margolis and Fisher 2003, Tillberg and Cohoon 2005], but not of the female computer science majors. The consistency of these reports led us to believe that the propensity to tinker might play an important role in end-user debugging effectiveness.

Given this evidence of tinkering behaviors in our previous study's data, along with that study's results tying low self-efficacy in females to their lack of acceptance of important debugging features, we began to wonder whether there is a tie between tinkering and self-efficacy in the sort of problem-solving software environment used by end-user programmers.

The combined evidence on both tinkering and self-efficacy in our own research and in the literature led to an empirical study investigating the effect of tinkering and its relationship to gender, self-efficacy and debugging effectiveness. This study is presented in the following sections.

4.3 Experiment

We designed our experiment to consider the effects of two treatments, Low-Cost and High-Support, on males' and females' tinkering, self-efficacy, and debugging in a spreadsheet environment. In the Low-Cost treatment, tinkering was easy to do, since the cost in terms of user action was low. The High-Support treatment was designed to provide greater support for the debugging features, but had the side effect of increased tinkering cost. (Chapter 5 covers the specific design details of the more supportive environment.)

The underlying spreadsheet environment for the two treatments was the same and was presented in Chapter 3.

4.3.1 Environment: Two Treatments

The Low-Cost and High-Support treatments varied in three ways: WYSIWYT input devices, explanation content and interaction, and the number of task-supporting features available within the environment. This environment was exactly the same as presented in Chapter 3, Section 3.2.2.

4.3.2 Low-Cost

In the Low-Cost treatment, WYSIWYT interaction required one click for placing or removing a testing decision (left-click for checkmark, right-click for X-mark). Explanations (same as presented in Chapter 3), provided through tool tips, were as short as possible, to keep their reading cost low [Wilson et al. 2003].

4.3.3 High-Support

The aim of the High-Support treatment was two-fold, first to encourage low self-efficacy users to take advantage of the features that can help users debug, and second to provide an environment in which learning was supported through fuller explanation content. A side effect of the additions was a higher user-action cost, requiring more clicks, reading, and choice of features to use. This higher cost applied to tinkering as well as to other actions.

In the High-Support treatment, along with the checkmark meaning that a value is correct and X-mark meaning that it is incorrect, the users also could make decisions for values that “seem right maybe,” or “seem wrong maybe.” The purpose of this mechanism was to encourage low self-efficacy users by reassuring them that confident decisions were not a prerequisite in using the devices. The colors reflecting these more tentative “seems” decisions were the same hues but less saturated than those of the other decisions. The system’s inferences about which cells were tested or faulty were the same as for the Low-Cost treatment, but the system also propagated the amount of tentativeness, allowing the user to discern which statuses were based on the “seems” decisions. The input device required a user first to click on the “?” in the decision box, which brings up the four choices shown in Figure 16, and then to click on their choice. (In contrast, recall that placing a checkmark or X-mark in the Low-Cost treatment required only one click.)

In addition, the explanations were expanded to support users who wanted more guidance than the explanations given in the Low-Cost treatment. The mechanism was as follows. In addition to the tool tip content of the Low-Cost treatment, additional information was available via a “Tips” expander (Table 4-3), which could be expanded and dismissed on user demand. The expanded “Tips” included further information on why the object was in its current state and possible actions to take next. Once expanded, the tip would stay visible until the user dismissed it, supporting non-linear problem solving and requiring less memorization by the user.

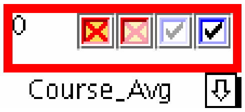
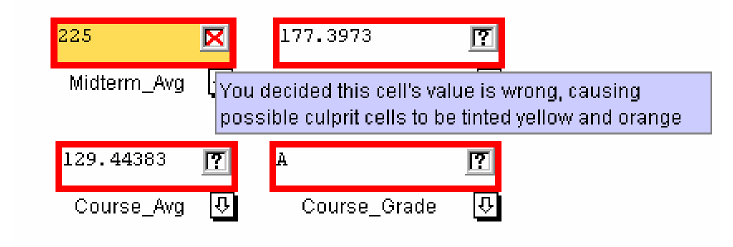
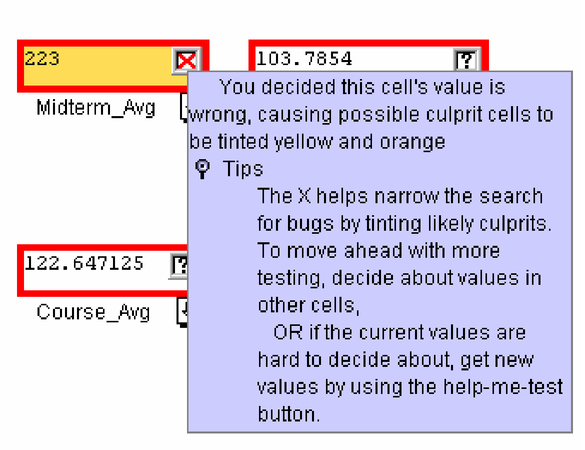


Figure 16. 4-Tuple Testing Choices: Clicking on the decision box turns it into the four choices. Each choice has a tool tip, starting with the left-most X these are “it’s wrong,” “seems wrong maybe,” “seems right maybe,” “it’s right.”

Table 4-3. Explanation Examples in Tinkering Experiment: Participants were assigned to one of two environments.

Environment	Example
Low-Cost Environment: This environment was expected to be more encouraging of tinkering.	 225 177.3973 Midterm_Avg 129.44383 A Course_Avg Course_Grade You decided this cell's value is wrong, causing possible culprit cells to be tinted yellow and orange
High-Support Environment: included additional feature explanations and a "help me test" scaffolding feature. These support features were expected to bolster low self-efficacy participants, but the additional richness of the features also added to the cost and complexity, requiring extra clicks to access the additional support features and producing more feedback for the users to interpret.	 You decided this cell's value is wrong, causing possible culprit cells to be tinted yellow and orange Tips The X helps narrow the search for bugs by tinting likely culprits. To move ahead with more testing, decide about values in other cells, OR if the current values are hard to decide about, get new values by using the help-me-test button.

The "Help Me Test" feature [Fisher II et al. 2002, Wilson et al. 2003] was provided to the High-Support group (but not to the Low-Cost group) to help users overcome difficulties in finding additional test cases. Sometimes it can be difficult to find test values that will cover the untested logic in a collection of related formulas, and Help Me Test tries to find inputs that will lead to coverage of untested logic in the spreadsheet, about which users can then make testing decisions. Help Me Test is not fully automated testing but rather scaffolding: it provides new test inputs, but does not make decisions about the outputs that result, so does not actually "test" the spreadsheet.

The differences between the two environments are summarized in Table 4-3.

4.4 Procedures

The participants were randomly divided into two groups: a group of 37 participants (20 males and 17 females) received the Low-Cost treatment, and a group of 39 participants (16 males and 23 females) received the High-Support treatment. We recruited participants from the university and community; we required all participants to have some spreadsheet experience, and also limited the amount of programming experience they could have to a very small amount. Statistical tests on questionnaire data showed no significant differences between the groups in grade point average, spreadsheet experience, or programming experience.

The same pre-experiment questionnaire collected participant background and self-efficacy data. We administered a 35-minute "hands-on" tutorial to familiarize participants with their treatment. The participants were then given two tasks. We captured their actions in electronic transcripts, as well as their final spreadsheets.

Following the tutorial participants had to test two spreadsheets, Gradebook and Payroll, as in the study reported in Chapter 3. The experiment was counterbalanced with respect to task order in order to distribute learning effects evenly. The participants were instructed, "Test the … spreadsheet to see if it works correctly and correct any errors you find."

At the conclusion of each task, we administered questionnaires that included questions regarding how users perceived their performance on that task. The final questionnaire included a follow-up post-self-efficacy questionnaire identical to the pre-self-efficacy questionnaire, as well as questions assessing participants' comprehension of the X-mark feature and their attitudes toward the features they had used. Taking two measures of self-efficacy (one prior to the experiment and another following the final task) is valuable information, because according to self-efficacy theory (as reported in Chapter 2), people working in a new and unfamiliar environment have malleable self-efficacy much of which is based on their first experiences, and in particular early perceived failures can have especially pronounced effects on self-efficacy.

4.4.1 Tutorial

In the tutorial, participants performed actions on their own machines with guidance at each step. Although the Low-Cost and the High-Support tutorials both described the checkmark feature (including its associated testedness-colored arrows feature), neither tutorial included any debugging or testing strategy instruction. Furthermore, neither tutorial explained the X-mark feature beyond showing that it was possible to place X-marks (with time to explore any aspects of the feedback – through explanations – that they found interesting). At the end of the tutorial, we gave both groups time to explore the features they had just learned by working on the tutorial spreadsheet debugging task.

The High-Support tutorial explained the additional features of the treatment, allowing ample time to explore the choices in check and X-marks (Figure 16), the Help Me Test feature, and the expanded tool tips. To compensate for the extra time it took to explain the additional features in the High-Support treatment, the Low-Cost group had several extra minutes at the end of the tutorial to explore and/or work further with tutorial spreadsheet debugging task.

As in the features experiment in Chapter 3, half of the tutorial sessions were presented by a male and half by a female, balanced so that 50% of participants were instructed by a same-gender instructor and 50% by the opposite gender.

4.5 Results

We have already pointed out that ties have been found between tinkering and educational goals, and within that context tinkering seems to be a male characteristic. However, in the domain of end-user debugging, the goal is not education per se, but rather productivity or effectiveness in fixing the bugs. Still, educating oneself about features that seem useful to the task could be a necessary subgoal.

Thus, we consider whether there were gender differences in tinkering as a way to master new features, and how such differences might tie to debugging effectiveness.

4.5.1 Tinkering by Gender: How Much

Our measures were tinkering frequency, tinkering episodes, and tinkering rate within episodes (a measure of commitment to tinkering within an episode). We operationally define a tinkering instance as turning a feature "on" immediately followed by turning the feature "off," such as placing a checkmark or turning on an arrow and then removing it as the next action. Although a tinkering instance is simple to perform in this environment, the complex feedback users receive when tinkering is where the constructive experience begins to occur: through users' tinkering actions they can construct concrete, visual paths backwards ("breadcrumbs" of where they've been) and forwards (where they need to go to achieve 100% testedness). Tinkering frequency is simply a count of the number of tinkering instances. A tinkering episode is defined to be a sequence of one or more tinkering instances, terminated by a cell edit or the end of the task. The episode count for each participant serves as a measure of consistent use of tinkering. Finally, tinkering rate, computed as tinkering frequency per episode, measures "vestedness" in tinkering within an episode: once a participant starts to tinker, how committed does he or she stay to tinkering before moving on (indicated by editing a cell)?

Our expectations, given previous literature, were that males would make greater use of tinkering than females by all of these measures, regardless of treatment.

Our expectations were wrong. The analysis of the tinkering frequency measure (illustrated in Figure 17) revealed that the Low-Cost males stood apart from the others. A 2 (gender) by 2 (treatment) ANOVA revealed a significant main effect of treatment (F[1,72]=7.15, $p<0.01$) and a significant interaction effect (F[1,72]=4.42, $p<0.05$), although the main effect of gender alone fell short of significance at the .05 level ($p<0.10$, F[1,72]=2.82). Thus, the treatment affected the genders differently. In particular, treatment made almost no difference for the amount of tinkering females

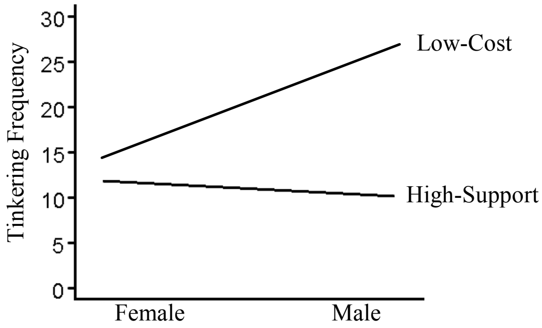


Figure 17. Tinkering Frequency Interaction: This interaction plot of 4 means depicts the gender x treatment interaction in tinkering frequency.

did. However, for males a follow-up analysis (using the Tukey method) revealed a significant effect of treatment (p<0.05) on their tinkering frequency.

Table 4-4 shows the means of each gender for all three of our tinkering measures (for completeness each of the measures is reported in this table, however, discussion of tinkering rate is in later sections). On the tinkering consistency measure (number of episodes) ANOVA analysis still showed a significant effect of treatment (F[1,72]=9.64, p<0.01), but did not approach significance for gender or for gender x treatment. Thus, as the episodes rows in Table 4-4 show, the High-Support group had less consistent emphasis on tinkering as a problem-solving device.

Table 4-4. Means of Tinkering Measures: Results for each measure are shown as a group at the right, with significant results (p<0.05) bold faced, marginally significant results (between 0.05 and 0.10) in regular font, and insignificant results in grey font.

	Low-Cost	**High-Support**	***p*-value**
Frequency			Gender: <0.10
Males	27.3	10.1	**Treatment: <0.05**
Females	13.7	11.7	**Interaction: <0.05**
Episodes			Gender: 0.769
Males	9.5	5.1	**Treatment: <0.05**
Females	7.9	6.0	Interaction: 0.230
Rate			Gender: 0.142
Males	2.6	1.8	Treatment: 0.251
Females	1.7	1.9	Interaction: <0.10

4.5.2 Discussion

Even though males and females had equal tinkering opportunities, these results show trends towards opposite effects of treatment on gender in the Low-Cost treatment. In particular, we found a surprisingly large effect of treatment on males' tinkering, which we will consider further in later sections. We now turn our attention to whether tinkering actually helped either gender in their debugging efforts.

4.5.3 Does Tinkering Matter to Effectiveness?

In educational settings exploration has been encouraged for improved performance [Rowe 1978]. Although the education setting is different from the domain of end-user debugging, our expectations were that high-tinkering males would be more effective than the others since their tinkering is higher than that of the other participants.

To investigate this possibility, we used the following dependent measures. The first was bugs fixed, because fixing bugs was an explicit goal assigned to the participants. The second was percent testedness of the spreadsheet (as seen at the top of Figure 10), since in previous experiments (include the features experiment reported in Chapter 3) this has been significantly tied with success in debugging [Burnett et al. 2004]. This relationship of percent testedness to bugs fixed was found again in this experiment for both genders (linear regression, males: F[1,34]=27.16, R^2=0.44, p<0.01; females: F[1,38]=10.51, R^2=0.22, p<0.01). The third was the participants' understanding of the debugging feature (X-mark) as measured in the post-session questionnaire.

A 2 (gender) by 2 (treatment) ANOVA showed no significant differences of the outcomes of these measures by gender, treatment, or gender x treatment, as Table 4-5 suggests. However, there were surprising gender differences in the ways tinkering predicted these results.

Table 4-5. Effectiveness Measures: Means of effectiveness measures.

	Low-Cost	**High-Support**
Bugs fixed		
Males	5.9	6.5
Females	6.3	5.3
% Testedness		
Males	61.9	63.2
Females	67.9	65.3
Understanding		
Males	6.8	7.7
Females	8.1	7.1

Figure 18 summarizes the differences in the way tinkering related to effectiveness for the males versus the females. As it shows, males' and females' tinkering affected their debugging effectiveness, but in essentially opposite ways. Females' tinkering was a significant predictor of their final percent testedness (linear regression, episodes: F[1,38]=4.63, R^2=0.11, $p<0.05$). Recall from above that final percent testedness was in turn highly predictive of bugs fixed. (That is, tinkering did not directly predict bugs fixed; rather testedness was a mediating factor for the relationship.)

For the males, however, no measure of tinkering was predictive of their percent testedness. Thus no mediated relationship to bugs fixed existed and, in fact, tinkering rate was found to be a *negative* predictor of bugs fixed (linear regression:

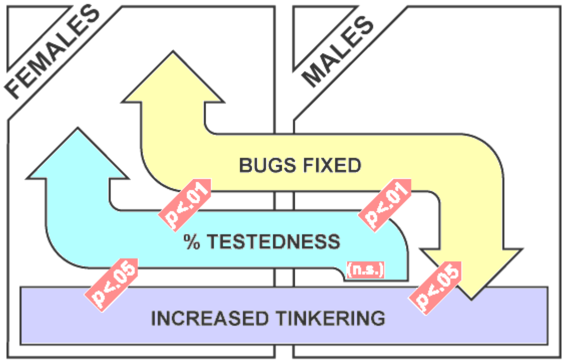


Figure 18. Effects of Tinkering: Left: females, Right: males. Direction of stylized arrows depicts increase/decrease in a measure, and shaded arrows show significance of the regression relationships between measures.

F[1,34]=8.04, R^2=0.19, $p<0.01$).

4.5.4 Discussion

Although there is previous research, including our own, showing that software environments are often designed in ways better aligned with males' need rather than females' needs [Huff 2002], in the realm of tinkering, this section's results point to a disadvantage for the males. In particular, in contrast to our expectations, tinkering was tied to negative outcomes for the males. The females, however, had positive outcomes tied with tinkering.

Tinkering for females showed similar benefits to understanding as it did for effectiveness: Their tinkering significantly predicted their understanding (linear regression: frequency: F[1,38]=4.56, R^2=0.11, $p<0.05$; episodes: F[1,38]=4.44, R^2=0.10, $p<0.05$). For the males no measure of tinkering was predictive of their understanding.

4.5.5 Tinkering and Self-Efficacy

As referred to earlier, in Chapter 3 we established pre-self-efficacy as an important factor in female end-users' debugging. The current study has confirmed some of the results we found in Chapter 3 (see Table 4-6). With respect to tinkering and pre-self-efficacy, we expected that increased tinkering would increase females' post-self-efficacy. We also expected the set of features in the High-Support treatment would increase females' post-self-efficacy; however, adding additional features carries the risk of reducing self-efficacy—even if the features provide more support—by making the environment more complex.

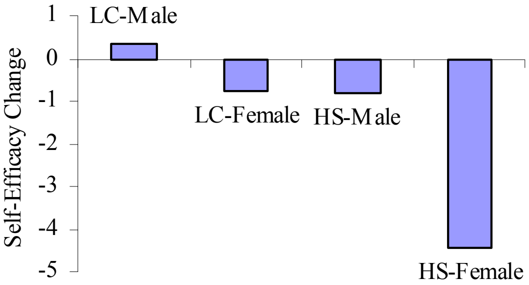


Figure 19. Self-Efficacy Change: The change from pre-self-efficacy to post-self-efficacy. The High-Support females' drop was significant.

In fact, we found a dramatic fall in post-self-efficacy for females using the High-Support treatment (Figure 19). Most groups had little to no difference in their self-efficacy change between pre-self-efficacy and post-self-efficacy. However, the High-Support females' self-efficacy dropped significantly over the course of the study (paired t-test: t=3.19, df=22, $p<0.01$).

The relationships between tinkering and post-self-efficacy were also surprising. For High-Support females, the rate of tinkering per episode was predictive of the *drop* in self-efficacy reported above (linear regression: F[1,21]=6.32, $R^2=0.23$, $p<0.05$). Also, we found suggestive evidence linking increased tinkering frequency to increased post-self-efficacy for the females in the Low-Cost condition (linear regression: F[1,15]=3.51, $R^2=0.19$, $p<0.10$). Males, however, did not exhibit any significant relationships between their tinkering and post-self-efficacy.

	p-value	**F [1,74]**	**R^2**
Previous experience → SE	**<0.05**	4.46	0.06
SE → % testedness	**<0.01**	7.10	0.09
SE → Understanding	**<0.01**	11.38	0.13

Table 4-6. Self-Efficacy Predictors: Self-efficacy (SE) results of current experiment replicated those from the features experiment of Chapter 3. Previous experience was predictive of pre-self-efficacy, and pre-self-efficacy was predictive of the two other factors.

4.5.6 Discussion

What conclusions can be drawn from the extreme drop in self-efficacy by High-Support females and the relationships between tinkering and post-self-efficacy? One possible conclusion is that the High-Support females did not perceive tinkering as helpful for understanding how their debugging environment worked. Therefore, the more they tinkered, the more they reinforced their perception of their inability to understand what was happening in the environment. This relates to a finding in the features experiment of Chapter 3, in which females believed more than males that it would take them too long to learn the X-mark feature—even though in actuality they understood it as well as the males.

Taken in combination with our current tinkering effectiveness results, it appears that the tinkering issue for females is complex. Females were better than the males at consistently extracting problem-solving benefits from tinkering, but were worse than the males at maintaining their self-efficacy levels.

4.6 Tinkering Considered Harmful?

Recall that male tinkering was highly dependent upon the treatment, with the males in the Low-Cost group being significantly higher tinkerers than the others. A closer look at the understanding scores among the Low-Cost males revealed that their understanding scores tended to be quite extreme: either nearly perfect or extremely low. To shed light on potential factors that may explain these results, we consider *types* of tinkering and also tinkering's relationship to *reflection*.

4.6.1 Tinkering: Exploratory and Repeated

First we consider two types of tinkering, which may help explain why males' tinkering was not effective, at least from a statistical standpoint, whereas females' tinkering was effective.

For example, one of the Low-Cost males was observed turning on a feature, then immediately turning off that feature, and then again turning the same feature on and off again, all actions occurring on the same cell. This participant's understanding score was very low (5.5 out of a possible 12) and he did not fix any of the 10 bugs, suggesting that this type of tinkering was not particularly useful to him.

To investigate whether different types of tinkering might have opposing effects—some increasing debugging effectiveness and some interfering with it—we partitioned the tinkering behaviors into two subsets: exploratory tinkering and repeated tinkering. *Repeated tinkering instances* are the number of tinkering instances in a sequence of two or more consecutive tinkering instances on the same feature and the same cell, as with the Low-Cost male above. For example, turning the arrows for a cell on and then immediately back off again three times in a row is three repeated tinkering instances. Hence, the repetitions can only repeat information already revealed by the previous tinkering instances. *Exploratory tinkering instances*, which we define as the difference between total tinkering instances and repeated tinkering instances, potentially can impart new information.

Oddly, the males' exploratory tinkering was *not* statistically predictive of understanding or of any of the effectiveness measures. In contrast to this, an increase in female exploratory tinkering (which accounted for 91% of their overall tinkering) was a significant predictor of increased understanding (linear regression: F[1,38]=4.61, R^2=0.11, $p<0.05$).

Interestingly, as Figure 20 shows, repeated tinkering instances accounted for a significantly greater proportion of the Low-Cost males' repeated tinkering than any other group, with a significant effect of interaction between gender and treatment (ANOVA: F[1,72]=5.82, $p<0.05$). In fact, nearly 17% of the Low-Cost males' tinkering instances were of this type, almost twice as many as the next highest group.

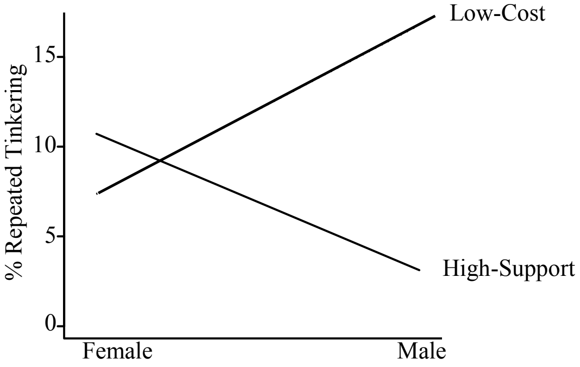


Figure 20. Repeated Tinkering Interaction: This interaction plot (of means only) illustrates the gender x treatment interaction in percentage of repeated tinkering.

Repeated tinkering, which was done predominantly by the Low-Cost males, had a significant negative relationship to understanding (linear regression: males: F[1,18]=6.0, R^2=0.25, p<0.05). See Figure 21.

4.6.2 Discussion:

The High-Support treatment, because of its increase in tinkering cost over the Low-Cost treatment, not only greatly reduced males' tinkering, it *selectively* reduced primarily the ineffective type of tinkering. Since it did not affect the females' tinkering behavior, from a tinkering effectiveness perspective, this approach appears to be the better of the two treatments for both genders, albeit for different reasons.

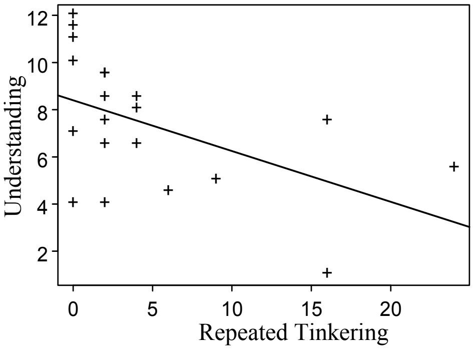


Figure 21. Repeated Tinkering and Understanding: The negative regression relationship between repeated tinkering and understanding for the Low-Cost males.

Still, the lack of relationship between males' exploratory tinkering and effectiveness suggests that the different types of tinkering do not alone explain how males' tinkering related to their debugging effectiveness.

4.7 Reflection

Research from the education field has shown that when students are given "wait-time" of three seconds or more after a classroom response, their critical thinking about that response improves [Rowe 1978]. To see whether such wait-times also apply to end-user debuggers, we likewise define pauses as three or more seconds of inactivity after a user action. Note that here we consider pauses after *any* action (not just tinkering instances), because every action provides feedback. If Rowe's research applies to our domain, then increased pauses between actions should result in increased understanding and effectiveness.

For our analysis, we counted the frequency of pauses. Overall, females had significantly more pauses than males, with a mean of 220 versus 190 occurrences (ANOVA: F[1,74]=4.22, R^2=0.05, p<0.05). Thus, males did not take the time that might have been used to reflect upon the feedback from their actions as often as the females did.

These pauses mattered. Participants who paused long enough to reflect on the system's responses understood the debugging devices better and used features more effectively than the other participants. Specifically, for both genders, more frequent pauses were predictive of greater effectiveness, as measured by bugs fixed (linear regression: all: F=[1,74]=5.36, R^2=0.07, p<0.05), final percent testedness (linear regression: all: F=[1,74]=31.3, R^2=0.30, p<0.01), and understanding of features (linear regression: all: F=[1,74]=14.11, R^2=0.16, p<0.01).

One question that arises is whether the longer the pause the better the effect. Our analysis suggests that this is not the case: for both genders, longer pauses (which were also tied with fewer tinkering instances; linear regression: F[1,74]=6.13, R^2=0.08, p<0.05) were not helpful. They were significantly predictive of a decrease in bugs fixed, percent testedness, and understanding (linear regression: bugs fixed: F[1,74]=11.85, R^2=0.14, p<0.01; percent testedness: F[1,74]=37.07, R^2=0.33, p<0.01; understanding: F[1,74]=11.81, R^2=0.14, p<0.01).

4.7.1 Discussion

These results help to shed light on the inverse relationship between tinkering and effectiveness for males as compared to the positive relationship of tinkering and effectiveness for females. Males' tendency to tinker more appears to be useful only when they make regular use of pauses: and in our study, their tendency to pause too little may have interfered with the benefits of tinkering.

4.8 Chapter Summary

In this chapter, we have demonstrated that some tinkering habits are counter-productive and that these are more often exhibited by males. If we can encourage both females and males to avoid such habits, while exploring end-user programming tools in a productive way, it should be possible to provide genuine benefits for *both* genders.

An assumption in much past work on gender differences in computing is that males' behaviors are reasonably well-matched to today's problem-solving features in environments such as for end-user programming. A further assumption is that these environments need to evolve to allow females' behaviors to achieve success at the same levels. Our results show, however, that males' behaviors may sometimes be the ones that are not as well-supported as the females' behaviors.

More specifically, our findings included:

- As in previous research, males tinkered more than females but, surprisingly, males' tinkering was often counterproductive to their effectiveness in debugging.

- One factor in the above result was the fact that the Low-Cost treatment led some males to engage in unproductive, repeated tinkering, which was linked to poor understanding.
- Although they tinkered less, females' tinkering was very effective: it was significantly tied to understanding and to successfully testing and debugging, regardless of treatment. However, when tinkering in the more complex environment, females' tinkering was predictive of lower self-efficacy.
- Tinkering with pauses allows for reflection and was helpful to everyone, but females were more likely than males to pause.

These results show that tinkering can be a valuable activity in end-user debugging, but the prescriptions on how an environment should be designed to guide male and female tinkering are different. Females should be encouraged to tinker because it helps them to be effective, with the important caveat that tinkering in a complex environment carries a risk of damaging the females' self-efficacy. In contrast to this, males' self-efficacy did not seem at risk, but our results suggest that males need to be guided to tinker less repeatedly and more "pausefully."

5. *Designing for Self-Efficacy*[11]

Chapter 3 uncovered several gender differences regarding the use of the environment features that negatively affected the females' performance. This chapter addresses the process and results of theory-based design changes centered on those gender differences. Table 5-1 enumerates the particular gender differences we set out in this chapter to address.

5.1 From Problem to Solution 1: "No Confidence Required"

From a high-level design perspective, we are dealing with an "ill-structured" [Simon 1973] 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.

For our ill-structured problem we drew from a combination of existing empirical results, theories, and human-computer interaction (HCI) design techniques to approach

Table 5-1. Results of Features Experiment: Summarized results from the features experiment in Chapter 3.

Result 1: Females had lower self-efficacy 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).
Result 2: 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.
Result 3: 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: high effective usage was a significant predictor of ability to fix bugs.

[11] The contents of this chapter are based on [Beckwith et al. 2005c].

design changes. Following Ko et al.'s example [2004], we use the concept of "barriers" to help organize the problem space. Table 5-2 lists barriers and potential solutions to help females overcome these barriers. We derived the barriers and potential solutions by going back to the theories (self-efficacy, attention investment, etc. as presented in Chapter 2) to help specifically state potential barriers and guide potential solutions.

5.1.1 Barriers and Potential Solutions

As we have already pointed out, Barrier 1, low confidence in females in computer-related tasks has been widely reported, as has risk aversion in females (Sections 2.1

Table 5-2: Barriers and Potential Solutions: Barriers females faced related to the findings of Chapter 3 and potential solutions, both informed by theories.

Barrier	**Potential Solutions**
Barrier 1: Low computer-related confidence in females.	Emphasize low risk nature of judgments by providing a way to make it acceptable to express less confident judgments. (For example: not very sure to very sure)
	Provide a "what if these cells were wrong" feature, where users can get feedback, but do not have to commit to saying that the cells are definitely wrong.
	Experience helps in increasing confidence.
Barrier 2: Low feature usage by females.	A WYSIWYT Skill Builder (similar to a Wizard, but set up to facilitate learning without being overly directive) to introduce users and lead them to greater skills.
Barrier 3: Perception that it will take too long to learn the X-mark feature.	Clearly state X-mark's usefulness, to emphasize the value of learning the X-mark.
	Watch someone else use X-marks.
	Enhance fault localization feedback to help users understand how fault localization narrows down the potentially faulty formulas.
	Expand content of explanations to help users make more accurate assessment of risks and benefits of using the X-mark feature.

and 2.3 in Chapter 2).

The attention investment model [Blackwell 2002], and its focus on users taking actions only if they believe that the action's benefits are greater than their perceived costs (also factoring in perceived risks), implies that our approach 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 (the first two potential solutions listed in Table 5-2). The third potential solution for Barrier 1, increasing experience to help increase confidence, is based on Bandura's self-efficacy theory [Bandura 1977]. Bandura argues that the best way to increase self-efficacy is to give the low-confidence individual more experience in personally accomplishing the task.

Barrier 2, low feature usage by females, is not independent of the other barriers, but is present in our table because it encourages thinking directly about usage, rather than concentrating only on underlying causes, as in the other barriers. A proposed solution is to provide a "wizard-like" entity, such as Excel's Chart wizard, to facilitate feature usage and to build skills. This approach draws from minimalist learning [Carroll 1998, Rosson et al. 1990], which advises that new system features should be introduced by engaging users in activity and providing scaffolding to help them gradually increase their skills. As this learning theory advocates, the scaffolding would avoid being so overly directive that users blindly follow instructions; thus the device would be somewhat different from traditional wizards, which tend to be very directive.

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. The second solution, drawn from self-efficacy theory [Bandura 1977], indicates that observing peers accomplishing the task is an important source of self-efficacy. This would be realized by a low self-efficacy female observing another female peer.

It is also possible that the feedback about the results of X-marks led to Barrier 3. If so, then enhancing the feedback would help reduce the barrier. From a theoretical perspective, Norman's action cycle [Norman 1988] points out that to carry out a task successfully, users must correctly interpret feedback on their actions. Arroyo [2003] and 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.

5.1.2 Claims Analyses

For each solution in Table 5-2 we performed a claims analysis. Claims analysis [Carroll and Rosson 1992] is a technique for evaluating design solutions. In claims analysis the researchers identify positive and negative consequences of each solution with respect to the intended users. Our claims analyses were instrumental in helping us to improve our solutions and to choose which solutions to implement. For example, the claims analysis for the first solution in the table (which became our "Solution 1") is shown in Table 5-3. (Aspects of other potential solutions were adapted into design changes, these will be highlighted later in the chapter.)

Table 5-3. Claims Analysis: The claims analysis for Barrier 1 of Table 5-2.

Problem (re: Barrier 1): Females might use checkmarks or X-marks only when they are confident about their judgments.
Potential Solution: Emphasize low risk nature of judgments by providing a way to make it acceptable to express less confident judgments.
Pros: + may increase willingness to use checkmarks or X-marks. + user receives feedback that encourages placing a mark at the moment he/she questions a cell. + optional—user not forced to use it—yet noticeable.
Cons: - another step for users to perform, taking more time. - may be seen as greater complexity. - might be too many "status choices" to keep track of.

5.1.3 Solution 1's Prototype

Solution 1's goal was to communicate to users the notion that they did not have to be confident to judge the correctness or incorrectness of values. 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 22. The lighter colored 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

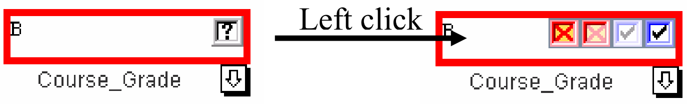


Figure 22. 4-Tuple: Clicking on the checkbox turns it into the four choices. The tool tips over the choices, starting with the left-most X, are "it's wrong," "seems wrong maybe," "seems right maybe," "it's right."

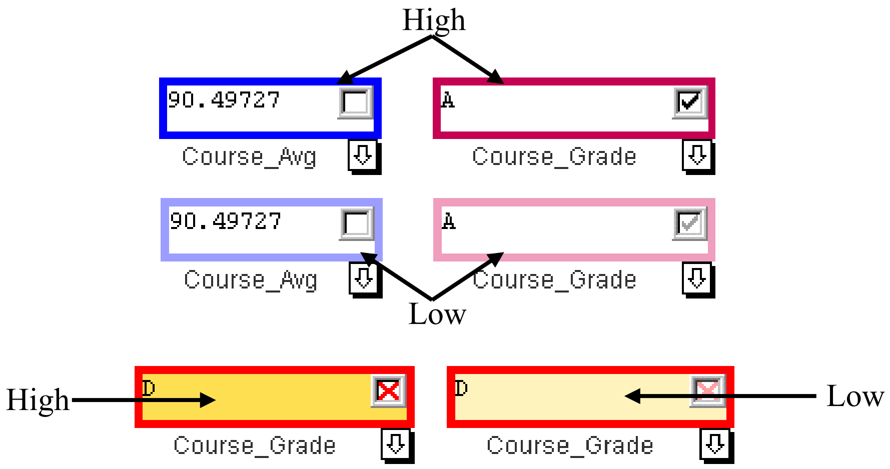


Figure 23: Effects of 4 Testing Decisions: Saturation of border color (top) and interior color (bottom) reflect confidence of user judgments of values being correct or incorrect.

saturations of interior colors reflecting the affected cells' fault likelihood. See Figure 23. 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. (The confidence value does not increase or decrease the testedness or fault likelihood values.)

5.1.4 Feedback from Users

As the prototype evolved, we brought in end users with no programming experience, one at a time, (two males and six females) to use our 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 Chapter 3. The tasks were followed by interviews.

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.

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."

However S3, a female, did not use the low-confidence marks and later told us she did not see their importance:

> 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."

We now turn to use of checkmarks and X-marks over several studies, using quantitative methods.

5.1.5 Usage Statistics

Since starting to collect the gender of participants we have conducted five large-scale quantitative studies from which we have gleaned information about checkmark and X-mark usage. Two of these studies (the tinkering study of Chapter 4 and a study looking of gender differences in strategies [Beckwith et al. 2007]) used the 4-tuple. The usage of checkmarks and X-marks from these studies is shown in Table 5-4. In each study users had access to the checkmarks and X-marks, but other factors differed between the individual studies (for example, different tutorials, number of problem-solving features available, and spreadsheet layout).

Table 5-4 only contains total checkmark and X-mark usage without separating this out from the use of the low- and high-confidence decisions (this is considered later). The goal of the 4-tuple is to encourage checkmark and X-mark usage, regardless of whether the marks being used are low- or high-confidence marks.

Table 5-4. Usage Statistics of Checkmarks and X-marks: Means (std. dev.) for usage of checkmarks, X-marks, and percentage of participants engaged with X-marks. Engagement is defined, as in Chapter 3, as using a feature more than once in at least one of the two tasks the participants completed. The shaded columns were experiments using the 4-tuple. Each of the 5 experiments had different sets of research questions, and several had different debugging features available. Also, pre-task tutorials ranged from detailed to a simple "tour of features" for Study 5.

	Study 1 [Ruthruff et al. 2004] 31 Males 23 Females	**Study 2 [Ruthruff et al. 2005]** 22 Males 16 Females	**Study 3 (Chapter 3)** 27 Males 24 Females	**Study 4a (Chapter 4 low-cost)** 20 Males 17 Females	**Study 4b (Chapter 4 high-support)** 16 Males 23 Females	**Study 5 [Beckwith et al. 2007]** 24 Males 37 Females
Check Usage						
Males	61.7 (32.7)	34.5 (13.9)	59.6 (27.1)	51.5 (20.9)	50.1 (24.9)	71.2 (32.7)
Females	59.7 (30.6)	38.4 (14.9)	49.3 (26.3)	56.2 (31.5)	52.5 (22.0)	63.7 (33.6)
X Usage						
Males	10.0 (8.6)	3.8 (3.8)	8.3 (6.9)	10.8 (9.6)	6.3 (7.0)	6.1 (11.7)
Females	6.3 (8.9)	3.0 (4.0)	4.5 (6.1)	4.9 (3.8)	5.8 (4.9)	2.3 (3.9)
X Engaged						
Males	90% (28/31)	54% (12/22)	81% (22/27)	70% (14/20)	56% (9/16)	42% (10/24)
Females	61% (14/23)	50% (8/16)	50% (12/24)	76% (13/17)	78% (18/23)	32% (12/37)

One interpretation of this data (Table 5-4) is that by comparing Studies 1-4a to Study 4b[12] the results show a change in usage of the X-marks. Considering X-mark data, both statistics (X usage and X Engaged) in Study 4b have the females ahead or close to the males in usage. This is a switch from the earlier studies – where generally the females were lower for at least one of these two measures.

Table 5-5 shows the number of high- and low-confidence checkmarks and X-marks, and the percentage of the total marks placed that were high-confidence. Both males

[12] We do not consider Study 5 in this interpretation because of a large difference in the tutorial. In the earlier studies when we introduced the checkmark and X-mark in the tutorial, we also made strategy suggestions of when to use the features, whereas in Study 5 no strategy suggestions were made for any features, making comparison difficult. As we discuss in the next section, strategy assistance seemed to interact with females' confidence.

Table 5-5. Usage of Confidence Marks: The usage of the high- and low-confidence marks, and the percentage of those marks that were high-confidence (excluding participants who only used high-confidence marks for the latter measure, since they are not the ones who necessarily see value in having high- and low-confidence choices). The two studies that used the 4-tuple are included in this table. The greater usage of the low-confidence marks (percentage wise), by both genders, is with the X-mark.

	Study 4b 16 Males 23 Females		**Study 5** 24 Males 37 Females	
Checkmarks	High/Low	% High-Conf.	High/Low	% High-Conf.
Males	49.0 / 1.1	94%	68.0 / 3.2	90%
Females	50.8 / 1.7	87%	59.3 / 4.4	87%
X-marks	High/Low	% High-Conf.	High/Low	% High-Conf.
Males	6.1 / 0.2	60%	4.9 / 1.2	49%
Females	5.4 / 0.4	64%	1.3 / 1.0	38%

and females used the low-confidence marks more frequently (as a percentage of total marks placed) when placing X-marks than checkmarks, suggestive of the value to both genders of having multiple choices when making testing decisions. Interestingly, qualitative follow-ups have revealed a second reason, other than confidence, to use the low-confidence marks. This is a "to-do" marker – an initial guess, with the intent to revisit that decision later.

5.2 Solution 2: Explanations

The addition of low-confidence marks may have helped with the usage of marks, but the evidence is not overwhelming. To strengthen our design solution, we decided to tackle Barrier 3 (Table 5-2) as well, perceived difficulty of learning, via the learning support vehicle in the system, explanations.

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."

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 minimalist learning theory in mind, 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.

5.2.1 Gender and Explanation Style

Several studies have found that the style of explanations that best help males and females succeed is different (summarized in [Arroyo 2003]). For example, Arroyo found that for children using a math tutoring system with a variety of hint types (provided after a mistake is made on a problem), girls' performance improved with highly interactive hints whereas boys' performance improved with less interactive hints [Arroyo 2003]. The same research also found that girls paid attention to any hint provided, whereas boys ignored them.

Another study from the same research team [Beck et al. 1999] found that the girls' confidence increased with highly interactive help whereas the boys' confidence increased the most with short (less interactive) help messages. As of the time of Solution 1, the design of our explanations still fit most closely with the type of help the males preferred in those studies: (1) our explanations were very brief, and (2) although they suggested an action, it was not elaborated upon. As we have just seen, both of these characteristics, at least in children, appear to favor males.

Gender differences in information processing [Meyers-Levy 1989] also suggest that supporting multiple explanation styles may be needed to support both genders well. Thus, we chose as a requirement for Solution 2 that our approach needed to support more than one explanation style. Note that we did not want to support females at the expense of the males, and we already had empirical evidence that our explanations were working reasonably well for a number of participants [Wilson et al. 2003, Robertson et al. 2004]. Thus, we elected to continue with the same explanations accessible through tool tips, but to also add support for expanded explanations on specific subtopics.

5.2.2 Requirements on Types of Explanation Content

We returned to the theories in Chapter 2 to also 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 above short explanations may not be well suited to females.

Anson's essay on minimalist learning theory, a second important influence on Solution 2, discusses content and delivery of minimalist documentation [Anson 1998]. Anson described content using the terms *conceptual*, *procedural*, and *problem solving*. These terms provide a useful framework for organizing requirements on explanations' content types presented in this chapter. Anson did not provide precise definitions, but we adopt the term "conceptual" to mean content relating to concepts and semantics, "procedural" to mean how to perform actions, and "problem solving" to mean 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 requirement's aim.

A final influence came from research on learning [Gorriz and Medina 2000] and problem-solving [Ames 2003] styles. These works have 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.

5.2.3 *Applying the Requirements*

The content type requirements of Section 5.2.2 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 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). 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 24.

In addition, the actual content of each type necessitated an orthogonal set of requirements. Table 5-6 lists the requirements, along with their originating theories.

5.2.3.1 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."

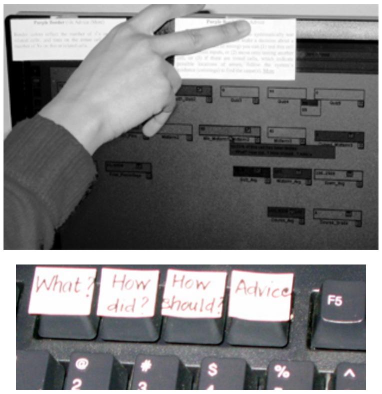


Figure 24. Low-cost Prototype: In our low-cost prototype, the user's request for an additional explanation component (bottom) caused the examiner to add it to the screen (top). Note the support for non-linear approaches—a user can view many unrelated components simultaneously.

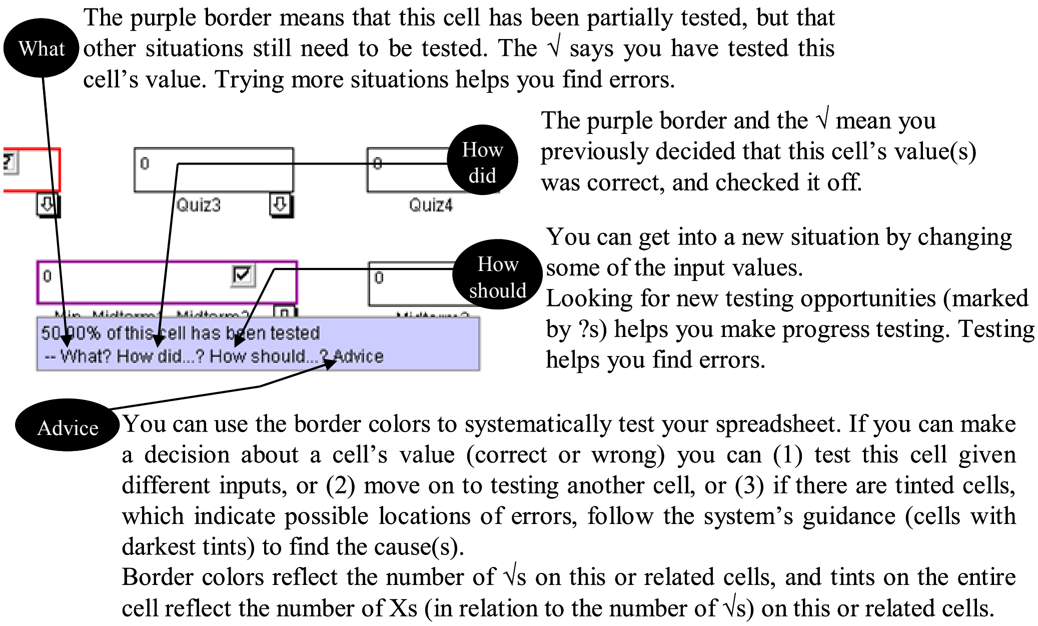
Table 5-6: Content Requirements: The explanation content requirements.

Content Requirements	**Sources**
1. Is task oriented.	Minimalist learning [Carroll 1998]
2. Keeps user active.	Various learning theories [Bransford 1999, Carroll 1998]
3. Explanation not too directive.	Various learning theories [Bransford 1999, Carroll 1998]
4. Explains how to evaluate whether an action taken was the right one to take.	Norman's action cycle [Norman 1988]
5. Is context-appropriate: User should care about information when presented.	Minimalist learning [Carroll 1998]
6. Suggests strategies for a difficult task.	Minimalist learning [Rosson et al. 1990]
7. Encourages user to take advantage of prior knowledge.	Various learning theories [Bransford 1999, Carroll 1998]
8. Explains why task is meaningful (relate to big picture).	Motivation (See Section 2.2.4 for summary of related literature.)
9. Provides enough information for users to accurately assess risks and benefits.	Risk, information processing [Meyers-Levy 1989], Attention investment [Blackwell 2002]
10. Makes obvious the actions that need to be taken.	Minimalist learning [Carroll 1998]
11. Makes sure rewards are clear.	Attention investment [Blackwell 2002]

Figure 25 shows an example of a short explanation ("50% of this cell has been tested") and the additional components. The goal of the "what" component is to communicate the semantics of the object in more detail than the short explanation. Thus, for this example, the "what" component is:

> *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.*

The first two sentences of this "what" component demonstrate Requirement 5 well (Table 5-6). This theory suggests that information be presented to users only when the information is relevant [Carroll 1998]. Separating the "what" component from the



50.00% of this cell has been tested
-- What? How did...? How should...? Advice

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. Trying more situations helps you find errors.

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

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. Testing helps you find errors.

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 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 (in relation to the number of √s) on this or related cells.

Figure 25. Explanation Component Example: 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.

"how" and "advice" components is one way we applied this theory, because it gives the user a way to communicate what question they are wondering about. We also applied this theory by tying the explanations to specific objects, where users, through their hovering actions, get information on exactly *which* object, in *which* state, they are curious about.

The last sentence of this component, "Trying more situations helps you find errors" demonstrates Requirements 8 and 11, by relating the object's current situation to the big picture and keeping the rewards clear.

Note the emphasis on testing, rather than on the actions and feedback. Several learning theories dissuade giving users information that is too directive (Requirement 3), resulting in users simply taking the action without thinking or learning from it [Bransford et al. 1999, Carroll 1998]. Thus, we elected to stress testing situations, rather than checking cells off to achieve a blue cell border as the goal.

5.2.3.2 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.*

This component was particularly influenced by Requirements 4 and 7. Its tie to Requirement 4 is simply that it helps the user to interpret the meaning of the feedback. Requirement 7, which comes from various learning theories, allows omission of information the user may well already know (and if not, they can always ask again via “how should”). According to these theories, this encourages users to make ties among the different explanation components and their experience using the spreadsheet features. These interconnections help them learn.

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.”

5.2.3.3 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.*

The second sentence of the above example aligns especially with two main themes of minimalist learning theory: keeping the user task-oriented and active (Requirements 1 and 2, respectively). It also reminds the user of the focus – testing – and suggests specific actions they can take to make progress on this task. Note that it also reminds them of the meaning of the "?" feedback device (Requirement 4), to help them evaluate the result of the action if they do take it.

5.2.3.4 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 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 (in relation to the number of √s) on this or related cells.*

The "advice" component satisfies Requirement 6, which is important when the complexity of a task is high and users need ideas on how to approach the task. In this example, the advice component suggests three strategies.

There is a fine balance in the advice components between providing enough information (Requirements 9, 10, and 11) without providing too much (Requirements 3 and 5). As Requirement 10 clarifies, it is important for the user to know how to follow through. Further, pertinent to satisfying Requirement 9, research on gender differences in perceived risk, risk aversion, and the way the females process information suggests that females may need many details before taking an action. Finally, according to the attention investment model, the explanation component may be important in decreasing users' perceptions of risk and/or increasing their perceptions of benefits.

5.3 Explanation Design Follow-Up

As with the 4-tuple, the evidence in favor of, or directly against the explanation as described above was scant; users of a small think-aloud rarely, if ever, requested the additional information. To best address the needs of users in explanations, other researchers on our team designed a bottom-up approach to finding the answer to what kind of information users need while debugging.

We[13] investigated end users' explanations needs through a bottom-up approach focusing on end users' information gaps during debugging. Pairs of end users debugged a spreadsheet together in Forms/3, but without any explanations and with only minimal introduction to the environment before starting on the debugging task. The environment was changed to take away the explanations since the explanation might give them information that would change their original questions.

Analysis of the think-aloud data resulted in a set of 10 codes which encapsulated all the users' information gaps. Table 5-7 shows the 10 codes with a short description of each. The most common information gaps were "Oracle/Specification," accounting for 40% of the information gaps. These were questions about debugging the spreadsheet, not about the environment's debugging features. Their information gaps would not have been answered by our "Explanation Components" as described in Section 5.2.

The second most common information gap group was "strategy" (including Strategy Hypothesis, Strategy Question, How Goal, and Concept from Table 5-7), which accounted for 30% of information gaps users expressed. These areas were well covered in the Explanations Components of Section 5.2, but were mostly not considered in the original explanation design. Of particular interest to our research on gender and self-efficacy was the category of "Self-Judgments" which accounted for 9% of information gaps. These were not accounted for by the Explanation

[13] This work was led by Kissinger, and reported in his master's thesis, and also a paper [Kissinger et al. 2006].

Table 5-7: Coding Scheme: The coding scheme as reported in [Kissinger et al. 2006]. The right most column includes where this information would be supplied in the explanations described in Section 5.2.3.

Code	Description	Explanation Component
Feature/ Feedback	Question or statement expressing general lack of understanding of the meaning of a specific visual feedback or action item, but with no goal stated.	Conceptual
Explanation	Explanation to help partner overcome an information gap. The explanation may be right or wrong.	Conceptual Procedural
Whoa	Exclamation of surprise or of being overwhelmed by the system's behavior.	
Help	Question or statement explicitly about the need for additional help.	Procedural Problem Solving
Self-Judgment	Question or statement containing the words "I" or "we," explicitly judging the participant or the pair's mastery of the environment or task.	
Oracle/ Specification	Question or statement reasoning about a value and/or a formula.	
Concept	Question about an abstract concept, as opposed to a question about a concrete feature/feedback item on the screen.	Conceptual
Strategy Question	Explicitly asks about what would be a suitable process or what to do next.	Procedural Problem Solving
How Goal	Asks how to accomplish an explicitly stated goal or desired action. (An instance of Norman's Gulf of Execution [Norman 1988].)	Procedural Problem Solving
Strategy Hypothesis	Suggests a hypothesized suitable strategy or next step to their partner.	Procedural Problem Solving

Components of Section 5.2. They are important for end users' self-efficacy, as both positive and negative self-judgments can impact end users' self-efficacy, and explanations that can help users make accurate assessments may be important for accurate self-efficacy assessment.

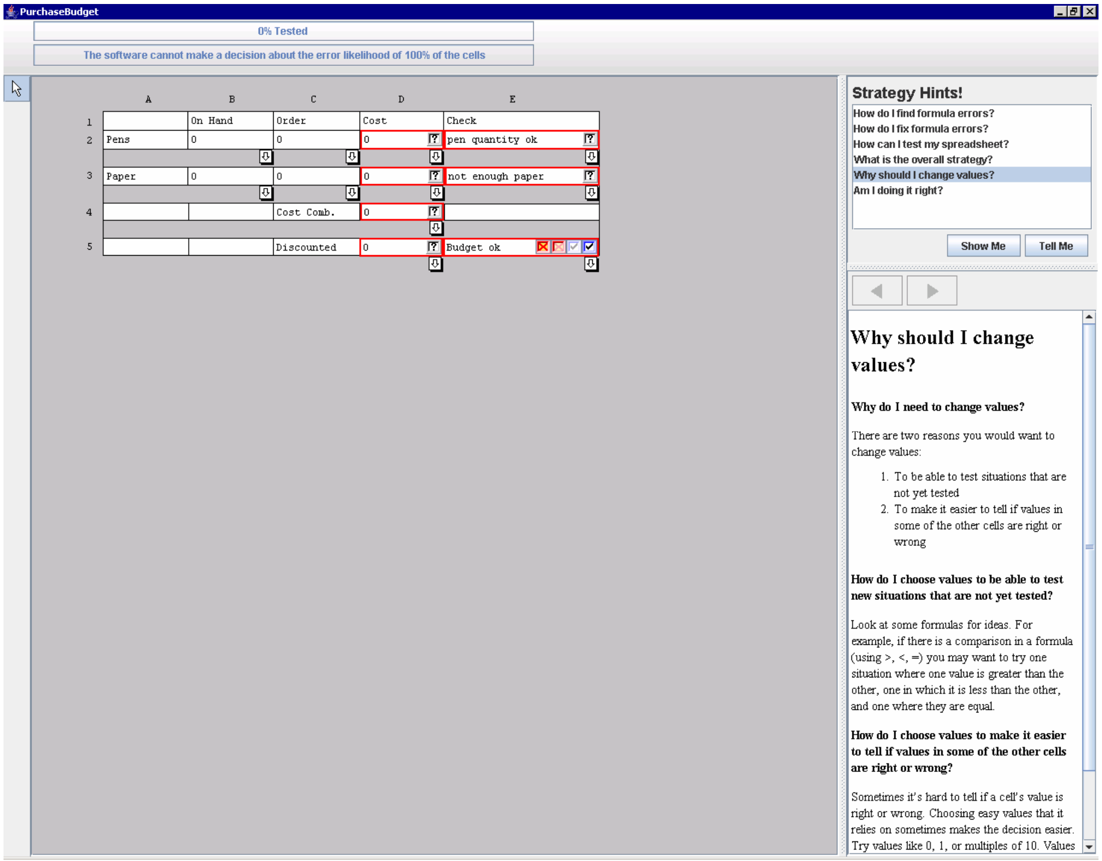
One of the most important take away messages from this empirical work was that debugging explanations for end-user programmers should not be primarily focused on how the debugging features work. In our study, feature-oriented explanations addressed only a fraction of what our participants wanted to know [Kissinger et al. 2006].

5.3.1 From Requirements to Prototype

The approach taken to address users' information needs was to expand the existing explanations (the version described in Chapter 3) by adding "strategy hints" to the environment [Subrahmaniyan et al. 2007]. This work extended the Explanation Components of "How do" (Section 5.2.3.2) and "Advice" (Section 5.2.3.4). These hints (in the form of recorded demonstrations and textual equivalent of the videos) focused on explaining debugging strategy (to address the 30% of the information gaps from Kissinger's et al.'s research [2006] that were about debugging strategy). The hints also included the self-judgment information gaps in our target (an additional 9% of the information gaps [Kissinger et al. 2006]). The hints did not focus on feature-specific explanations.

Strategy hint topics included: "How do I find formula errors?" "Am I doing it right?" "How can I test my spreadsheet?" There were 6 strategy hints in total. The topics and design of these strategy hints evolved from the Explanations Components of Section 5.2 and the Potential Solution of Barrier 2 in Table 5-2.

As Figure 26 shows, users can access the strategy hints from a side panel on the right side of their spreadsheet. After choosing a strategy hint they can click "Show Me" or "Tell Me." The "Show Me" button brings up a Windows Media Player in another window with a video (approximately two minutes in length) showing a male and female working through a problem on their own spreadsheet on the topic of the strategy hint. The "Tell Me" version was designed to be a quick reference to the strategy hints, and provides the same information, using exactly the same words as the video script (except for deletions of references to the example). However, there were no pictures of people or a spreadsheet in the "Tell Me" version.



0% Tested

The software cannot make a decision about the error likelihood of 100% of the cells

	A	B	C	D	E
1		On Hand	Order	Cost	Check
2	Pens	0	0	0	pen quantity ok
3	Paper	0	0	0	not enough paper
4			Cost Comb.	0	
5			Discounted	0	Budget ok

How do I find formula errors?
How do I fix formula errors?
How can I test my spreadsheet?
What is the overall strategy?
Why should I change values?
Am I doing it right?

Why should I change values?

Why do I need to change values?

There are two reasons you would want to change values:

1. To be able to test situations that are not yet tested
2. To make it easier to tell if values in some of the other cells are right or wrong

How do I choose values to be able to test new situations that are not yet tested?

Look at some formulas for ideas. For example, if there is a comparison in a formula (using >, <, =) you may want to try one situation where one value is greater than the other, one in which it is less than the other, and one where they are equal.

How do I choose values to make it easier to tell if values in some of the other cells are right or wrong?

Sometimes it's hard to tell if a cell's value is right or wrong. Choosing easy values that it relies on sometimes makes the decision easier. Try values like 0, 1, or multiples of 10. Values

Figure 26. Strategy Hints: Users can access the strategy hints from the right side-panel or through the tool-tip based explanations. All strategy hints have both a text and video (not shown) versions.

5.3.2 *Experiment*

Members of our research group ran a think-aloud study with the new "Strategy Hints" [Subrahmaniyan et al. 2007]. Their analysis of the data (10 participants: 7 males, 3 females) did not include gender in particular. We drew from their data to perform the qualitative analysis we present here.

During the study the participants were given a short "list of features" tutorial introducing them to the features, but they were not taught strategies instructing them how to use the features. The tutorial included an introduction to using the strategy hint features. During the actual task (the Payroll spreadsheet – see Figure 12) participants had 50 minutes, and were interrupted 20 minutes into the task and asked to read or watch a strategy hint of their choice. This was done to ensure that all participants read/watched at least one strategy hint during the task, as one of the main research goals was to assess the effect of the strategy hints on end users' information gaps. Participants were video- and audio-taped, and screen capture software captured their actions.

The main results from our colleagues' analyses were [Subrahmaniyan et al. 2007]:

Positive influences: There was a statistically significant effect on participants' strategy choices. Females were also particularly responsive to the confidence-boosting goal. Regarding information gaps, 56%, 49%, and 75% of the strategy, self-judgment, and oracle/specification gaps, respectively, were closed.

Issues: The explanations were not a panacea. Issues included participant misinterpretations of the explanations and lack of motivation or interest in them.

Presentation: Pronounced differences in participants' use of different media (video versus textual) demonstrated the critical importance of supporting a mix of presentation choices.

Using the data collected during the study, we then qualitatively analyzed the data to look for gender differences. For this analysis we used 6 of the 10 original participants (3 males, 3 females). Our three females were the only three females to participate in the study. The three males were chosen because they exhibited a range of task success and use of the strategy hints.

We had two main gender-related research questions: First, how did males and females react to the strategy hints? Second, how did the strategy hints appear to affect males' and females' use of environment features? Recall that part of the research that eventually led to the strategy hints was the understanding that users needed more information than our features-based explanation system (in the tooltips) provided, which may have been disproportionately negatively affecting the females.

Table 5-8 provides general information about the use of the strategy hints for the six participants.

5.3.2.1 The Videos: Watching "Show Me"

The strategy hint videos have particular qualities that the textual versions lack. For example, choosing to watch a video takes the user away from their spreadsheet. (The media player covers their spreadsheet.) Another difference is that the content of the video is set in the context of another spreadsheet, and the listeners are led through a story-like progression of the content. (In comparison, the text abstracts the points made in the video, since the text does not include pictures.) The video was designed to help boost those with low self-efficacy through "vicarious experience" (see Chapter 2, Table 2-1) – as the females watching may relate to the female in the video (Neeraja) and the males may relate to the male in the video (Jared). Each of these differences could impact males and females differently, particularly the effects of vicarious experience.

When Jessica and Marcia watched videos, their reactions were distinctive from Sean and Christopher's reactions. The females generally sat closer to the monitor during the videos, and nodded frequently, appearing to closely follow the story-line that Neeraja and Jared were working through. Particularly during the tutorial, when the two females watched videos they would smile often during the video, again something

Table 5-8. Strategy Hints Usage: Participants (fictitious names) and their use of "Tell Me" text and "Show Me" Videos. Their feature usage over the study.

Participant (self-efficacy)	Text	Video	√-mark on/off	Arrow on/off	X-mark on/off
Jessica (33)	3	1	40 / 9	6 / 5	5 / 0
Marcia (36)	3	2	13 / 1	31 / 37	3 / 1
Kimberly (38)	4	–	34 / 13	14 / 5	2 / 0
William (44)	1	–	42 / 2	6 / 12	0 / 0
Sean (33)	4	2	45 / 5	3 / 0	2 / 0
Christopher (40)	–	1	32 / 7	8 / 9	2 / 0

the males did not do often. In contrast, when Sean brought up a video on finding errors in the first few minutes of the task, he looked noticeably self-conscious, and just a few second into the video appeared to consider stopping it, as he hovered his mouse over the stop button. Instead he sat back and listened, but as soon as he heard something he wanted to apply to his spreadsheet he nodded and stopped the video. Sean and Christopher both sat back during their video watching, although they also appeared to be listening, as evidenced by an occasional nod during the video.

In addition to these differences, the females often switched between watching Neeraja and Jared and the video's spreadsheet indicated by their head and eye movements. Christopher and Sean appeared more focused on the spreadsheet rather than on Neeraja and Jared. In fact, when Sean watched his second video (near the end of the task) he was quite focused on the formula that Neeraja and Jared were discussing. At one point, as Neeraja paused while talking, Sean briefly turned to the area of the screen where their faces were, particularly noticeable since he had not been watching Neeraja and Jared earlier in the same video.

In their analysis of this study, Subrahmaniyan et al. found that Jessica and Marcia – the two females who watched videos – reported that the videos "made me feel more confident;" none of the males agreed with this statement [Subrahmaniyan et al. 2007]. This has links to self-efficacy. From both the females' actions while watching the videos and this later statement, there is suggestive evidence that in fact, these videos were impacting these two females in different ways than the videos affected the two males who watched them.

5.3.2.2 Watching "Show Me" and the Selectivity Hypothesis

As mentioned above, Sean brought up a video near the beginning of the task, listened to some of it, but then closed the video as soon as he had something he could take away from it. Sean's choice to return to his spreadsheet without watching more of the video is potentially an indication of heuristic processing [Meyer-Levy 1989]. Recall from Chapter 2 that this is the type of processing more often associated with males.

Jessica, in contrast appeared to follow more comprehensive processing when she returned to a video she had watched during the tutorial. Immediately after bringing up the video she skipped forward to exactly a part of the video that covered a cell's border turning purple. Her face visibly changed at this point, and her mouse followed along as she listened to the video. She appeared especially attentive to what Neeraja had to say. In fact, she said "oh" as Neeraja talked, indicating an insight about how to further test a cell. Despite this insight (recall the male above stopped as soon as he had the information he needed) she did not stop the video; she continued to watch until the end (just under another minute). This behavior matches more closely with comprehensive processing than with heuristic processing [Meyers-Levy 1989], since she took in more information than she needed to answer the specific question she had going into the video.

Sean and Jessica's actions present some evidence for Hypothesis SH-4: "Gender differences in information processing will impact the amount of information males and females desire prior to making problem-solving decisions. In particular, females may desire more information than males."

If males are more likely to watch videos in "Sean's style" – displaying evidence of heuristic processing – their actions may have the potential to lead to negative consequences. For example, understanding only that features exist does not necessarily lead to the correct, or even helpful, use of those features. This also has ties to Hypothesis SH-3: "Gender differences in information processing may affect males and females differently in their search for errors, with males being more prone to overlook specific cues about the location of errors within problem-solving environments such as spreadsheets."

5.3.2.3 Effects of Strategy Hints on Behavior

Several participants chose to use a feature after watching or reading the strategy hints. For example, Sean watched the video on finding errors and immediately placed checkmarks and X-marks on his own spreadsheet.

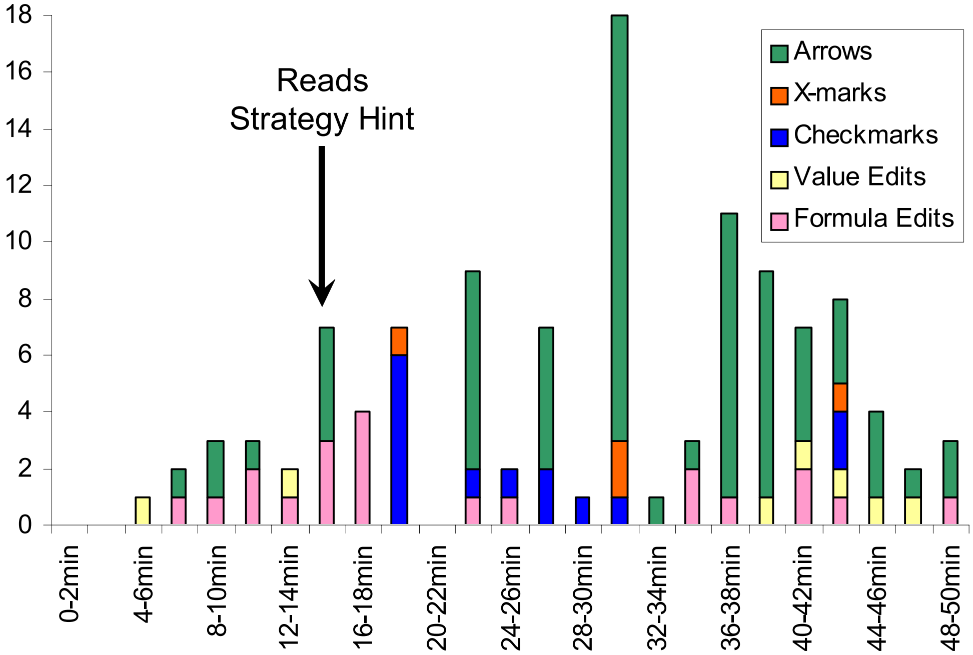


Figure 27. Actions after Strategy Hint: After reading about “How do I fix errors?” Marcia became serious about using arrows, and started using checkmarks and X-marks.

As Marcia read the strategy hint on how to fix errors, she read each bullet (vocalizing several words from each bullet). She then returned to her spreadsheet and used arrows, as the hint had suggested. As Figure 27 and the information from Table 5-8 show, Marcia became a frequent arrows user. Shortly after turning on the arrows and finding the results of using the arrows (in this particular usage occasion) did not lead to an immediate fix to her formula error she returned to the “Tell Me” and took another suggestion for moving parentheses in her formula around to try solving the formula error.

Jessica also appeared influenced by a strategy hint. She watched a video on testing during the tutorial. Less than 10 minutes into the task she placed her first checkmark, and followed this by changing an input value and getting the cell to 100% tested. She then said “okay, cool!” This kind of reward is what we aim to achieve. Jessica’s ability to get a cell blue was clearly a reward for her. She first saw a cell’s progression from red to purple to blue by watching a strategy hint, and then was able to apply the steps from the strategy hint to her own spreadsheet. Unfortunately, Jessica's attention to testing caused her to not focus on actually fixing bugs. It is an open question on how to ensure proper understanding of the information obtained in a strategy hint.

Christopher's behavior also changed after he watched "how to fix errors." The focus within this video was mainly on the formula that Neeraja and Jared were fixing. However, when the video started many cells had interior coloring due to X's placed in various areas of the spreadsheet, and other cells had been checked as being correct. This is never discussed in the video, but based on Christopher's actions after the video suggested he noticed: shortly after the video he said "let's go check all the boxes that I know are correct." Prior to this point in the task he had only made 3 formula changes, 6 value changes and used arrows one time, and as is shown in Figure 28, his actions

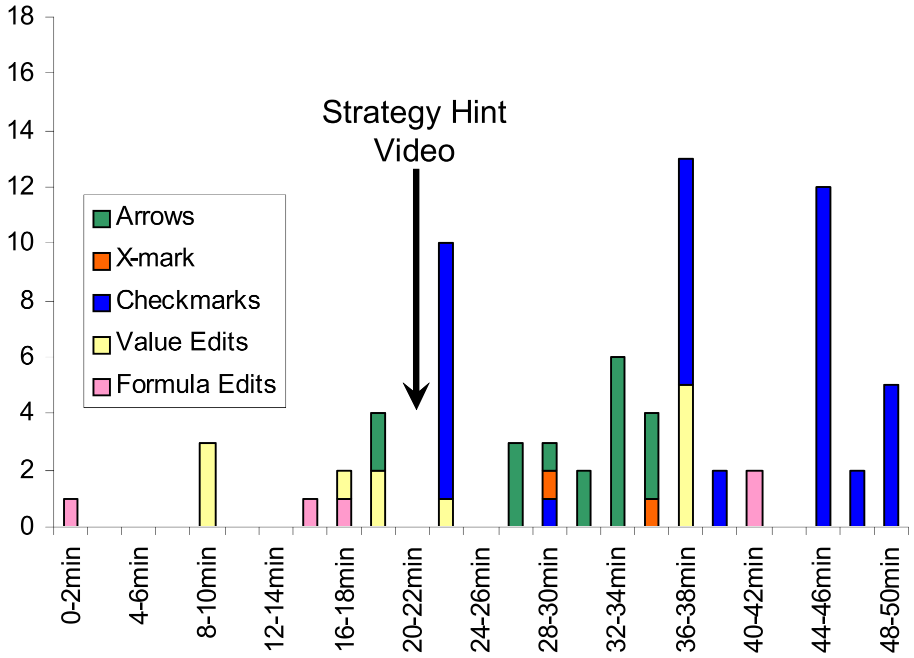


Figure 28. Actions After Strategy Hint: After watching the strategy hint: "How do I fix errors?" Christopher's use of features changed dramatically, with greater focus on testing and arrow usage.

after watching the video changed to very frequent use of the checkmarks, and some arrows.

In addition to the results reported throughout Section 5.3, this analysis suggests the following two new research questions:

What are the differences in males' and females' strategies to use the features they learn about in strategy hints? Are females more likely to follow exactly the suggestions proposed in the strategy hints?

5.4 Chapter Summary

This chapter covered the development of solutions to address the issues first realized in Chapter 3. The changes were specifically aimed at the females who appeared to be facing barriers in using our earlier design, although we intend that our changes will help both genders.

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 programming devices and strategies for combining the features.

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, minimalist learning theory, Norman's action cycle, and attention investment can be used to help understand barriers, derive requirements, and ultimately derive design ideas to address gender issues in end-user programming. Using the theory-derived design ideas, coupled with design techniques originally developed in HCI, we designed the specifics of our solutions, evaluated them analytically and through rapid prototyping, and informed our emerging approaches with an ongoing stream of users.

Result included:

- Evaluation of the 4-tuple “no confidence required” feature suggested the value to both genders of having multiple choices in making testing decisions.
- The strategy hints (with users’ choice of textual or video versions) helped close participants’ information gaps, and influenced participants’ overall debugging strategies.
- Gender differences in use of the strategy hints suggested that males and females may use different information processing strategies while watching videos.
- Females who watched the videos reported confidence-related benefits to watching them, but no males reported this benefit.

6. Excel Study

In the field of end-user programming, there have been many studies using academic prototypes, populations, and tasks. These studies often feature careful controls to limit the number of variables, and thus can achieve clean and clear results and insights. However, too often researchers never take the next step to explore the generalizability of their findings on real-world (widely used and commercially available) products. As a result, their findings cannot be trusted beyond the original, very limited setting.

This chapter takes the next step following up on the results of one of our earlier studies (the features experiment in Chapter 3) involving the Forms/3 academic prototype [Burnett et al. 2004]. The original study examined the effects of self-efficacy and gender on users' problem solving behaviors.

The purpose of the follow-up study was to explore how the original findings generalize to a broader population and a commercial spreadsheet environment. In doing so, the external validity of our original experimental results can be explored and expanded upon. Thus, in the study reported on in this chapter we examine how gender and self-efficacy impact end-user programmers' success and feature usage during an expanded end-user software engineering task. The main research question is: how do gender and self-efficacy results of end users generalize to a commercial environment as users engage in software engineering activities?

The term generalize is defined as "to give general applicability to" [Merriam-Webster 2007]. In experimental design a major threat to the external validity of the experiment is generalizability (in our case particularly to whether the results will generalize beyond the experimental environment and participant population). In order to generalize the results from our initial experiment we made the following changes:

- Different environment: Excel with unlimited access to features.
- Different population: Seattle-area real-world users of Excel.
- Different task: Spreadsheet modification, with emphasis on reliability of changes.

Another goal in follow-up experiment design is replication, to ascertain whether the same research results will occur if an experiment is replicated. (The term replicate is defined as "performance of an experiment or procedure more than once" [Merriam-Webster 2007].) There is a delicate balance between generalizing and replicating. If there are too many changes, the new study no longer replicates the original; if there are too few changes, hardly any generalization can occur. Thus, we replicated the initial experiment procedures to the extent possible given our generalization goals. The factors we replicated were:

- Task domain: End-user software engineering.
- Tutorial: Same style of teaching and introducing features to aid task.
- Design/Procedures: The design was the same, and the procedures were as similar as possible given the new setting.
- Research questions: the research questions were the same.

Our main interest was in the generalization to the widespread commercial system Excel. Unlike Forms/3, Excel has hundreds of features among which users must choose. Gender differences robust enough to apply to both these environments would allow understanding of the generality of how gender differences impact end-user programming activities.

The research questions of the "features experiment" (see Chapter 3) were the following:

> **RQ1:** Are there gender differences in self-efficacy that impact effective end-user programming?
>
> **RQ2:** Are there gender differences in users' likelihood of acceptance of unfamiliar features in end-user programming environments?

6.1 Related Work

Several researchers have studied real-world users and/or real-world situations, looking specifically at gender. Kelleher et al.'s work on gender and programming environments [Kelleher et al. 2007] has focused on middle school girls, and the types of environments that encourage computer programming. They found that girls become more engaged in programming and enjoy it more when the programming environment is designed for story-telling [Kelleher et al. 2007]. There are several differences between their work and our focus; first, their primary audience is children, second it is in the context of novice programming (i.e., those who aspire to enhance their programming skills) rather than end-user programmers (who may not have programming skill-building as a goal).

A few interview-style studies of end-user programmers in their "real lives" have also considered gender [Rode et al. 2004, Rosson et al. 2004]. Rode et al.'s research on home programming found different categories of appliances that were more likely to be programmed by men (e.g. entertainment devices) and by women (e.g. kitchen appliances). Their study involved both a real-world environment (of various home appliances) and real-world users. Rosson et al. [Rosson et al. 2004] also considered gender in their analysis of interviews with end-user web developers. In their sample they noticed that the four most sophisticated web developers were all males (7 of the 12 surveyed were males). Our study uses a real-world environment and real-world population, but the set-up is lab- and task-based, not interview-based.

6.2 Study Design

As mentioned in the Introduction, we replicated the features experiment's procedures (Chapter 3) to the extent possible.

6.2.1 Participants

We recruited participants from Microsoft's repository of Seattle-area residents interested in being part of a study for compensation of a Microsoft product. In order to be eligible to participate in this study, each participant had to meet the requirements of Table 6-1.

In total our participants were 21 males, 23 females. There were no gender differences in any background measure including: age, spreadsheet experience, programming experience, and education. Most participants (33/44) considered themselves Excel "intermediates." Median ages were 48 for males and 44 for females. Education was primarily the baccalaureate level. Only 9 participants (6 males and 3 females) had *never* created a spreadsheet for professional use.

Table 6-1. Participant Requirements: The minimum requirements participants had to meet in order to participate in the study.

Type	Requirement	Rationale
Age	20-60: 60 was the upper limit.	To generalize, we wanted a wide range of ages. Upper limit was set to avoid confounding factors due to deteriorating eyesight and other cognitive factors that occur with age.
Profession	Participants classifying themselves as a software developer, IT professional, computer engineer, or electrical engineer were disqualified.	Our interest was in end-user programmers. These professions are closer to professional programming than to end-user programming.
Excel experience	Participants could classify their Excel experience as: beginner, intermediate, advanced, or expert. Answering "no experience" disqualified them.	Since the population of interest to us is people already engaged in this type of end-user programming, some prior experience with Excel was required.
Programming background	Participants who had taken 2 or more courses in Java- and/or Perl- like programming were disqualified. (Web programming and Visual Basic were also allowed.)	Some programming coursework was allowed because, given modern business degree requirements, young business adults have usually taken 1-2 programming courses in high school and/or college.
Experience with macros	Participants who had programmed Excel macros were disqualified.	This level of sophistication with Excel is beyond that of many business users.
Disqualifying features	If participants had previously used data validation, the watch window, or evaluate formula they were disqualified.	We wanted to analyze the use of these specific features without the participants having prior knowledge of them.
Qualifying Features	Participants had to have used three or more of the functions from the following list: average, count, countif, hlookup, if, indirect, lookup, max, min, round, sum, and sumif.	To avoid spreadsheet illiteracy as a confound, it was important to ensure that participants had some experience with reasonably complicated formulas.

6.2.2 Environment

The experiment took place using the real-world environment of Microsoft Excel 2001. Because Excel is the mostly widely used end-user programming language, this environment is an ideal choice for examining the generalization of the features experiment results.

To as closely as possible replicate the purpose of the features experiment, this study followed along the same theme of end-user software engineering. We were interested in factors and features in Excel that would promote the reliability of the spreadsheet formulas. Excel's audit toolbar feature has several features that aid users in ensuring reliability, and allow them to engage in end-user software engineering activities. As shown in Figure 29, the audit toolbar contains 12 buttons. Five of these (numbers 2-6 in Figure 29) support operations with dataflow arrows. Two (numbers 1 and 7) relate to Excel's error checking of cells flagged as being inconsistent or otherwise suspicious. Two (numbers 9 and 10) relate to Excel's "validation" feature, in which users can check if any of their cells' values violate expected ranges (also set by the user). Finally, number 11 is for watching cell's values that may be off-screen or on another sheet all together, and number 12, evaluate formula, allows a user to go step-by-step in evaluating a formula.

Although we focused on the audit toolbar, we did not in any way restrict the participants to only these features. In comparison to the features experiment (with only 4 features – see Figure 29), participants in this study had to choose between hundreds of Excel features.

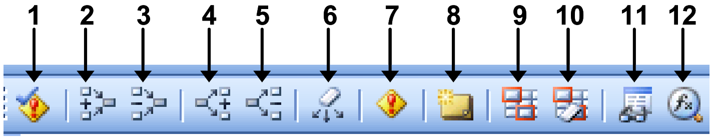


Figure 29: Audit Toolbar: To stay with the theme of end-user software engineering the experiment emphasized the use of the features in the audit toolbar to aid formula reliability.

6.2.3 Tutorial

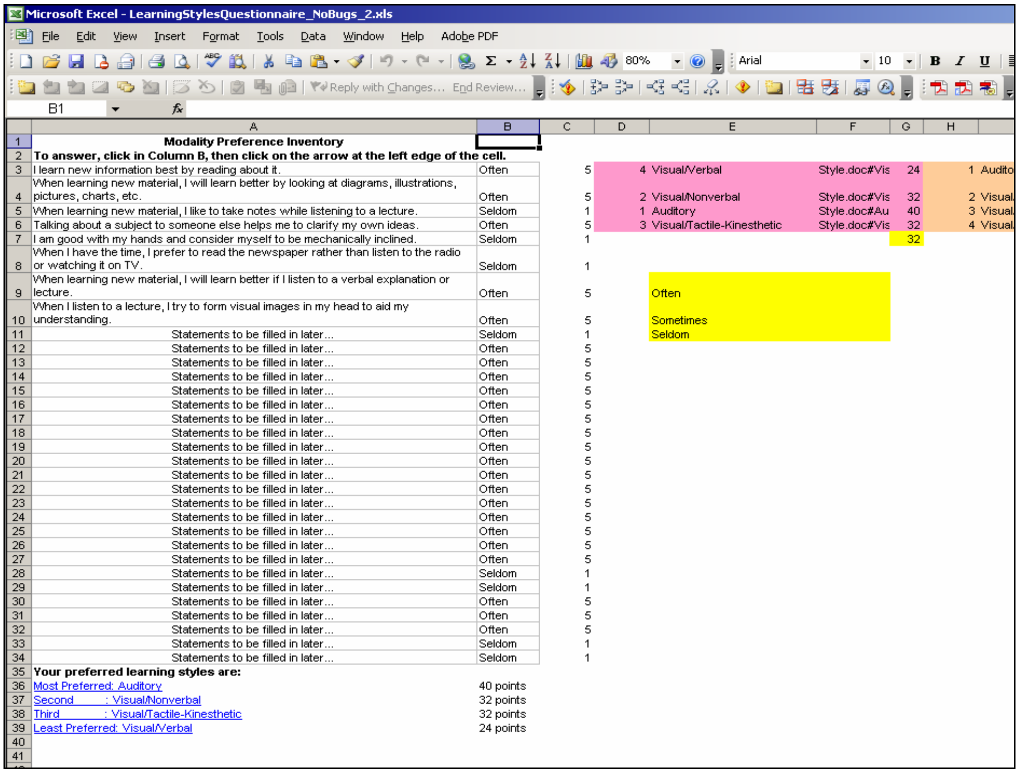
The tutorial was designed under the same requirements as the features experiment. As with the previous tutorial, it was hands-on, and lasted about 30 minutes. Its purposes were to (1) focus participants' attention on the goal of formula reliability, (2) teach features in the "taught" category, and (3) also call attention to (but not teach) features in the "untaught" category.

The taught features were the arrows (numbers 2-6 from Figure 29). For these features, the instructor described how to use the feature and its feedback once used. The taught features were used multiple times during the study. The untaught features singled out by the instructor were the error checking buttons and the evaluate formula button (numbers 1 and 12). Users were encouraged to explore all audit toolbar features.

Throughout the tutorial, participants learned to use the taught features, and experimented as they wished with the untaught ones focusing on the reliability of the spreadsheet as they worked. They also learned about the "IF" function in Excel, because in previous studies, a number of participants have stumbled on use of that function, and we wanted to avoid introducing "noise" relating to misunderstandings of IF into our data.

The spreadsheet they worked on during the tutorial came from the EUSES corpus of real-world spreadsheets [Fisher II and Rothermel 2005], with slight modifications for tutorial suitability. The spreadsheet (see Figure 30) was a learning styles questionnaire, participants' task during the tutorial was to introduce two new rows into the spreadsheet for two new learning styles questions – which would then have to be accounted for in several downstream formulas. One of these was completed step-by-step during the tutorial. This maintenance-style task was designed to be similar to one of the tasks in the main part of the experiment.

At the end of the tutorial, once one of the modifications had been made, the participants had several minutes to explore the features they had just learned about and to make the second modification (add the next question) to the spreadsheet.


Microsoft Excel - LearningStylesQuestionnaire_NoBugs_2.xls
File Edit View Insert Format Tools Data Window Help Adobe PDF
80%
Arial
10
Reply with Changes... End Review...
B1

	A	B	C	D	E	F	G	H
1	**Modality Preference Inventory**							
2	**To answer, click in Column B, then click on the arrow at the left edge of the cell.**							
3	I learn new information best by reading about it.	Often	5	4	Visual/Verbal	Style.doc#Vis	24	1 Audito
4	When learning new material, I will learn better by looking at diagrams, illustrations, pictures, charts, etc.	Often	5	2	Visual/Nonverbal	Style.doc#Vis	32	2 Visual
5	When learning new material, I like to take notes while listening to a lecture.	Seldom	1	1	Auditory	Style.doc#Au	40	3 Visual
6	Talking about a subject to someone else helps me to clarify my own ideas.	Often	5	3	Visual/Tactile-Kinesthetic	Style.doc#Vis	32	4 Visual
7	I am good with my hands and consider myself to be mechanically inclined.	Seldom	1				32	
8	When I have the time, I prefer to read the newspaper rather than listen to the radio or watching it on TV.	Seldom	1					
9	When learning new material, I will learn better if I listen to a verbal explanation or lecture.	Often	5		Often			
10	When I listen to a lecture, I try to form visual images in my head to aid my understanding.	Often	5		Sometimes			
11	Statements to be filled in later...	Seldom	1		Seldom			
12	Statements to be filled in later...	Often	5					
13	Statements to be filled in later...	Often	5					
14	Statements to be filled in later...	Often	5					
15	Statements to be filled in later...	Often	5					
16	Statements to be filled in later...	Often	5					
17	Statements to be filled in later...	Often	5					
18	Statements to be filled in later...	Often	5					
19	Statements to be filled in later...	Often	5					
20	Statements to be filled in later...	Often	5					
21	Statements to be filled in later...	Often	5					
22	Statements to be filled in later...	Often	5					
23	Statements to be filled in later...	Often	5					
24	Statements to be filled in later...	Often	5					
25	Statements to be filled in later...	Often	5					
26	Statements to be filled in later...	Often	5					
27	Statements to be filled in later...	Often	5					
28	Statements to be filled in later...	Seldom	1					
29	Statements to be filled in later...	Seldom	1					
30	Statements to be filled in later...	Often	5					
31	Statements to be filled in later...	Often	5					
32	Statements to be filled in later...	Often	5					
33	Statements to be filled in later...	Seldom	1					
34	Statements to be filled in later...	Seldom	1					
35	**Your preferred learning styles are:**							
36	Most Preferred: Auditory	40 points						
37	Second : Visual/Nonverbal	32 points						
38	Third : Visual/Tactile-Kinesthetic	32 points						
39	Least Preferred: Visual/Verbal	24 points						
40								
41								

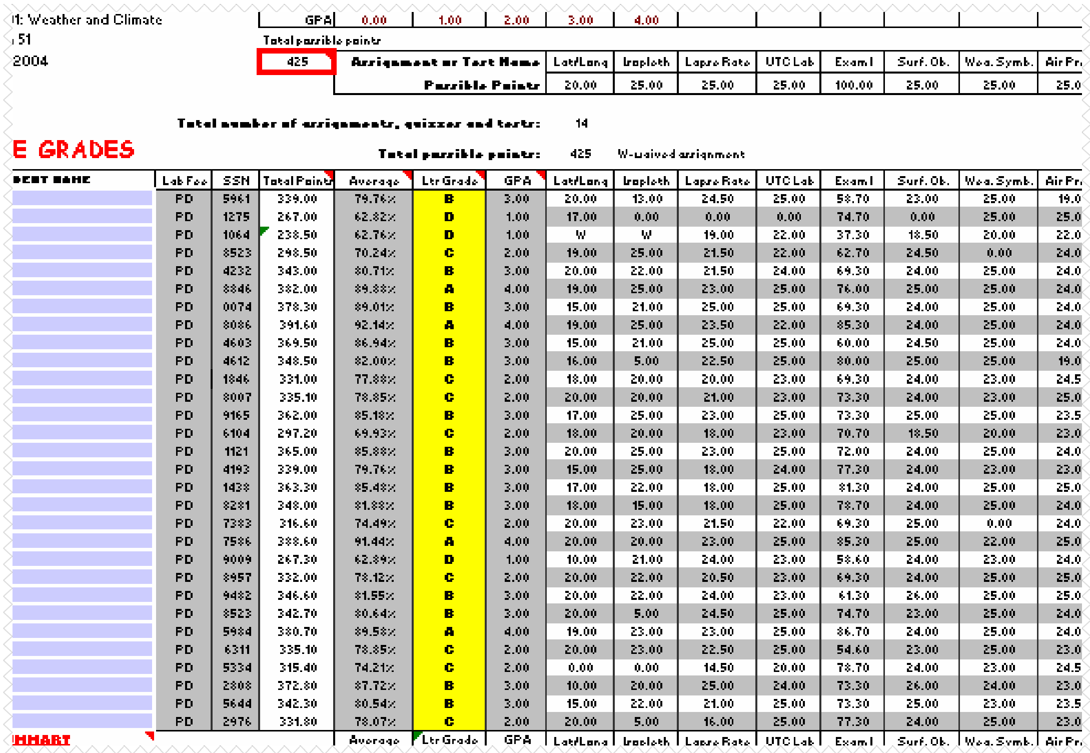
Figure 30. Tutorial Spreadsheet: The spreadsheet participants added two new rows to during the tutorial. This spreadsheet was also used to introduce the taught and untaught features. Figure is shrunk and cropped to give a sense of the overall size.

6.2.4 Main Spreadsheet and Tasks

The main experiment required participants to make two modifications to a grade book spreadsheet. This spreadsheet, obtained from the EUSES Spreadsheet Corpus of real-world spreadsheets [Fisher II and Rothermel 2005], is shown in Figure 31.

We chose to make the tasks modification tasks—instead of debugging as in the features experiment—for generalization purposes. Modification includes debugging, and hence covers both the "create" and the "debug" phase of end-user programming.

The modification tasks were designed with two criteria in mind. First, they needed to be grounded in the real world. For this reason, we drew the spreadsheet and the task ideas from the EUSES Corpus of real-world spreadsheets. Second, the tasks needed to be complicated enough to warrant use of the auditing toolbar features. If the



1: Weather and Climate
51
2004

	GPA	0.00	1.00	2.00	3.00	4.00

Total possible points

425	Assignment or Test Name	Lat/Long	Isopleth	Lapse Rate	UTC Lab	Exam I	Surf. Ob.	Wea. Symb.	Air Pr.
	Possible Points	20.00	25.00	25.00	25.00	100.00	25.00	25.00	25.0

Total number of assignments, quizzes and tests: 14

E GRADES

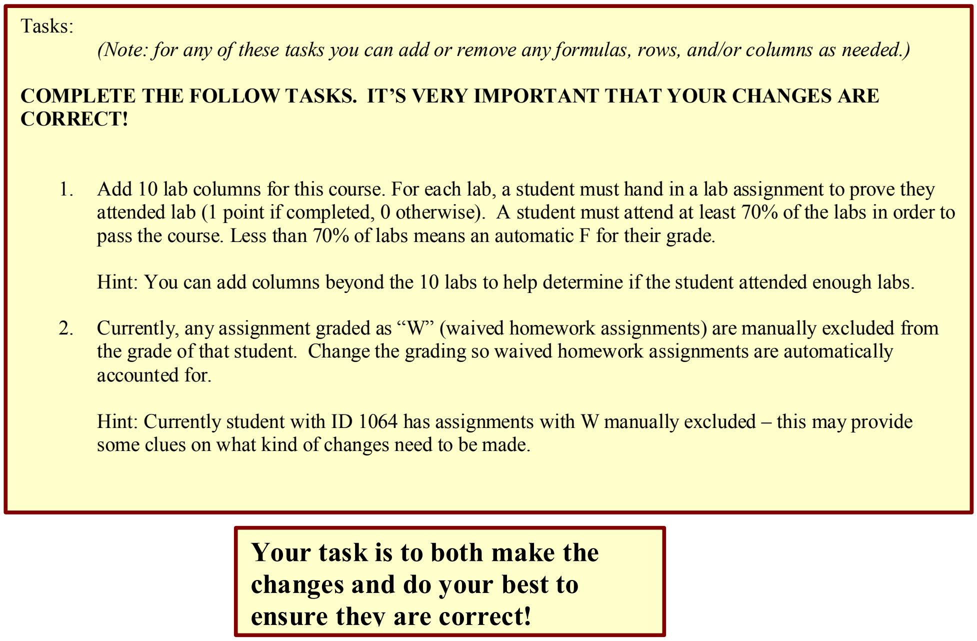
Total possible points: 425 W-waived assignment

DENT NAME	Lab Fee	SSN	Total Point	Average	Ltr Grade	GPA	Lat/Long	Isopleth	Lapse Rate	UTC Lab	Exam I	Surf. Ob.	Wea. Symb.	Air Pr.
	PD	5961	339.00	79.76%	B	3.00	20.00	13.00	24.50	25.00	58.70	23.00	25.00	19.0
	PD	1275	267.00	62.82%	D	1.00	17.00	0.00	0.00	0.00	74.70	0.00	25.00	25.0
	PD	1064	238.50	62.76%	D	1.00	W	W	19.00	22.00	37.30	18.50	20.00	22.0
	PD	8523	298.50	70.24%	C	2.00	19.00	25.00	21.50	22.00	62.70	24.50	0.00	24.0
	PD	4232	343.00	80.71%	B	3.00	20.00	22.00	21.50	24.00	69.30	24.00	25.00	24.0
	PD	8846	382.00	89.88%	A	4.00	19.00	25.00	23.00	25.00	76.00	25.00	25.00	24.0
	PD	0074	378.30	89.01%	B	3.00	15.00	21.00	25.00	25.00	69.30	24.00	25.00	24.0
	PD	8086	391.60	92.14%	A	4.00	19.00	25.00	23.50	22.00	85.30	24.00	25.00	24.0
	PD	4603	369.50	86.94%	B	3.00	15.00	21.00	25.00	25.00	60.00	24.50	25.00	24.0
	PD	4612	348.50	82.00%	B	3.00	16.00	5.00	22.50	25.00	80.00	25.00	25.00	19.0
	PD	1846	331.00	77.88%	C	2.00	18.00	20.00	20.00	23.00	69.30	24.00	23.00	24.5
	PD	8007	335.10	78.85%	C	2.00	20.00	20.00	21.00	23.00	73.30	24.00	23.00	25.0
	PD	9165	362.00	85.18%	B	3.00	17.00	25.00	23.00	25.00	73.30	25.00	25.00	23.5
	PD	6104	297.20	69.93%	C	2.00	18.00	20.00	18.00	23.00	70.70	18.50	20.00	23.0
	PD	1121	365.00	85.88%	B	3.00	20.00	25.00	23.00	25.00	72.00	24.00	25.00	24.0
	PD	4193	339.00	79.76%	B	3.00	15.00	25.00	18.00	24.00	77.30	24.00	23.00	23.0
	PD	1438	363.30	85.48%	B	3.00	17.00	22.00	18.00	25.00	81.30	24.00	25.00	25.0
	PD	8281	348.00	81.88%	B	3.00	18.00	15.00	18.00	25.00	78.70	24.00	25.00	24.0
	PD	7383	316.60	74.49%	C	2.00	20.00	23.00	21.50	22.00	69.30	25.00	0.00	24.0
	PD	7586	388.60	91.44%	A	4.00	20.00	20.00	23.00	25.00	85.30	25.00	22.00	25.0
	PD	9009	267.30	62.89%	D	1.00	10.00	21.00	24.00	23.00	58.60	24.00	23.00	24.0
	PD	8957	332.00	78.12%	C	2.00	20.00	22.00	20.50	23.00	69.30	24.00	25.00	25.0
	PD	9482	346.60	81.55%	B	3.00	20.00	22.00	24.00	23.00	61.30	26.00	25.00	25.0
	PD	8523	342.70	80.64%	B	3.00	20.00	5.00	24.50	25.00	74.70	23.00	25.00	24.0
	PD	5984	380.70	89.58%	A	4.00	19.00	23.00	23.00	25.00	86.70	24.00	25.00	24.0
	PD	6311	335.10	78.85%	C	2.00	20.00	23.00	22.50	25.00	54.60	23.00	25.00	23.0
	PD	5334	315.40	74.21%	C	2.00	0.00	0.00	14.50	20.00	78.70	24.00	23.00	24.5
	PD	2808	372.80	87.72%	B	3.00	10.00	20.00	25.00	24.00	73.30	26.00	24.00	23.0
	PD	5644	342.30	80.54%	B	3.00	15.00	22.00	21.00	25.00	73.30	25.00	23.00	23.5
	PD	2976	331.80	78.07%	C	2.00	20.00	5.00	16.00	25.00	77.30	24.00	25.00	23.0
MMARY				Average	Ltr Grade	GPA	Lat/Long	Isopleth	Lapse Rate	UTC Lab	Exam I	Surf. Ob.	Wea. Symb.	Air Pr.

Figure 31. Main Task Spreadsheet: A snapshot of part of the grade book spreadsheet from the EUSES Spreadsheet Corpus [Fisher II and Rothermel 2005]. Participants' tasks were to make modification to formulas, and add a new lab section. Figure is shrunk and cropped to give a sense of the overall size.

modification tasks were too easy, we feared there would be no reason for participants to consider use of these features.

The first modification task (#1) was drawn directly from a second real-world spreadsheet from the corpus, in which the teacher was incorporating completion of lab assignments into the students' grade. The second modification (#2) was to solve an error proneness problem with the current spreadsheet. Without the second modification, teachers would have to manually override formulas for students with waived homework assignments; the modification was thus to instead change the formulas so that they could calculate the grades for any student with or without waived homeworks. Figure 32 shows the wording for both tasks.



Tasks:

(Note: for any of these tasks you can add or remove any formulas, rows, and/or columns as needed.)

COMPLETE THE FOLLOW TASKS. IT'S VERY IMPORTANT THAT YOUR CHANGES ARE CORRECT!

1. Add 10 lab columns for this course. For each lab, a student must hand in a lab assignment to prove they attended lab (1 point if completed, 0 otherwise). A student must attend at least 70% of the labs in order to pass the course. Less than 70% of labs means an automatic F for their grade.

   Hint: You can add columns beyond the 10 labs to help determine if the student attended enough labs.

2. Currently, any assignment graded as "W" (waived homework assignments) are manually excluded from the grade of that student. Change the grading so waived homework assignments are automatically accounted for.

   Hint: Currently student with ID 1064 has assignments with W manually excluded – this may provide some clues on what kind of changes need to be made.

Your task is to both make the changes and do your best to ensure they are correct!

Figure 32. Spreadsheet Tasks: The two tasks for participants to complete. The first was based on another spreadsheet from the EUSES Spreadsheet Corpus [Fisher II and Rothermel 2005].

6.2.5 Questionnaires

A pre-session questionnaire collected participant background data and participants' self-efficacy. The following background data were collected: gender, degree program, highest degree completed, current job title, programming experience, previous spreadsheet experience, professional spreadsheet experience, and whether English was their primary language.

Following the experiment a post-session questionnaire assessed comprehension of the audit toolbar features with a set of 22 multiple choice and true/false questions. Both pre-session and post-session questionnaires are in Appendix B.

6.3 Results that Generalized

6.3.1 Effectiveness

Features Experiment Result: Females' self-efficacy was predictive of their effectiveness at using the debugging features (which was not the case for the males).

To analyze this question, we used the measure possible with the data that was most related to effectiveness with the debugging features. This was a measure of success on the task – specifically how much of the two tasks were attempted and/or completed. To determine "how much" we divided the tasks into small sub-tasks. Points were assigned for correctly completing tasks, with partial credit for (incorrectly) attempting completion of the subtask. The sum of points is the "task success."

Females' self-efficacy was a predictive indicator of their task success (linear regression: F(1,21)=7.2, ß=0.47, R^2=0.26, p<0.01). Males' self-efficacy, however, was not a significant predictor of their task success (linear regression: F(1,19)=2.19, ß=0.15, R^2=0.10, p<0.16). Figure 33 shows the males' and females' relationships between self-efficacy and task success. These findings are consistent with the above features experiment result.

Self-efficacy was also a significant predictor of task success for all participants (F(1,42)=6.99, ß=0.24, R^2=0.14, p<0.01), but this was due to the females. This is

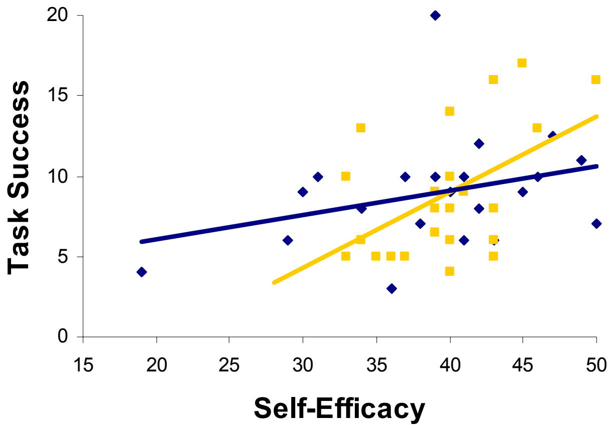


Figure 33. Self-efficacy and Task Success: For the females (light), self-efficacy predicted task performance. This relationship did not for the males (dark).

indicated by the R^2 values—a measure of how much of the variance in task success self-efficacy described—showing that the separate analysis of the preceding paragraph provides a better fit to the female data than with the combined group, and that most of the male outcomes were not explained by self-efficacy.

6.3.2 Comprehension

Features Experiment Result: No difference in the males' and females' ability to learn the new features.

One possible explanation for the result of Section 6.3.1 is that the females made better judgments (through the rating of their self-efficacy) than the males did regarding their abilities to understand and use the features effectively.

In the features experiment, this was not the case. A comprehension post-test showed that there was no difference in males' and females' understanding of how the debugging features worked, or interpretation of their feedback, etc. Females' low self-efficacy was a self-fulfilling prophecy: low belief in their ability impacted their willingness to engage with the features, although their understanding would not have predicted this difference.

Turning to the current study, the comprehension post-test also showed no difference in males' and females' comprehension of the audit toolbar features with females scoring a median of 12 points (22 possible), males a median of 11 (t-test: $t=0.72$, $df=42$, $p<0.47$). Furthermore, self-efficacy also did not predict comprehension for either gender (linear regression: males: $F(1,19)=0.16$, $ß=-0.04$, $R^2=0.008$, $p<0.69$; females: $F(1,21)=0.56$, $ß=0.12$, $R^2=0.03$, $p<0.46$). These results suggest that, as in the features experiment, females' self-efficacy was more of a self-fulfilling prophecy than an accurate assessment of abilities.

6.3.3 Familiar Features

Features Experiment Result: Females had a higher adoption rate of the Type Familiar feature (formula edits) than the males did.

Familiar features were those features not defined as taught or untaught features (see Section 6.2.3). The category included formula and value manual edits (i.e., without using replicate features), and basic features such as bold, copy/paste, and insert function. A t-test revealed no gender differences in the overall use of the familiar features (t-test: t=0.51, df=42, p<0.62). But, as Figure 34 clearly suggests, females' and males' relationships between their self-efficacy and use of the familiar features differed. Females' self-efficacy was inversely predictive of their use of these features: as their self-efficacy decreased their use of the familiar features increased (linear regression: F(1,21)=10.08, ß=-15.8, R^2=0.32, p<0.005). For males, the relationship is not significant (F(1,19)=0.49, ß=-2.65, R^2=0.03, p<0.49).

The regression relationship for the females is consistent with the features experiment. Specifically, low self-efficacy females concentrated more of their efforts on the familiar features, particularly when compared to the high self-efficacy females.

Closer scrutiny of these relationships showed that they were almost entirely due to (manual) value edits. Unlike the features experiment, which had only one feature classified as familiar—formula edits, not value edits—in this study, there were several groups classified as such, as shown in Table 6-2. (The role of values was different in

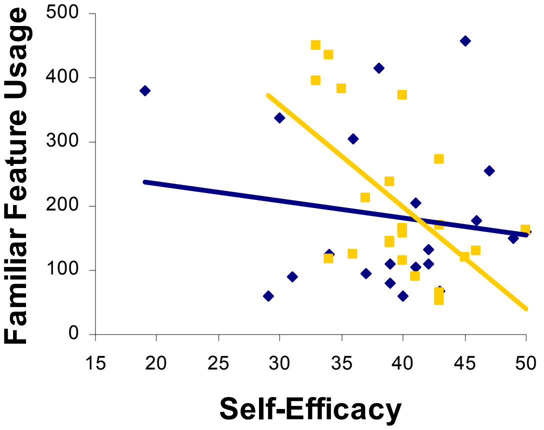


Figure 34. Self-Efficacy and Familiar Feature Usage: For the females (light), lower self-efficacy was a significant predictor of higher usage of familiar features. For the males (dark), there was no relationship.

Table 6-2. Familiar Feature Usage: Mean (std. dev.) for the components of the familiar features. No significant gender differences in usage.

Familiar Features	Males	Females
Value (manual) edits	125.1 (111.8)	151.7 (124.3)
Formula (manual) edits	29.5 (31.5)	26.0 (16.5)
Other basic features	28.0 (33.5)	25.2 (15.8)

the previous study, due to the connection with a testing tool.)

Females' self-efficacy inversely predicted use of these manual value edits (but no other sub-category), as shown in Figure 35 (linear regression: edit values: F(1,21)=12.08, ß=-17.22, R^2=0.37, p<0.002; edit formulas: F(1,21)=2.10, ß=1.14, R^2=0.09, p<0.16; other features: F(1,21)=0.03, ß=0.14, R^2=0.001, p<0.86). For males, self-efficacy did not predict any use of the three sub-categories (linear regression: edit values: F(1,19)=0.32, ß=-1.9, R^2=0.02, p<0.58; edit formulas: F(1,19)=0.21, ß=-0.44, R^2=0.01, p<0.65; other features: F(1,19)=0.08, ß=-0.29, R^2=0.004, p<0.78).

The distinction between values entered manually and those entered using replicate (e.g., by copy/pasting to several cells, fill functions, and the cross bar in the lower-right corner of a cell) provided insights into the females' behaviors. In fact, values entered using replicate were predicted by self-efficacy for the females, but the

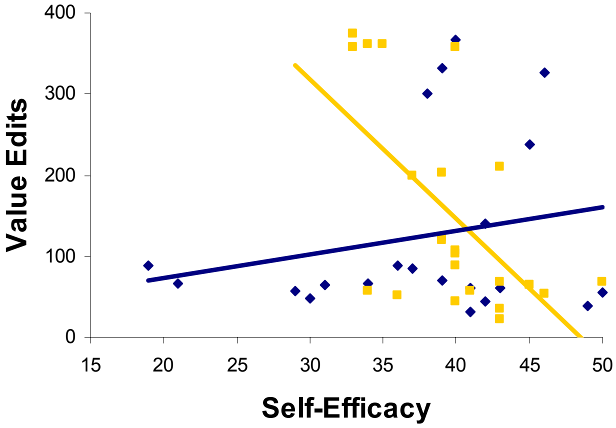


Figure 35. Self-Efficacy and Value Edits: Self-efficacy (inversely) predicts value edits for females (light), but not for males (dark).

relationship is *opposite* than that of the manual value edits. Shown in Figure 36, self-efficacy is a predictive indicator for the percentage of females' total value edits that are filled using replicate, but not so for the males (linear regression: males: F(1,19)=2.37, ß=1.17, R^2=0.11, p<0.14; females: F(1,21)=6.26, ß=3.9, R^2=0.23, p<0.02). This same pattern is consistent for formula edits entered using replicate (linear regression: males: F(1,19)=0.94, ß=2.19, R^2=0.05, p<0.34; females: F(1,21)=9.05, ß=14.6, R^2=0.30, p<0.007).

Self-efficacy theory provides an interpretation for this difference. According to self-efficacy theory "people tend to avoid tasks and situations they believe exceed their capabilities, but they undertake and perform assuredly activities they judge themselves capable of handling" [Bandura 1977]. From this perspective, low self-efficacy females spending their time manually entering values may be avoiding a challenging part of the task.

For example, one aspect of task #1 (see Section 6.2.4), providing additional columns, could be interpreted to mean that many data values should be typed in. Another subtask was to write an "IF" formula, arguably the most challenging sub-task of task #1. Of the 10 low self-efficacy females (defined as females with self-efficacy below the median of 40), only 1 attempted the "IF" (unsuccessfully), compared with the high

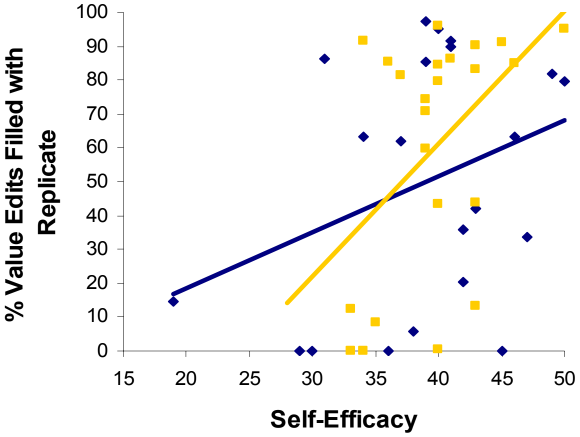


Figure 36. Self-Efficacy and Values Replicated: Self-efficacy predicts percentage of value edits filled using replicate for the females (light), but not the males (dark).

self-efficacy females of whom 6 of the 13 made the change correctly: a statistically significant result (t-test: t=2.3, df=21, p<0.03). This result suggests that low self-efficacy females, in avoiding a subtask they believed exceeded their capabilities, focused on the part of the task they knew they could do—manually entering values.

6.4 Unconfirmed Results

Some of the features experiment results were not confirmed. Those are discussed briefly here.

Features experiment Result: females had significantly lower self-efficacy than the males.

In the current study, both males and females had a median self-efficacy of 40 (t-test: t=0.41, df=42, p<0.68). We also found no gender differences in self-efficacy in the tinkering study of Chapter 4. For researchers and designers concerned about gender differences, the bottom line is that no assumptions should be made regarding whether females will or will not have lower self-efficacy than the males. Even so, for females, low self-efficacy when present had more detrimental effects than for low self-efficacy males.

Features experiment Result: Males were more willing to adopt the new features: they performed significantly more Type Taught actions than females. Furthermore, significantly more males used Type Untaught features than females did.

For the type taught and untaught features, there were no statistically significant gender differences in usage (t-test: taught: t=-0.76, df=42, p<0.45; untaught: t=0.065, df=42, p<0.95). However, the relationship between self-efficacy and untaught feature usage is revealing. Figure 37 shows the suggestive relationships, and how those differ for the males and females. For the females, their self-efficacy was suggestively predictive of untaught feature usage, but for the males suggestive relationship is the opposite (linear regression: males: F(1,19)=0.41, ß=-0.08, R^2=0.02, p<0.53; females: F(1,21)=1.92, ß=0.21, R^2=0.084, p<0.18). In essence, gender differences in the use of untaught features were not confirmed for a commercial spreadsheet environment.

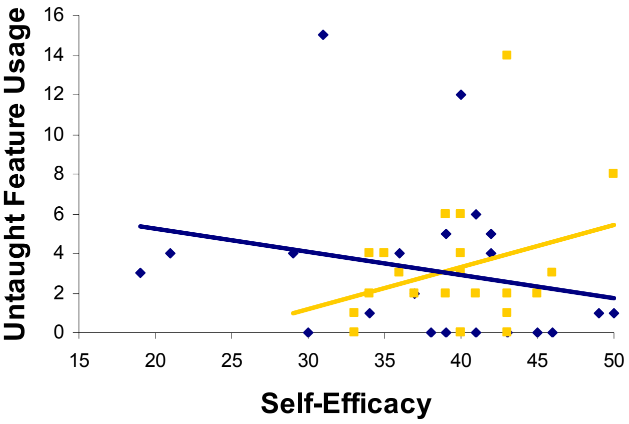


Figure 37. Self-Efficacy and Untaught Feature Usage: Notice that the suggested relationship between self-efficacy and untaught feature usage is nearly the opposite for the males (dark) and females (light).

Features experiment Result: no significant difference between the females' and males' performance in fixing seeded bugs, but the females introduced significantly more bugs than the males did.

There was no gender difference in task success: males and females scored a median of 10 and 9 points respectively (t-test: t=0.46, df=42, p<0.65). In the features experiment the gender differences in introduced bugs may be due to the females' lower self-efficacy in that study. The tinkering study of Chapter 4 study also found no gender differences in performance. These findings, in combination with our original results, amount to this: when there are no differences in self-efficacy, there is no evidence of lower task performance by female end-user programmers.

6.5 Discussion

This work has generalized one particular study. We would have liked to consider the results of the tinkering study's applicability to Excel features, but there was not enough activity on any one feature. Overall, participants used 61 different Excel features, with a median of 12 different features per participant.

A central outcome from the current study is that neither gender alone nor self-efficacy alone was a particularly useful predictor of the outcomes for this task of spreadsheet maintenance. Rather, the impact of self-efficacy on behavior was different for male than for female end-user programmers. This outcome is consistent with the studies presented in earlier chapters as well.

Because this phenomenon is consistently present in our studies, including this one showing its presence in a commercial environment, there is now significant evidence that it is real, at least for spreadsheets. To encourage other researchers interested in exploring its applicability to other end-user programming environments, Figure 38 summarizes how to repeat it.

1. Choose an environment, preferably one that supports logging (for easy data capture).
   Our study: We chose Excel.
   Alternatives for replication/generalization: Other environments.
2. Choose participants. Ensure that their experience level does not interfere with your choice of features to study.
   Our study: Real end-user developers, familiar with Excel, subject to the limitations described in Section 6.2.1.
   Alternatives for replication/generalization: End-user developers with at least some prior experience with the environment.
3. Choose a task for the participants to complete.
   Our study: Modification task, with both the original program and the modification ideas drawn from real-world spreadsheets from the EUSES Spreadsheet Corpus.
   Alternatives for replication: Different spreadsheets from the same corpus; programs drawn from some other corpus of software developed by end users.
   Alternatives for generalization: Debugging, with bugs harvested from other end users; some other end-user software development task, such as comprehending, reusing, testing, ...
4. Create a tutorial that teaches the “taught” features, briefly calls attention to the “untaught” features, and illustrates the correspondence between the usefulness of features and the task the participants are doing.
   Our study: Described in Section 6.2.3.
   Alternatives for generalization: An on-line self-study guide.
5. Create a pre-task and post-task questionnaire.
   Our study: Described in Section 6.2.5.
   Alternatives for generalization: Questionnaire could be tailored for different research goals.

Figure 38. Replication: How to replicate or generalize the experiment in other environments.

6.6 Chapter Summary

We have presented our process and subsequent results of replicating and generalizing the features experiment that first revealed gender differences in end-user programming environments. We have found that several of the results from that study generalize to the commercial environment, namely:

- Females' self-efficacy predicted task success, but the same did not hold true for the males.
- Low self-efficacy females were more engaged with the type familiar features, particularly value edits. Self-efficacy theory suggests that they may have avoided more challenging aspects of the tasks. (Males' usage did not suggest this same explanation.)
- The above results cannot be attributed to females being better judges of their weaknesses: Females' comprehension of the software features were no different than the males' and were not predicted by self-efficacy.

This is the first study in a commercial environment, but the fourth study in total, in which we have found that the effects of self-efficacy play out differently for male and female end-user programmers.

7. Conclusion

There is ample evidence that gender differences exist in the ways people solve problems. Our results show that these differences are highly relevant to users' ability to gain benefits from the features that exist in end-user programming environments. Through the series of studies presented in the last six chapters, we have shown consistent results of males and females interacting and benefiting in different ways while engaged in end-user software engineering tasks. Our results also indicate that carefully designed features in end-user programming environments can encourage females to engage with features they may have otherwise avoided. Likewise, these same features also, in our tinkering study, dissuaded males from engaging in the same level of unproductive tinkering.

Our research also opens many new research questions about what kinds of effects the supportive features, such as our strategy hints, have on males and females during their problem solving. Our research has addressed two (SE-1 and R-1 from Table 2-3) of the hypotheses of Chapter 2 in depth, scratching only the surface of the others. The hypotheses provide a jumping off point for other researchers, who we encourage to explore some of these hypotheses.

Recall "Ashley" from the introduction. In fact, "Ashley" is a male. His story is true. Ashley went on to a college career in art, and ultimately won the most prestigious academic award his university bestows. He enjoys art, but regrets his decision not to pursue graphic design. Now that he has graduated, the barriers to making the switch back to graphic design are even higher, because he no longer has access to the educational support structures available to students. Still, at home after work, he is working to overcome the barriers that prevent information technology from being a good fit to his strengths.

Although gender differences in self-efficacy, motivations, problem-solving styles, learning styles, and information processing styles are all implicated in the studies we conducted, it is important to remember that no single female is likely to have every trait statistically associated with females, nor is any single male likely to have every trait statistically associated with males. For example, some males process information in the comprehensive style statistically associated with females, and some females process information in the more linear style associated with males. Thus, designing software in ways that support these differences does not penalize either gender—it helps everyone.

Bibliography

[Abraham and Erwig 2007] Abraham, R. and Erwig, M. U-check: A spreadsheet unit checker for end users. *Journal of Visual Languages and Computing 18*(1), 2007, 71-95.

[Allwood 1984] Allwood, C. Error detection processes in statistical problem solving. *Cognitive Science 8*(4), 1984, 413-437.

[Ames 2003] Ames, P. Gender and learning styles interactions in student's computer attitudes. *Journal of Educational Computing Research 28*(3), 2003, 231-244.

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

[Arnone and Small 1995] Arnone, M. P. and Small, R. V. Arousing and sustaining curiosity: Lessons from the arcs model. *In Proc. of Annual National Convention of the Association for Educational Communications and Technology*. 1995.

[Arroyo 2003] Arroyo, I. Quantitative evaluation of gender differences, cognitive development differences and software effectiveness for an elementary mathematics intelligent tutoring system. Ph.D. Thesis, Univ. Mass. Amherst, 2003.

[Bandura 1977] Bandura, A. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review 8*(2), 1977, 191-215.

[Bandura 1986] Bandura, A. *Social Foundations of Thought and Action: A social Cognitive Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1986.

[Barke et al. 1997] Barke, R., Jenkins-Smith, H. and Slovic, P. Risk perceptions of men and women scientists. *Social Science Quarterly 78*(1), 1997, 167-176.

[Bauer 1960] Bauer, R. A. Consumer behavior as risk taking. In Cox, D. F. (Eds.) *Risk Taking and Information Handling in Consumer Behavior*. Harvard University Press, Cambridge, MA, 1960, 389-398.

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

[Beckwith et al. 2005a] Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S. and Hastings, M. Effectiveness of end-user debugging software features: Are there gender issues? . *In Proc. of ACM Conference on Human Factors in Computing Systems (CHI'05)*. 2005, 869-878.

[Beckwith et al. 2005b] Beckwith, L., Chintakovid, T., Wiedenbeck, S. and Burnett, M. Mining qualitative behavioral data from quantitative data: A case study from the gender HCI project. *In Proc. of Psychology of Programming Interest Group*. 2005.

[Beckwith et al. 2005c] Beckwith, L., Sorte, S., Burnett, M., Wiedenbeck, S., Chintakovid, T. and Cook, C. Designing features for both genders in end-user programming environments. *In Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing Languages and Environments*. 2005, 153-160.

[Beckwith et al. 2006] Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A. and Cook, C. Tinkering and gender in end-user programmers' debugging. *In Proc. of ACM Conference on Human-Computer Interaction (CHI'06)*. 2006, 231-240.

[Beckwith et al. 2007] Beckwith, L., Grigoreanu, V., Subrahmaniyan, N., Wiedenbeck, S., Burnett, M., Cook, C., Bucht, K. and Drummond, R. Gender differences in end-user debugging strategies. Oregon State University: CS07-60-01, 2007. http://eecs.oregonstate.edu/library/files/2007-10/StrategiesPaper-TechReport.pdf.

[Bennett et al. 2004] Bennett, D., Brunner, C., McCermott, M. and Greene, L. Designing for diversity: Investigating electronic games as pathways for girls into information technology professions. *In Proc. of NSF ITWF & ITR/EWF Principal Investigator Conference*. 2004, 16-21.

[Beyer and Bowden 1997] Beyer, S. and Bowden, E. M. Gender differences in self-perceptions: Convergent evidence from three measures of accuracy and bias. *Personality and Social Psychology Bulletin 23*(2), 1997, 157-172.

[Beyer et al. 2003] Beyer, S., Rynes, K., Perrault, J., Hay, K. and Haller, S. Gender Differences in Computer Science Students. *In Proc. of SIGCSE: Special Interest Group on Computer Science Education*. ACM, 2003, 49-53.

[Blackwell 2002] Blackwell, A. First steps in programming: A rationale for attention investment models. *In Proc. of IEEE Human-Centric Computing Languages and Environments*. 2002, 2-10.

[Blais and Weber 2001] Blais, A.-R. and Weber, E. U. Domain-specificity and gender differences in decision making. *Risk Decision and Policy 6*, 2001, 47-69.

[Bransford 1999] Bransford, J. D., Brown, A. L. and Cocking, R. R. *How People Learn: Brain, Mind, Experience, and School*, National Academy Press, Washington DC, 1999.

[Brosnan 1998] Brosnan, M. *Technophobia: The Psychological Impact of Information Technology*, Routledge, London and New York, 1998.

[Brosnan and Lee 1998] Brosnan, M. and Lee, W. A cross-cultural comparison of gender differences in computer attitudes and anxieties: The United Kingdom and Hong Kong. *Computers in Human Behavior 14*(4), 1998, 559-577.

[Brunner et al. 1998] Brunner, C., Bennett, D. and Honey, M. Girl games and technological desire. In Cassell, J. and Jenkins, H. (Eds.) *From Barbie to Mortal Kombat: Gender and Computer Games*. MIT Press, Cambridge, MA, 1998, 72-88.

[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. 2003] Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J. and Wallace, C. End-user software engineering with assertions in the spreadsheet paradigm. *In Proc. of International Conference on Software Engineering*. 2003, 93-103.

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

[Busch 1995] Busch, T. Gender differences in self-efficacy and attitudes toward computers. *Journal of Educational Computing Research 12*, 1995, 147-158.

[Busch 1996] Busch, T. Gender, group composition, cooperation, and self-efficacy in computer studies. *Journal of Educational Computing Research 15*(2), 1996, 125-135.

[Byrnes et al. 1999] Byrnes, J. P., Miller, D. C. and Schafer, W. D. Gender differences in risk taking: A meta-analysis. *Psychology Bulletin 125*(3), 1999, 367-383.

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

[Carroll 1998] Carroll, J. M. (Ed.) *Minimalism Beyond "The Nurnberg Funnel"*. MIT Press, Cambridge, MA, 1998.

[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.

[Clance and Imes 1978] Clance, P. R. and Imes, S. The imposter phenomenon in high achieving women: Dynamics and therapeutic intervention. *Psychotherapy Theory, Research and Practice 15*(3), 1978.

[Colley and Comber 2003] Colley, A. and Comber, C. Age and gender differences in computer use and attitudes among secondary school students: What has changed? *Educational Research 45*(2), 2003, 155-165.

[Compeau and Higgins 1995] Compeau, D. and Higgins, C. Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly 19*(2), 1995, 189-211.

[Corston and Colman 1996] Corston, R. and Colman, A. M. Gender and social facilitation effects on computer competence and attitudes toward computers. *Journal of Educational Computing Research 14*(2), 1996, 171-183.

[Czerwinski et al. 2002] Czerwinski, M., Tan, D. and Robertson, G. G. Women take a wider view. *In Proc. of ACM Conference on Human-Computer Interaction (CHI'02)*. 2002, 195-202.

[Davis 1989] Davis, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly 13*(3), 1989, 319-340.

[Durndell and Haag 2002] Durndell, A. and Haag, Z. Computer self-efficacy, computer anxiety, attitudes toward the Internet and reported experience with the Internet, by gender, in an East European sample. *Computers in Human Behavior 18*, 2002, 521-535.

[Durndell et al. 2000] Durndell, A., Haag, Z. and Laithwaite, H. Computer self-efficacy and gender: A cross cultural study of Scotland and Romania. *Personality and Individual Differences 28*, 2000, 1037-1044.

[Erwig et al. 2006] Erwig, M., Abraham, R., Cooperstein, I. and Kollmansberger, S. Gencel: A program generator for correct spreadsheets. *Journal of Functional Programming 16*(3), 2006, 293-325.

[Fallows 2005] Fallows, D. How women and men use the internet. *Pew Internet*, 2005.

[Featherman and Fuller 2003] Featherman, M. and Fuller, M. Applying TAM to e-services adoption: The moderating role of perceived risk. *In Proc. of Hawaii International Conference on System Sciences*. IEEE, 2003.

[Finucane et al. 2000] Finucane, M. L., Slovic, P., Mertz, C. K., Flynn, J. and Satterfield, T. A. Gender, race, and perceived risk: The 'white male' effect. *Health, Risk, & Society 2*(2), 2000, 159-172.

[Fisher II et al. 2002] Fisher II, M., Cao, M., Rothermel, G., Cook, C. and Burnett, M. Automated test case generation for spreadsheets. *In Proc. of Int'l. Conf. on Software Engineering*. 2002, 141-151.

[Fisher II and Rothermel 2005] Fisher II, M. and Rothermel, G. The EUSES spreadsheet corpus: A shared resource for supporting experimentation with spreadsheet dependability mechanism. *In Proc. of WEUSE05: 1st Workshop on End-User Software Engineering*. 2005, 47-51.

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

[Green and Petre 1996] Green, T. R. G. and Petre, M. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages and Computing 7*(2), 1996, 131-174.

[Gustafson 1998] Gustafson, P. E. Gender differences in risk perception: Theoretical and methodological perspective. *Risk Analysis 18*(6), 1998, 805-811.

[Hargittai and Shafer 2006] Hargittai, E. and Shafer, S. Differences in actual and perceived online skills: The role of gender. *Social Science Quarterly 87*(2), 2006, 432-448.

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

[Hou 2006] Hou, W., Kaur, M., Komlodi, A., Lutters, W. G., Boot, L., Cotton, S. R., Morrell, C., Ozok, A. A. and Tufekci, Z. "Girls don't waste time": Pre-adolescent attitudes toward ict. *In Proc. of ACM Conference on Human Factors in Computing Systems*. ACM Press, 2006, 875-880.

[Hudgen and Fatkin 2001] Hudgens, G. A. and Fatkin, L. T. Sex differences in risk taking: Repeated sessions on a computer-simulated task. *The Journal of Psychology 119*(3), 2001, 197-206.

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

[Jianakoplos and Bernasek 1998] Jianakoplos, N. A. and Bernasek, A. Are women more risk averse? *Economic Inquiry 36*, 1998, 620-630.

[Jones et al. 2000] Jones, M. G., Brader-Araje, L., Carboni, L. W., Carter, G., Rua, M. J., Banilower, E. and Hatch, H. Tool time: Gender and students' use of tools, control, and authority. *Journal of Research in Science Teaching 37*(8), 2000, 760-783.

[Karraker et al. 1995] Karraker, K. H., Vogel, D. A. and Lake, M. A. Parents' gender-stereotyped perceptions of newborns: The eye of the beholder revisited. *Sex Roles 33*(9-10), 1995, 687-701.

[Kelleher et al. 2007] Kelleher, C., Pausch, R. and Kiesler, S. Storytelling alice motivates middle school girls to learn computer programming. *In Proc. of ACM Conference on Human-Computer Interaction*. 2007 (to appear).

[Kissinger et al. 2006] Kissinger, C., Burnett, M., Stumpf, S., Subrahmaniyan, N., Beckwith, L., Yang, S. and Rosson, M. B. Supporting end-user debugging: What do users want to know? *In Proc. of Advanced Visual Interfaces*. ACM Press, 2006, 135-142.

[Ko and Myers 2006] Ko, A. J. and Myers, B. A. Barista: An implementation framework for enabling new tools, interaction techniques and views for code editors. *In Proc. of ACM Conference on Human Factors in Computing Systems*. 2006, 387-396.

[Ko and Myers 2004] Ko, A. J. and Myers, B. A. Designing the whyline: A debugging interface for asking questions about program failures. *In Proc. of ACM Conference on Human Factors in Computing Systems*. 2004, 151-158.

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

[Krishna 2002] Krishna, V. B. Empirical Studies of a Spreadsheet Maintenance Environment. Masters Thesis, Oregon State University, 2002.

[Lepper and Malone 1987] Lepper, M. R. and Malone, T. W. Intrinsic motivation and instructional effectiveness in computer-based education. In Snow, R. E. and Farr, M. J. (Eds.) *Aptitude, learning, and instruction: Vol. 3. Conative and affective process analyses*. Lawrence Erlbaum, Hillsdale, NJ, 1987, 255-286.

[Loewenstein 1994] Loewenstein, G. The psychology of curiosity: A review and reinterpretation. *Psychology Bulletin 116*(1), 1994, 75-98.

[Lunderberg et al. 2004] Lunderberg, M., Fox, P. and Punchochar, J. Highly confidence but wrong: Gender differences and similarities in confidence judgments. *Journal of Educational Psychology 86*(1), 1994, 114-121.

[Malone and Lepper 1987] Malone, T. W. and Lepper, M. R. Making learning fun: A taxonomy of intrinsic motivations for learning. In Snow, R. E. and Farr, M. J. (Eds.) *Aptitude, learning and instruction. Volume 3: Conative and affective process analysis*. Lawrence Erlbaum, Hillsdale, NJ, 1987, 223-253.

[Margolis and Fisher 2003] Margolis, J. and Fisher, A. *Unlocking the clubhouse*, The MIT Press, Cambridge, MA, 2003.

[Margolis et al. 1999] Margolis, J., Fisher, A. and Miller, F. Caring about connections: Gender and computing. *IEEE Technology and Society Magazine 18*(4), 1999, 13-20.

[Martinson 2005] Martinson, A. M. Playing with technology: Designing gender sensitive games to close the gender gap. *Working Paper SLISWP-03-05, School of Library and Information Science, Indiana University*. 2005. http://www.slis.indiana.edu/research/working_papers/files/SLISWP-03-05.pdf, Accessed: September 12, 2005.

[Martocchio and Webster 1992] Martocchio, J. J. and Webster, J. Effects of feedback and playfulness on performance in microcomputer software training. *Personnel Psychology 42*, 1992, 553-578.

[Maykut and Morehouse 1994] Maykut, P. and Morehouse, R. *Beginning Qualitative Research*, The Falmer Press, London, 1994.

[McCoy et al. 2001] McCoy, L. P., Heafner, T. L., Burdick, M. G. and Nagle, L. N. Gender differences in computer use and attitudes on a ubiquitous computing campus. *In Proc. of AERA Annual Meeting* 2001.

[McIlroy et al. 2001] McIlroy, D., Bunting, B., Tierney, K. and Gordon, M. The relation of gender and background experience to self-reported computer anxieties and cognitions. *Computers in Human Behavior 17*, 2001, 21-33.

[Miller and Crouch 1991] Miller, C. J. and Crouch, J. G. Gender differences in problem solving: Expectancy and problem context. *Journal of Psychology 125*(3), 1991.

[Miller 1976] Miller, J. *Toward a new psychology of women*, Beacon Press, Boston, 1976.

[Merriam-Webster 2007] *Merriam-Webster online dictionary.* 2006-2007. *http://www.Merriam-webster.Com (accessed: March 18, 2007).*

[Meyers-Levy 1989] Meyers-Levy, J. Gender differences in information processing: A selectivity interpretation. In Cafferata, P. and Tybout, A. (Eds.) *Cognitive and Affective Responses to Advertising*. Lexington Books, Lexington, MA, 1989, 219-260.

[Meyers-Levy and Sternthal 1991] Meyers-Levy, J. and Sternthal, B. Gender differences in the use of message cues and judgments. *Journal of Marketing Research 28*, 1991, 84-96.

[Nardi 1993] Nardi, B. *A Small Matter of Programming: Perspectives on End-User Computing*, MIT Press, Cambridge, MA, 1993.

[Norman 1988] Norman, D. A. *The Design Of Everyday Things*, Basic Books, New York, 1988.

[O'Donnell and Johnson 2001] O'Donnell, E. and Johnson, E. N. Gender effects on processing effort during analytical procedures. *International Journal of Auditing 5*, 2001, 91-105.

[Pajares 2002] Pajares, F. Gender and perceived self-efficacy in self-regulated learning. *Theory Into Practice 41*(2), 2002, 116-125.

[Pane et al. 2002] Pane, J., Myers, B. A. and Miller, L. B. Using hci techniques to design a more usable programming system. *In Proc. of IEEE 2002 Symposia on Human Centric Computing Languages and Environments*. 2002, 198-206.

[Pane et al. 2001] Pane, J., Ratanamahatan, C. and Myers, B. A. Studying the language and structure in non-programmers' solutions to programming problems. *International Journal Human-Computer Studies 54*(2), 2001, 237-264.

[Panko 1998] Panko, R. What we know about spreadsheet errors. *Journal of End User Computing 10*(2), 1998, 15-21.

[Pulford and Colman 1997] Pulford, B. and Colman, A. Overconfidence: Feedback and item difficulty effects. *Journal of Personality and Individual Differences 23*(1), 1997, 125-133.

[Robertson et al. 2004] Robertson, T. J., Prabhakararao, S., Burnett, M., Cook, C., Ruthruff, J. R., Beckwith, L. and Phalgune, A. Impact of interruption style on end-user debugging. *In Proc. of CHI*. ACM Press, 2004, 287-294.

[Rode et al. 2004] Rode, J. A., Toye, E. F. and Blackwell, A. The fuzzy felt ethnography - understanding the programming patterns of domestic appliances. *In Proc. of 2nd International Conference on Appliance Design*. 2004, 10-22.

[Rosson et al. 2004] Rosson, M. B., Ballin, J. and Nash, H. Everyday programming: Challenges and opportunities for informal web development. *In Proc. of Visual Languages and Human-Centric Computing*. IEEE, 2004, 123-130.

[Rosson et al. 1990] Rosson, M. B., Carroll, J. M. and Bellamy, R. K. E. Smalltalk scaffolding: A case study of minimalist instruction. *In Proc. of ACM Conference on Human-Computer Interaction*. 1990, 423-429.

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

[Rowe 1978] Rowe, M. B. *Teaching Science as Continuous Inquiry: A Basic*, McGraw-Hill, New York, NY, 1978.

[Rowell et al. 2003] Rowell, G. H., Perhac, D. G., Hankins, J. A., Parker, B. C., Pettey, C. C. and Iriarte-Gross, J. M. Computer-related gender differences. *In Proc. of SIG Computer Science Education*. ACM Press, 2003, 54-58.

[Ruthruff et al. 2005] Ruthruff, J. R., Burnett, M. and Rothermel, G. An empirical study of fault localization for end-user programmers. *In Proc. of International Conference on Software Engineering*. 2005, 252-361.

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

[Shashaani and Khalili 2001] Shashaani, L. and Khalili, A. Gender and computers: Similarities and differences in Iranian college students' attitudes toward computers. *Computers & Education 37*, 2001, 363-375.

[Simon 1973] Simon, H. A. The structure of ill-structured problems. *Artificial Intelligence 4*, 1973, 181-202.

[Simon 2001] Simon, S. J. The impact of culture and gender on web sites: An empirical study. *The DATA BASE for Advances in Information Systems 32*(1), 2001, 18-37.

[Spotts et al. 1997] Spotts, T. H., Bowman, Mary Ann, and Mertz, Christopher Gender and use of instructional technologies: A study of university faculty. *Higher Education 34*(4), 1997, 431-436.

[Stipek and Gralinski 1991] Stipek, D. J. and Gralinski, J. H. Gender differences in children's achievement-related beliefs and emotional responses to success and failure in mathematics. *Journal of Educational Psychology 83*(3), 1991, 361-371.

[Subrahmaniyan et al. 2007] Subrahmaniyan, N., Kissinger, C., Rector, K., Inman, D., Kaplan, J., Beckwith, L. and Burnett, M. Explaining debugging strategies to end-user programmers. Oregon State University:CS07-60-03, 2007. http://eecs.oregonstate.edu/ library/files/2007-17/VLHCC_ExplanationTechReport.pdf

[Tan et al. 2003] Tan, D., Czerwinski, M. and Robertson, G. G. Women go with the (optical) flow. *In Proc. of CHI 2003*. ACM Press, 2003, 209-215.

[Teasdal and Lupart 2001] Teasdale, S. and Lupart, J. L. Gender differences in computer attitudes, skills, and perceived ability. *In Proc. of Canadian Society for Studies in Education*. 2001.

[Tillberg and Cohoon 2005] Tillberg, H. and Cohoon, J. M. Attracting women to the CS major. *Frontiers: A Journal of Women Studies 26*(1), 2005, 126-140.

[Torkzadeh and Koufteros 1994] Torkzadeh, G. and Koufteros, X. Factorial validity of a computer self-efficacy scale and the impact of computer training. *Educational and Psychological Measurement 54*(3), 1994, 813-821.

[Van Den Heuvel-Panheizen 1999] Van Den Heuvel-Panheizen, M. Girls' and boys' problems: Gender differences in solving problems in primary school mathematics in the Netherlands. In Nunes, T. and Bryant, P. (Eds.) *Learning And Teaching Mathematics: An International Perspective*. Psychology Press, UK, 1999, 223-253.

[Venkatesh and Morris 2000] Venkatesh, V. and Morris, M. G. Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior. *MIS Quarterly 24*(1), 2000, 115-139.

[Venkatesh et al. 2003] Venkatesh, V., Morris, M. G., Davis, G. B. and Davis, F. D. User acceptance of information technology: Toward a unified view. *MIS Quarterly 27*(3), 2003, 425-478.

[Vermeer et al. 2000] Vermeer, H. J., Boekaerts, M. and Seegers, G. Motivational and gender differences: Sixth-grade students' mathematical problem-solving behavior. *Journal of Educational Psychology 92*(2), 2000, 308-315.

[Yong 2007] Young, V. 10 Steps to Overcome the Impostor Syndrome. 2006. http://www.impostorsyndrome.com/overcome.htm Accessed: February 12, 2007

[Webster and Martocchio 1993] Webster, J. and Martocchio, J. J. Turning work into play: Implications for microcomputer software training. *Journal of Management 19*(1), 1993, 127-146.

[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. *In Proc. of ACM Conference on Human Factors in Computing Systems*. 2003, 305–312.

[Zeldin and Pajares 2000] Zeldin, A. L. and Pajares, F. Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers. *American Educational Research Journal 37*, 2000, 215-246.

Appendices

Appendix A: Chapter 3 Study Materials

Background Questionnaire

1. Gender (circle your selection): Male / Female
2. Age < 20 20 – 30 30 – 40 40 – 50 50 – 60 >60
3. Major or Educational Background: ______________________
4. Year or Degree Completed: Fresh. Soph. Jun. Sen. Post Bac. Grad.
5. Cumulative GPA: ______________________
6. Do you have previous programming experience?
   1. High school:
      - How many courses?
      - What programming languages?
   2. College:
      - How many courses?
      - What programming languages?
   3. Professional and/or recreational
      - How many years?
      - What programming languages?
7. Have you ever created a spreadsheet for (please check all that apply):

   □ A high school course How many? ____________

   □ A college course How many? ____________

   □ Professional use How many years? ____________

   □ Personal use How many years? ____________
8. Have you participated in any previous Forms/3 experiments? Yes / No
9. Is English your primary language? Yes / No

   If not, how long have you been speaking English? ______ years.

The following questions ask you to indicate whether you could use a new spreadsheet system under a variety of conditions. For each of the conditions please indicate whether you think you would be able to complete the job using the system.

Given a spreadsheet which performs common tasks (such as calculating course grades or payroll) I could find and fix errors:

... if there was no one around to tell me what to do as I go.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had never used a spreadsheet like it before.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had only the software manuals for references.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had seen someone else using it before trying it myself.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I could call someone for help if I got stuck.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if someone else had helped me get started.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had a lot of time to complete the task.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had just the built-in help facility for assistance.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if someone showed me how to do it first.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
... if I had used similar spreadsheets before this one to do this same task.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

Post-session Questionnaire (Gradebook)

I. Circle the answer corresponding to how much you agree or disagree with the following statements.

1. I am confident that I found all the bugs in the Gradebook spreadsheet? (circle one)

   Strongly Disagree   Disagree   Neither Agree Nor Disagree   Agree   Strongly Agree

2. I am confident that I fixed all the bugs in the Gradebook spreadsheet? (circle one)

   Strongly Disagree   Disagree   Neither Agree Nor Disagree   Agree   Strongly Agree

3. How much additional time would you need to complete this task?

_____ None. It only took me _____ minutes.
_____ None. I took about the entire time.
_____ I would need about _____ more minutes.
_____ I am not sure.

4a. Mark how you found the following features for **finding and fixing errors**:

Cell border colors helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Interior Cell Coloring (yellow and red) helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
X-marks helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Checkmarks (√) helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Pop up messages helped me make	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

progress					
Arrows helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Percent tested indicator helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Bug likelihood bar helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

4b. Rank your preference for the following features (**1 – most preferred feature; 2 – 2nd most preferred feature; 3 – 3rd most preferred feature; and so on**):

_______ Cell border colors

_______ Interior cell colorings

_______ X-marks

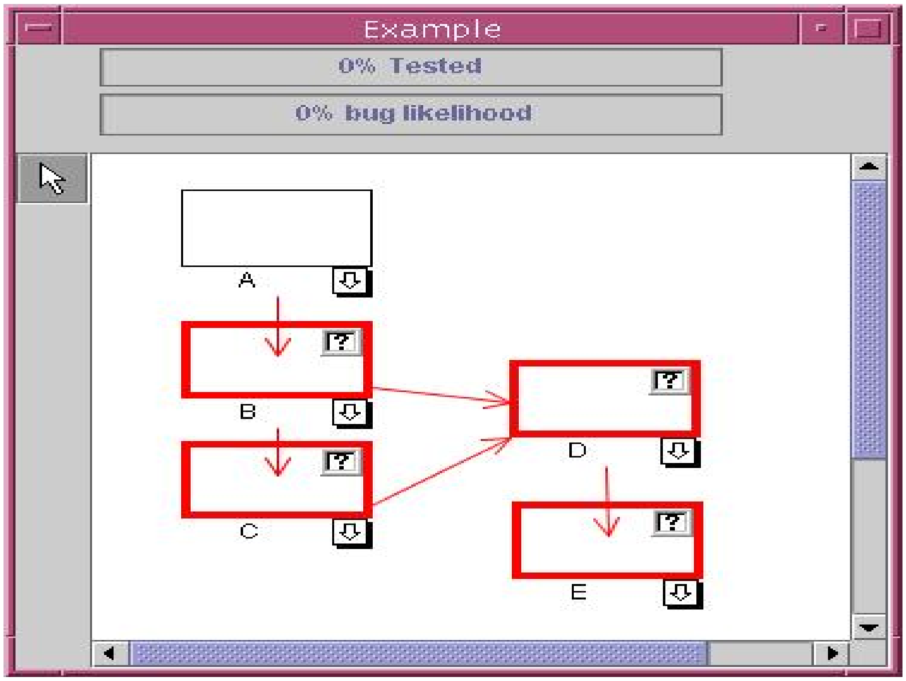
_______ Checkmarks

_______ Pop-up messages

_______ Arrows

_______ Percent testedness indicator

_______ Bug likelihood bar



Q5 to Q10: Refer to the Figure Above and choose your answers from the choices below.

One or more Questions can have the same answer.

5. If we place an X- mark in cell D the color of the cell D:
   - a. Remains the same
   - b. Gets darker
   - c. Gets lighter
   - d. Don't know

6. If we place an X- mark in cell D the color of the cell C
   - a. Remains the same
   - b. Gets darker
   - c. Gets lighter
   - d. Don't know

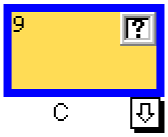
7. If we place an X- mark in cell D the color of the cell E
   - a. Remains the same
   - b. Gets darker
   - c. Gets lighter
   - d. Don't know

Assume for the next three Questions (8-10) that an X- mark has been placed on the cell D.

8. If we place an X- mark in cell C the color of the cell C
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

9. If we place an X- mark in cell C the color of the cell B
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

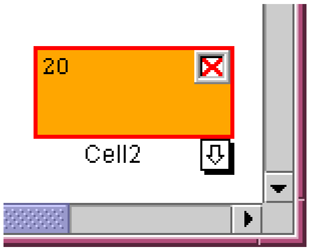
10. If we place a Checkmark in cell C the color of the cell D
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know



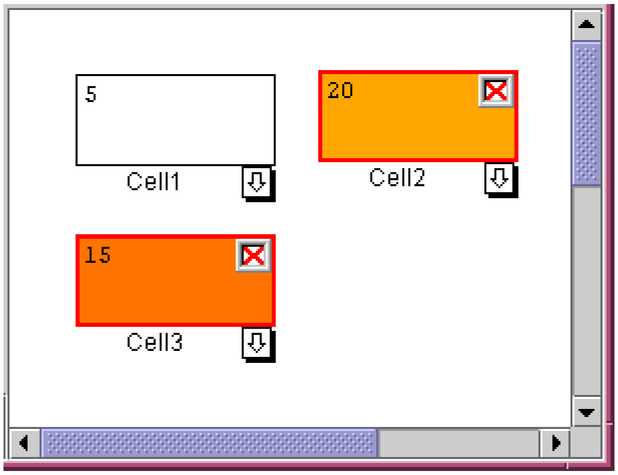
11. What does a blue border of a cell with a yellow-orange interior mean (refer to above figure)? (Circle 1 option for each part)

a) The value is: (circle 1)	CORRECT	WRONG	COULD BE EITHER
b) The cell is: (circle 1)	TESTED	UNTESTED	COULD BE EITHER
c) The cell has: (circle 1)	BUG LIKELIHOOD	NO BUG LIKELIHOOD	COULD BE EITHER
d) My answers to a, b, and c are just guesses.	YES, JUST GUESSES	NO, NOT GUESSES	
e) The combination of blue border and yellow-orange interior colors on this cell: (circle 1)	MAKES SENSE	MAKES NO SENSE	NOT SURE

12. What does the X- mark in the decision box mean?

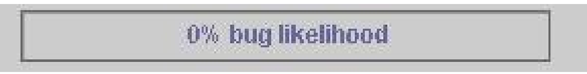


13. In the above figure what does the orange color in the interior of the cell mean?



14. In the above figure what does it mean when the colors in the interior of one cell is darker the others?

Please provide any other general comments you may have regarding the cell interior colorings:



15. In the above figure what does the bug likelihood bar mean?

Please provide any other general comments you may have regarding the bug likelihood bar:

Did you place X marks? If yes answer Question 16, otherwise answer Question 17

16. When I placed an X mark…

… the computer made bad decisions with them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried they would distract me from my original goal.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid that I would not use them properly.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… it seemed like they were causing problems with the spreadsheet.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried that they would not help achieve my goal(s).	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid I would take too long to learn them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

17. I did not place X marks because…

… the computer would make bad decisions with them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried they would distract me from my original goal.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid that I would not use them properly.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… it seemed like they could cause problems with the spreadsheet.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried that they would not help achieve my goal(s).	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid I would take too long to learn them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

18. If there are still errors in the spreadsheet this is because… (Circle **1** reason you agree with most)

a. The computer should have helped me spot the errors
b. I should have spent more time trying to find the errors
c. There was not enough time
d. None of the above

Tutorial

Hi, my name is Laura Beckwith, and I will be leading you through today’s study.

The other people involved in this study are Dr. Margaret Burnett, Dr. Curtis Cook, Shraddha Sorte, Michelle Hastings, and the assistants helping me out today.

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

Tutorial

Before we begin, I'd like to ask if anyone in here is 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.

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 couple of sample spreadsheet problems. After the tutorial, you will be given two different spreadsheets; asked to test the spreadsheets, and correct any errors you find in them.

- As we go through this tutorial, I want you to ACTUALLY PERFORM the steps I'm describing. For example, at times I will want you to click the left mouse button, at times I will want you to click the middle mouse button (the scroll button in the middle of your mouse) and at other times I will want you to click the right mouse button. 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 each 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 colored 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 *(show wave the mouse around the cell and then rest mouse on border)*. Note that a message will pop up and tell us what this color means. Can anyone tell me what the message says? *(PAUSE, look for a hand.)* Yes, it means that the cell has not been tested.

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 is just telling us that the cell has not been tested. It is up to us to make a decision about the correctness of the cells 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 do not have any special border color *(wave mouse around cells)*. Such cells without colored borders are called input cells. Cells with colored borders are called formula cells.

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 anyone tell me what the popup message says? *(PAUSE, wait for answer.)* Yes, it says that if the value of this cell is correct, we can left-click and if the value of the cell is wrong, we can right-click. It also tells us that these decisions help test and find errors.

So let's left-click the question mark in this decision box for TotalCost. Notice what happened. 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.

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 left-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 left-click in the decision box for TotalCost to put our checkmark back in that box.

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 let's 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 my 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 that 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 let's 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, my spreadsheet already has these values, so I 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 I 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 the cell's value is wrong to right-click. Well, this value is wrong, so go ahead and right-click on the question mark in this decision box.

Hey, look at that! 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". Let's right-click 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 my description, PaperTotalCost should be 1600. But our value is 2800, and that is wrong. So let's 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 let's change the 7 in this formula to a 4, and click the Apply button to finalize my changes.

Hey wait, my total spreadsheet testedness at the top of my 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. I have a new formula in this cell, so those tests are no longer valid. But, never fear, I 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 left-click to place a checkmark in the decision box for PaperTotalCost. Oh good, the percent testedness of my spreadsheet went up to 7%; I 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. So let's click in the decision box of this cell to place a checkmark.

But wait! The border of this cell is only purple. Let's rest our 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. Let's rest our 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. Let's 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 let's rest my mouse over the decision box of this PaperQCheck cell. Oh yeah, it means I'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 left-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 I make a testing decision on this cell, I 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, I have 31 paper on stock. Is this enough paper? The spreadsheet description says I need 400 reams of paper, but I only have 31. So this is not enough paper. And the PaperQCheck cell says "not enough paper". Well, this is correct, so let's left-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 let's 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 let's right-click on the question mark in PenQCheck's decision box.

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 everyone 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 we are done. (*wait a second*). Note that PenQCheck now displays the correct value. So let's go ahead and put a checkmark in this cell by left-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. 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 frontside 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 by left-clicking cell decision boxes, and use the X-marks by right-clicking decision boxes.

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

(*Task is 22 minutes*)

Spreadsheet & Descriptions

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.

If you purchase more than $1500 worth of paper and pens at one time you get a discount from the supplier of 10%.

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.

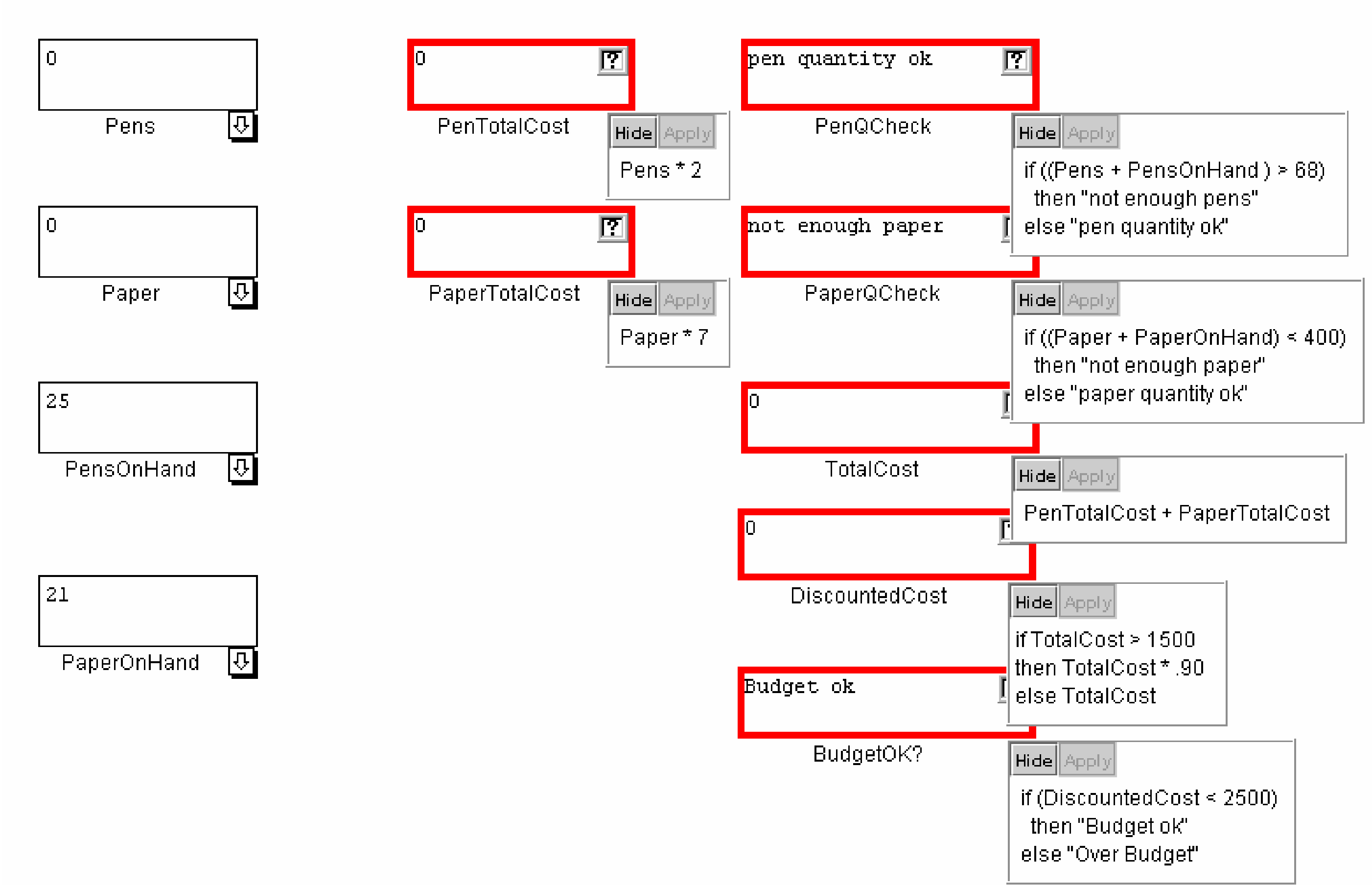
Pen and Paper Check
These cells are used to check to ensure you are ordering enough pens and paper to restock the shelves.

Total Cost
The total cost is the combined cost of pens and paper. A discount of 10% is taken if the total cost is greater than $1500. 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
DiscountedCost	1440
BudgetOK?	Budget ok

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



Pens: 0

Paper: 0

PensOnHand: 25

PaperOnHand: 21

PenTotalCost: 0

```
Pens * 2
```

PaperTotalCost: 0

```
Paper * 7
```

PenQCheck: pen quantity ok

```
if ((Pens + PensOnHand ) > 68)
  then "not enough pens"
else "pen quantity ok"
```

PaperQCheck: not enough paper

```
if ((Paper + PaperOnHand) < 400)
  then "not enough paper"
else "paper quantity ok"
```

TotalCost: 0

```
PenTotalCost + PaperTotalCost
```

DiscountedCost: 0

```
if TotalCost > 1500
then TotalCost * .90
else TotalCost
```

BudgetOK?: Budget ok

```
if (DiscountedCost < 2500)
  then "Budget ok"
else "Over Budget"
```

Figure 39. Purchase Budget Spreadsheet for Tutorial with Original Formulas

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.

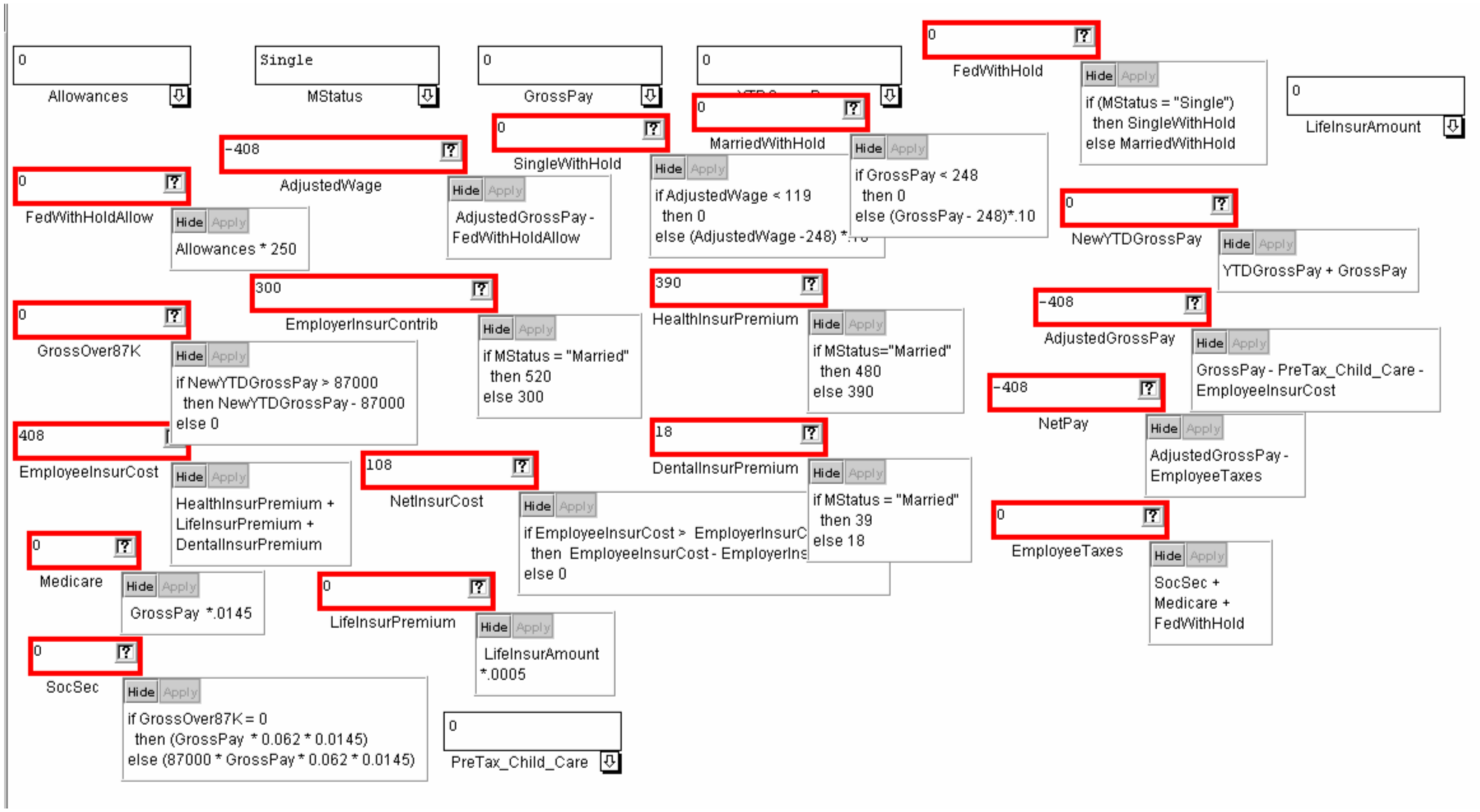
ADJUSTED GROSS PAY

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.

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	



0
Allowances

Single
MStatus

0
GrossPay

0

0
FedWithHold
Hide Apply
if (MStatus = "Single")
then SingleWithHold
else MarriedWithHold

0
LifeInsurAmount

0
MarriedWithHold
Hide Apply
if GrossPay < 248
then 0
else (GrossPay - 248)*.10

0
SingleWithHold
Hide Apply
if AdjustedWage < 119
then 0
else (AdjustedWage -248) *

-408
AdjustedWage
Hide Apply
AdjustedGrossPay -
FedWithHoldAllow

0
FedWithHoldAllow
Hide Apply
Allowances * 250

0
NewYTDGrossPay
Hide Apply
YTDGrossPay + GrossPay

300
EmployerInsurContrib
Hide Apply
if MStatus = "Married"
then 520
else 300

390
HealthInsurPremium
Hide Apply
if MStatus="Married"
then 480
else 390

-408
AdjustedGrossPay
Hide Apply
GrossPay - PreTax_Child_Care -
EmployeeInsurCost

0
GrossOver87K
Hide Apply
if NewYTDGrossPay > 87000
then NewYTDGrossPay - 87000
else 0

-408
NetPay
Hide Apply
AdjustedGrossPay -
EmployeeTaxes

408
EmployeeInsurCost
Hide Apply
HealthInsurPremium +
LifeInsurPremium +
DentalInsurPremium

108
NetInsurCost
Hide Apply
if EmployeeInsurCost > EmployerInsurC
then EmployeeInsurCost - EmployerIns
else 0

18
DentalInsurPremium
Hide Apply
if MStatus = "Married"
then 39
else 18

0
EmployeeTaxes
Hide Apply
SocSec +
Medicare +
FedWithHold

0
Medicare
Hide Apply
GrossPay *.0145

0
LifeInsurPremium
Hide Apply
LifeInsurAmount
*.0005

0
SocSec
Hide Apply
if GrossOver87K = 0
then (GrossPay * 0.062 * 0.0145)
else (87000 * GrossPay * 0.062 * 0.0145)

0
PreTax_Child_Care

Figure 40. The Payroll Spreadsheet with Original Formulas.

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 and Exams

The exam average is the average of the midterm average and the final exam.

The midterm average is the average of the third midterm and the higher of the first two midterms. The first midterm is out of 50 possible points. The second and third midterms are worth 100 points. Students achieving a non-zero grade on the third midterm receive a two point bonus. The final exam is out of 146 possible points. Exams not based on 100 points have their percents computed for later averaging.

There are five quizzes. The average is calculated on only four of these scores, dropping the lower of the first two quizzes.

Course Grade

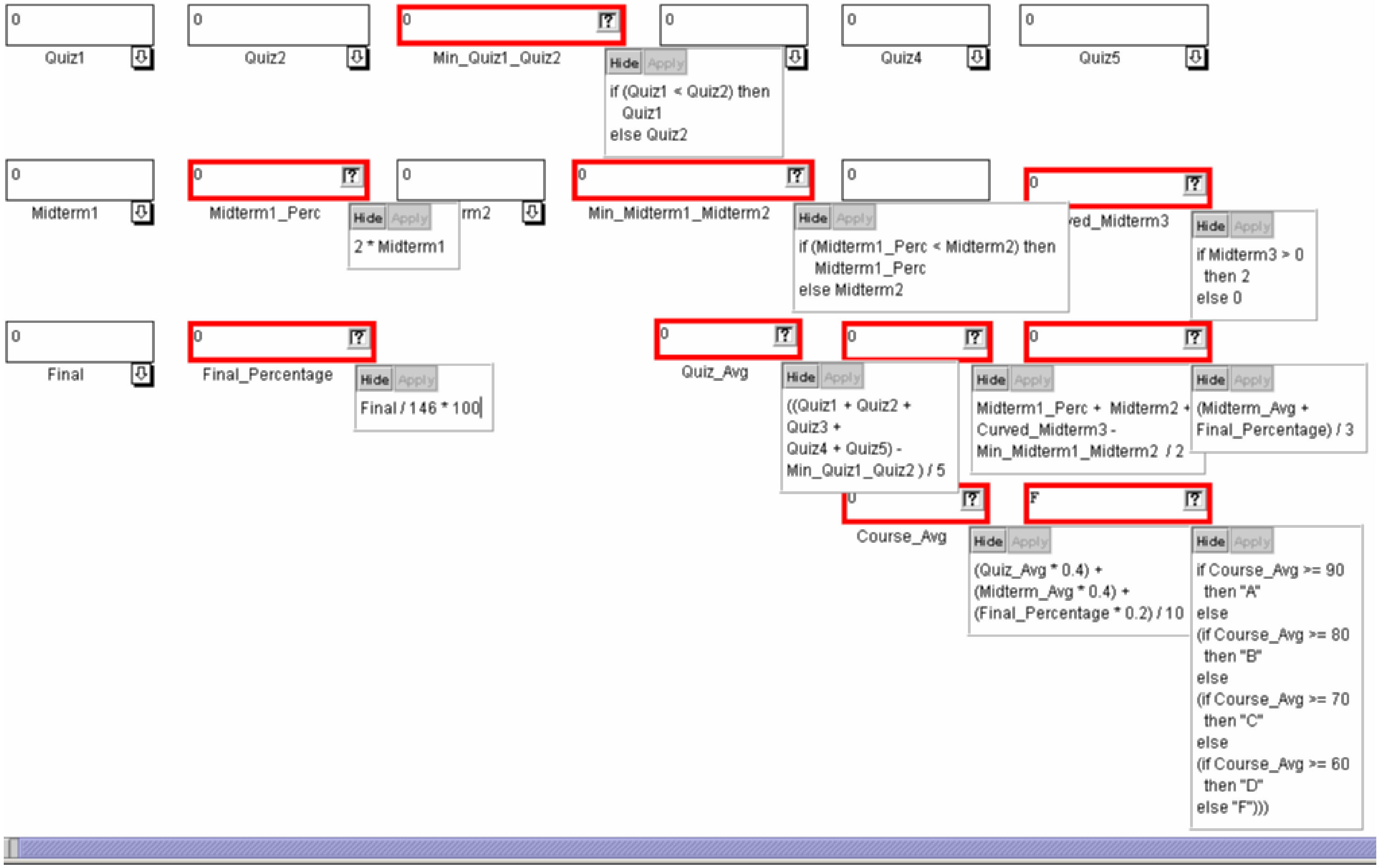
Quizzes are worth 40% of a student's grade. Midterms are worth 40% of a student's grade. The final contributes 20%. A student's course grade is determined by their course average, in accordance with the following scale:

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

Example Correct Gradebook Report Cards

John Doe	**Report Card**
Quiz1	81.25
Quiz2	100
Quiz3	100
Quiz4	96
Quiz5	100
Midterm1 (Original)	45
Midterm2	96
Midterm3 (Original)	80
Final	129
Course_Avg	92.87
Course_Grade	A

Mary Smith	**Report Card**
Quiz1	0
Quiz2	88.24
Quiz3	85
Quiz4	87
Quiz5	100
Midterm1 (Original)	24
Midterm2	61
Midterm3 (Original)	66
Final	106
Final_Percentage	
Course_Avg	76.34
Course_Grade	C



```
if (Quiz1 < Quiz2) then
  Quiz1
else Quiz2
```

```
2 * Midterm1
```

```
if (Midterm1_Perc < Midterm2) then
   Midterm1_Perc
else Midterm2
```

```
if Midterm3 > 0
  then 2
else 0
```

```
Final / 146 * 100
```

```
((Quiz1 + Quiz2 +
Quiz3 +
Quiz4 + Quiz5) -
Min_Quiz1_Quiz2 ) / 5
```

```
Midterm1_Perc + Midterm2 +
Curved_Midterm3 -
Min_Midterm1_Midterm2 / 2
```

```
(Midterm_Avg +
Final_Percentage) / 3
```

```
(Quiz_Avg * 0.4) +
(Midterm_Avg * 0.4) +
(Final_Percentage * 0.2) / 10
```

```
if Course_Avg >= 90
  then "A"
else
(if Course_Avg >= 80
  then "B"
else
(if Course_Avg >= 70
  then "C"
else
(if Course_Avg >= 60
  then "D"
else "F")))
```

Figure 41. Gradebook Spreadsheet with Original Formulas.

Appendix B: Chapter 4 Study Materials

(Pre-Session Questionnaire for both groups, and post-session questionnaire the same for low-cost treatment as Appendix A)

Post-session Questionnaire (Gradebook)

Circle the answer corresponding to how much you agree or disagree with the following statements.

1. I am confident that I found all the bugs in the Gradebook spreadsheet? (circle one)

Strongly	Disagree	Neither Agree	Agree	Strongly
Disagree		Nor Disagree		Agree

2. I am confident that I fixed all the bugs in the Gradebook spreadsheet? (circle one)

Strongly	Disagree	Neither Agree	Agree	Strongly
Disagree		Nor Disagree		Agree

3. How much additional time would you need to complete this task?

_____ None. It only took me _____ minutes.
_____ None. I took about the entire time.
_____ I would need about _____ more minutes.
_____ I am not sure.

4. If there are still errors in the spreadsheet this is because… (Circle **1** reason you agree with most)

a. The computer should have helped me spot the errors
b. I should have spent more time trying to find the errors
c. There was not enough time
d. None of the above

5. Mark how you found the following features for **finding and fixing errors**:

Cell border colors helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Interior Cell Coloring (yellow and red) helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
X-marks helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Checkmarks (√) helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Pop up messages helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Arrows helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Percent tested indicator helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
Bug likelihood bar helped me make progress	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

5a. Rank your preference for the following features (**1 – most preferred feature; 2 – 2nd most preferred feature; 3 – 3rd most preferred feature; and so on**):

________ Cell border colors

________ Interior cell colorings

________ X-marks

________ Checkmarks

________ Pop-up messages

________ Arrows

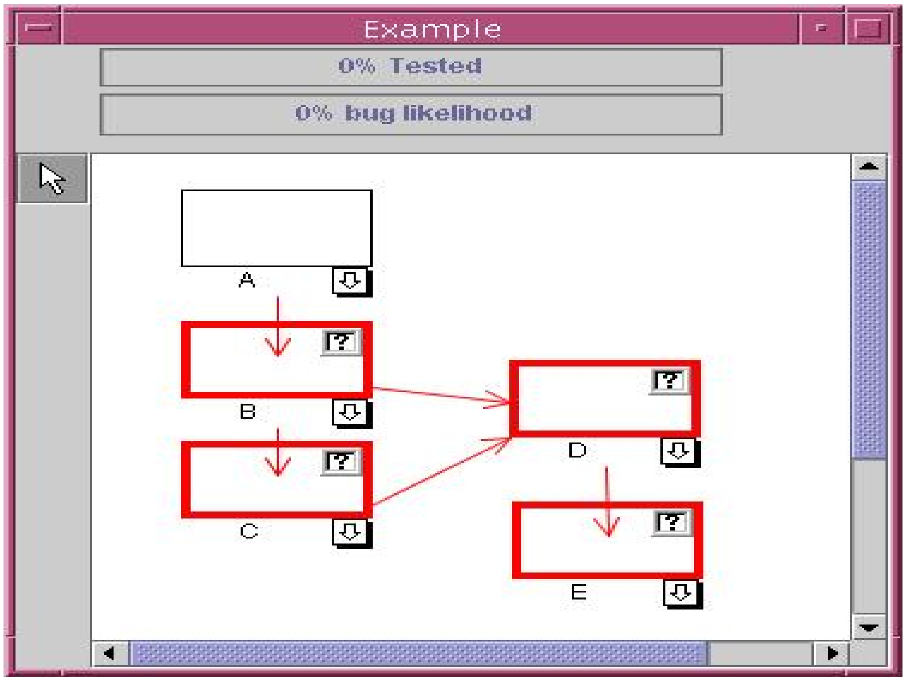
________ Percent testedness indicator

________ Bug likelihood bar

5b. Did you use the “help me test” button? (circle one)
YES
NO

5c. Rate the following statement regarding the **“help me test” button**:

I could get my spreadsheet tested without this feature	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree



Q6 to Q11: Refer to the Figure Above and choose your answers from the choices below.
One or more Questions can have the same answer.

6. If we place an X- mark in cell D the color of the cell D:
a. Remains the same
b. Gets darker
c. Gets lighter
d. Don’t know

7. If we place an X- mark in cell D the color of the cell C
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

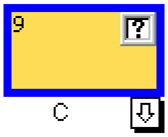
8. If we place an X- mark in cell D the color of the cell E
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

Assume for the next three Questions (8-10) that an X- mark has been placed on the cell D.

9. If we place an X- mark in cell C the color of the cell C
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

10. If we place an X- mark in cell C the color of the cell B
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know

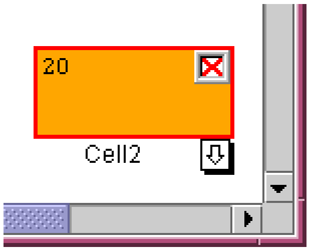
11. If we place a Checkmark in cell C the color of the cell D
   a. Remains the same
   b. Gets darker
   c. Gets lighter
   d. Don't know



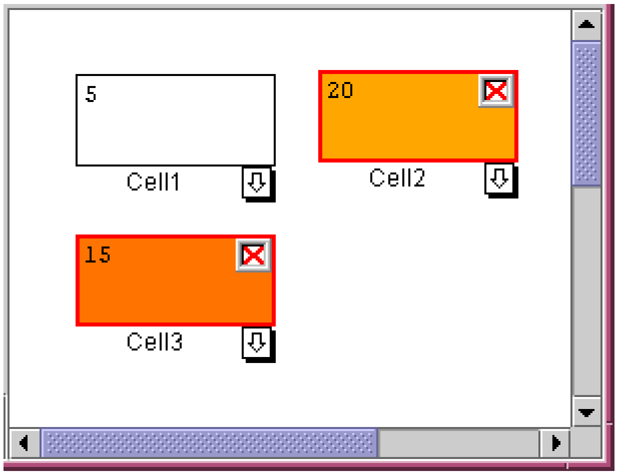
12. What does a blue border of a cell with a yellow-orange interior mean (refer to above figure)? (Circle 1 option for each part)

a) The value is: (circle 1)	CORRECT	WRONG	COULD BE EITHER
b) The cell is: (circle 1)	TESTED	UNTESTED	COULD BE EITHER
c) The cell has: (circle 1)	BUG LIKELIHOOD	NO BUG LIKELIHOOD	COULD BE EITHER
d) My answers to a, b, and c are just guesses.	YES, JUST GUESSES	NO, NOT GUESSES	
e) The combination of blue border and yellow-orange interior colors on this cell: (circle 1)	MAKES SENSE	MAKES NO SENSE	NOT SURE

13. What does the X- mark in the decision box mean?



14. In the above figure what does the orange color in the interior of the cell mean?



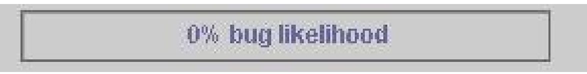
15. In the above figure what does it mean when the colors in the interior of one cell is darker than others?

Please provide any other general comments you may have regarding the cell interior colorings :

____

____

____



16. In the above figure what does the bug likelihood bar mean?

Please provide any other general comments you may have regarding the bug likelihood bar :

____

____

____

Did you place X marks? If yes answer Question 17, otherwise answer Question 18

17. When I placed an X mark…

… the computer made bad decisions with them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried they would distract me from my original goal.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid that I would not use them properly.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… it seemed like they were causing problems with the spreadsheet.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried that they would not help achieve my goal(s).	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid I would take too long to learn them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

18. I did not place X marks because…

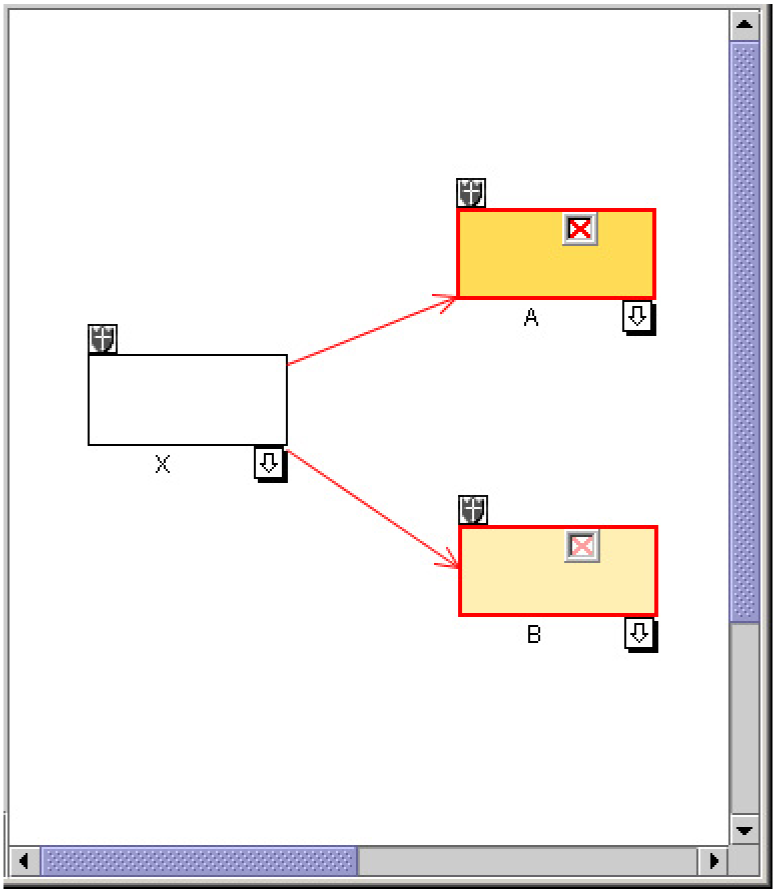
… the computer would make bad decisions with them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried they would distract me from my original goal.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid that I would not use them properly.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… it seemed like they could cause problems with the spreadsheet.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I worried that they would not help achieve my goal(s).	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
… I was afraid I would take too long to learn them.	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

19. If there are still errors in the spreadsheet this is because… (Circle **1** reason you agree with most)

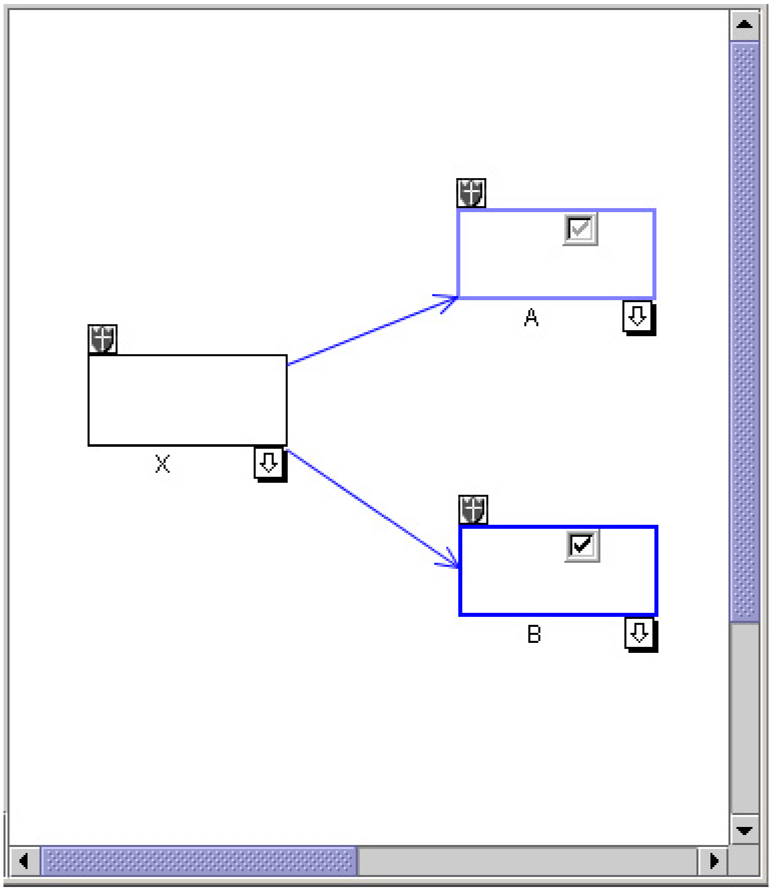
a. The computer should have helped me spot the errors
b. I should have spent more time trying to find the errors
c. There was not enough time
d. None of the above

20. would use the following marks (Circle all that apply)

☒	Very unlikely	Unlikely	Maybe or maybe not	Likely	Very likely
☒	Very unlikely	Unlikely	Maybe or maybe not	Likely	Very likely
☑	Very unlikely	Unlikely	Maybe or maybe not	Likely	Very likely
☑	Very unlikely	Unlikely	Maybe or maybe not	Likely	Very likely



21. For the spreadsheet to the left with the indicated cell relationships, what difference do you notice in cells A and B? Why?



22. For the spreadsheet to the with the indicated cell relationships, what difference do you notice in cells A and B? Why?

Tutorial (low-cost)

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

The other people involved in this study are Dr. Margaret Burnett, Dr. Curtis Cook, Shraddha Sorte, Joey Lawrance, Sienna Hiebert, and the assistants helping me out today.

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

Before we begin, I'd like to ask if anyone in here is 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.

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 couple of sample spreadsheet problems. After the tutorial, you will be given two different spreadsheets; asked to test the spreadsheets, and correct any errors you find in them.

- As we go through this tutorial, I want you to ACTUALLY PERFORM the steps I'm describing. For example, at times I will want you to click the left mouse button, at times I will want you to click the middle mouse button (the scroll button in the middle of your mouse) and at other times I will want you to click the right mouse button. 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 each 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 colored 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 *(show wave the mouse around the cell and then rest mouse on border)*. Note that a tooltip will pop up and tell us what this color means. Can anyone tell me what the message says? *(PAUSE, look for a hand.)* Yes, it means that the cell has not been tested.

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 is just telling us that the cell has not been tested. It is up to us to make a decision about the correctness of the cells 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 they just have values as you're going to see in a few minutes. Cell's with formulas have colored borders.

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 anyone tell me what the tooltip says? *(PAUSE, wait for answer.)* Yes, it says that if the value of this cell is correct, we can left-click and if the value of the cell is wrong, we can right-click. It also tells us that these decisions help test and find errors.

So let's left-click the question mark in this decision box for TotalCost. Notice what happened. 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 20% *(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 20% of this spreadsheet.

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 left-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 left-click in the decision box for TotalCost to put our checkmark back in that box.

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 let's hide them by middle-clicking on the TotalCost cell.

Now, let's test the BudgetOk cell (we'll skip over the DiscountedCell for the time being) by making a decision whether or not the value is correct for the inputs. What does the spreadsheet description say about my budget? Let me go back and read… "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 that 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 let's 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, my spreadsheet already has these values, so I 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 I 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 tooltip. To remind us what the question mark means, move your mouse to the BudgetOk decision box. The tooltip tells us that if the cell's value is wrong to right-click. Well, this value is wrong, so go ahead and right-click on the question mark in this decision box.

Hey, look at that! 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 tooltips.

Now let's make a decision about DiscountedCost's value. For the current set of inputs, DiscountedCost should be 1600. But our DiscountedCost cell says 2,520. That means the value associated with the DiscountedCost cell is "Wrong". Right 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 tooltips.

TotalCost's value should also be 1600 for the current set of inputs, but our TotalCost cell says 2800. Place an X-mark on this cell as well. Take a few seconds to explore anything that might have changed by moving your mouse over the items and viewing the tooltips.

Finally, I notice that, according to the example spreadsheet in my description, PaperTotalCost should be 1600. But our value is 2800, and that is wrong. So let's 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 let's change the 7 in this formula to a 4, and click the Apply button to finalize your changes.

Hey wait, my total spreadsheet testedness at the top of my 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. I have a new formula in this cell, so those tests are no longer valid. But, never fear, I 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, left-click to place a checkmark in the decision box for PaperTotalCost. Oh good, the percent testedness of my spreadsheet went up to 5%; I 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. So let's click in the decision box of this cell to place a checkmark.

But wait! The border of this cell is only purple. Let's rest our mouse over this cell border to see why. The tooltip says that this cell is only 50 percent tested.

Middle-click on this cell to bring up the cell's arrows. Hey, the arrows are both purple too. Let's rest our 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 I make a testing decision on this cell, I 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, I have 31 paper on stock. Is this enough paper? The spreadsheet description says I need 400 reams of paper, but I only have 31. So this is not enough paper. And the PaperQCheck cell says "not enough paper". Well, this is correct, so let's left-click on the PaperQCheck cell's decision box. Alright! The border changed to blue, and even more, the spreadsheet is now 25% tested.We don't need those arrows on PaperQCheck anymore, so let's 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. 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 let's right-click on the question mark in PenQCheck's decision box.

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 everyone 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 we are done. (*wait a second*). Note that PenQCheck now displays the correct value. So let's go ahead and put a checkmark in this cell by left-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. 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.

<Give them more time than treatment group by aprx. 4-5 min!!!>

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 frontside 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 by left-clicking cell decision boxes, and use the X-marks by right-clicking 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 frontside 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 by left-clicking cell decision boxes, and use the X-marks by right-clicking decision boxes.

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

(*Task is 35 minutes*)

Tutorial (high-support)

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

The other people involved in this study are Dr. Margaret Burnett, 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 tutorial of Forms/3, and then you will have a few experimental tasks to work on including a post session questionnaire.

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

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 tooltip 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 by pressing the expander, which we will try later on.

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 they just have values as you're going to see in a few minutes. Cell's with formulas have colored borders.

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 tooltip 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.

The questionmark is replaced by 4 choices– 2 X marks and 2 check marks. The tooltips for each choice, starting from the left, are, "It's wrong", "Seems wrong maybe", "Seems right maybe" and the rightmost tooltip says, "It's right". Also, next to each tooltip was a keyboard short cut which we will use in just a moment. 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 <pause>. 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 20% *(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 20% of this spreadsheet.

What about that other checkmark that we saw? We'll try that one, by first undoing the checkmark by left clicking on it. 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)***

Again, you can "uncheck" the decision about TotalCost by clicking on that checkmark in TotalCost's decision box- try this. ***(Try it, and Pause)*** Notice that 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.

As I pointed out before the tool tips showed something you could use as the shortcut for each decision. Let's try one of those since we've already decided the value in the TotalCost cell is correct. The shift key means the value is good, and combining an alt key places the alternative checkmark. Try one of those now. By holding down the shift key and clicking the questionmark, or the shift and alt keys.

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. The tooltip says 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 (the scroll wheel). 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 the tooltip 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 does, 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 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…, "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 our spreadsheet to match 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 click on the little button with an arrow on the bottom-right-hand side of the cell. 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 in the Forms/3 environment, you can place your mouse over that item, and wait for the tooltip. To remind us what the question mark means, move your mouse to the BudgetOk decision box. The tooltip 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. Let's read the tooltips on the X's, the leftmost tooltip says, "It's wrong" and the other tooltip says "Seems wrong, maybe". Go ahead and click the X you think is most appropriate.

As you probably noticed, 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 tooltips?

Now let's make a decision about DiscountedCost's value. For the current set of inputs, DiscountedCost should be 1600. But our DiscountedCost cell says 2,520. That means the value associated with the DiscountedCost cell is "Wrong". Place another X, this time try holding down the control key while clicking on the ?. 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 viewing the tooltips.

TotalCost's value should also be 1600 for the current set of inputs, but our TotalCost cell says 2800. Place an X-mark on this cell as well (again you can do this by holding down the control key and clicking, or just clicking on the ?). Take a few seconds to explore any new changes.

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 four 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, click (with the shift key held down if you want) to place a checkmark in the decision box for PaperTotalCost. Oh good, the percent testedness of my spreadsheet went up to 5%; We got some of our 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 reams of paper, which is enough to fill our shelves. Since the PaperQCheck cell says “paper quantity ok”, its value is correct. Place a checkmark in the decision box of this cell.

But wait! The border of this cell is only purple. Rest your mouse over this cell border to see why. The tooltip says that this cell is only 50 percent tested. What should we do? At this point let’s get some more information by clicking on the expander at the bottom of the tooltip. <**pause**> <read it to them> It suggests changing some input values, but what should we change?

Middle-click on this cell to bring up the cell’s arrows (if you need to go ahead and unexpand the tips). 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.

Remember the tips information told us to try new values.

Change the value of the Paper cell to see if we can find this other situation. Open the formula. 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. change Paper’s 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. However, this is not all that has changed. Go ahead and open the Tips expander for PaperQCheck’s border tip again. **<PAUSE>** Notice that Forms/3 has updated the Tips to help us for this new situation. <PAUSE so they can read it> Go ahead and close that tip.

Now, 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 to place a checkmark. Alright! The border changed to blue, and even more, the spreadsheet is now 25% 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. 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.

Let's test the DiscountedCost cell. Is the value of 40 correct for this cell? Well, since according to our spreadsheet description we haven't bought enough to get a discount (we needed to buy $1500 to get the discount) this value does appear to be correct. So, let's check off this cell's value. The border is purple, rest your mouse there to be reminded of what this means, right, it's just 50% tested.

Expand the tips part of this tooltip. Notice the part about the help-me-test, last time we had a purple cell we tried to think of what inputs to change to get a new situation, but Forms/3 can also help, to use help-me-test select the Discounted Cost cell (you may need to dismiss the expanded tooltip first then left clicking on the middle of the cell). Now, do you see the help button at the top of the screen? Go ahead and click it. If you're looking you may or may not see values in the cells changing, but when Help stops you'll notice a couple of other things.

The thicker borders just let us know which cells the Help feature modified to give us a new situation.

So help me test changed the values of the pen's cell to be 1000, and notice also that DiscountedCost now has a ? in the decision box. Let's test this cell, does this value of 1836 appear to be correct given the inputs? Well, according to the description the price should be discounted 10%, so this appears to be correct. Go ahead and check off that cell

I had you select the cell you wanted to get a new value for before pressing the help me test button, but the button also works if you don't have any cell selected, then it just tries to find any new test case for the whole spreadsheet.

Now let's look at the PenQCheck cell. This cell is displaying "not enough pens". Is this correct? Our spreadsheet description says you must keep more than 68 boxes of pens on hand. We have 1025 boxes of pens in stock, because the Pens cell is 1000 and the PensOnHand cell is 25. So even though we have enough pens, the PenQCheck cell is displaying "not enough pens". 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 make sure everyone made the same change. (*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 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. 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 frontside 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.

Don't forget you can always get more help from the tooltip with the expanded stuck information.

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 frontside 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.

Don't forget you can always get more help from the tooltip with the expanded stuck information.

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

(*Task is 35 minutes*)

Appendix C: Excel Study Materials

Background Questionnaire

(since we used a questionnaire package, we just restate the questions here):

Collected information:

- Gender
- Age group (<20, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60+)
- Major or educational background
- Highest degree completed (Less than high school, High school, Some college, Associate's Degree, Baccalaureate Degree, Masters Degree, PhD, Other (please comment below)
- Current job
- Programming experience (If yes, then programming languages)
- Spreadsheet experience (years)
- Professional spreadsheet use (If yes, then length in years)
- English as primary language (If no, then length speaking English)
- Self-efficacy questionnaire (as worded in Appendix A).

Post-Session Questionnaire:

The following questions ask information about the task and Excel.

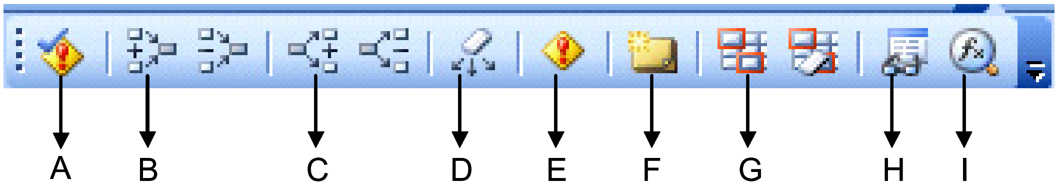
1. Rate your level of agreement with the following statement. I am confident that my changes were accurate and correctly changed how the spreadsheet works.
2. How long did it take you to complete the task? (You can also specify if you did not have enough time.)

Mark how you found the following features for modifying and ensuring the correctness of your spreadsheet. (On 5-point likert scale from Strongly Disagree to Strongly Agree.)

3. Error checking helped me make progress.
4. Trace Precedents arrows helped me make progress.
5. Trace Dependents arrows helped me make progress.
6. Trace Error helped me make progress.
7. Comments helped me make progress.
8. Circle Invalid Data helped me make progress.
9. Show Watch Window helped me make progress.
10. Evaluate Function helped me make progress.

11. Order your preference for the following features: Error checking, Trace Precedent Arrows, Trace Dependents Arrows, Trace Error, Comments, Circle Invalid Data, Show Watch Window, Evaluate Function

The following questions look at your understanding of the Excel features.



Using the picture and letters [above] answer the following questions.

12. Which feature will show you where a value is used?
13. Which feature will help you find inconsistent formulas?
14. Which feature will guide you through a formula?
15. Which feature will allow you to see a value which is off screen?

16. Which feature will allow you to step through multiple layers of calculations?
17. Which feature will allow you to see what cells affect the current formula?
18. Which feature gives suggestions for changes to cells with potential errors?

The follow questions are a set of true false questions.

19. The evaluate function feature allows you to edit the formula you’re evaluating.
20. The evaluate function feature allows you to "step into" other cell’s functions.
21. The evaluate function feature allows you to get help on the function you’re interested in.
22. The error checking feature provides the option for help on the errors it finds.
23. The error checking feature will offer to fix your formula for you.
24. The error checking feature will offer more than one fix suggestion for you formula.
25. The trace precedents arrows can be "moved" to refer to a different cell.
26. The trace precedent arrows for one cell can be removed one at a time.
27. The trace precedent arrows show just one arrow coming from a whole range of values.
28. The trace dependent arrows button can be made to show arrows that also come into a cell.
29. The trace error feature brings up arrows.
30. The trace error feature makes suggestions on how to fix an error.
31. The watch window allows values which are off screen to be easily seen.
32. Using the watch window for a cell you can also see its formula along with its value.
33. A formula can be changed for a cell that’s in the watch window.

Tutorial

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

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

So long as you're comfortable, you've got you chair adjusted and your mouse in a comfortable position, we can get started. I also want to re-iterate what you just read, that if you experience any difficulty with the software this is not your fault, and you should let me know about it.

<Pause>

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, your task will be to do some modifications to a spreadsheet, since some of them will be rather challenging I'll first leading you through a tutorial covering Excel features which may help you in your task. Following the task you'll have a questionnaire to fill out.

During the course of the tutorial I will give you several minutes to explore different features and get more comfortable. During these times I encourage you to explore things we have already learned in the tutorial.

Some of what I'll be talking about in the tutorial you may already be familiar with. If this is the case please follow along with the tutorial and stay focused, it's important that I cover the same information for everyone in the study.

As I just mentioned, in this experiment, you will be working with Excel to make some modifications to a spreadsheet. One aspect of making changes to a spreadsheet is ensuring your changes actually do what you intended.
During this tutorial I will show you different features in Excel which may help you to both make the modification to the spreadsheet, and then make sure the changes do what you intended them to do.

- As we go through this tutorial, I want you to ACTUALLY PERFORM the steps I'm describing. 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 the spreadsheet that we will be working with, you will have a sheet of paper describing what the spreadsheet is does the tasks for you to complete.

Now open the Learning Style Preferences spreadsheet by selecting the bar labeled Learning Style Preferences at the bottom of the screen.

Let's go over this paper and the spreadsheet together.

The basic idea of this spreadsheet is that when a person wants to know what their preferred learning style is (how they learn best) they read each statement and say how often they do that particular statement. We have an example of the answers already filled in.

On the paper that area is labeled "Box A."

Once all the questions have been answered Box B shows where the learning style preferences will appear. In one column it shows the name of the learning style, and in the next the number of "points" that style received. The rest of the spreadsheet is doing calculations which help to decide which is the most common learning style.

So, the area in the pink is getting the total points for each learning style and assigning a preference number. While the area in the orange is then ordering those learning styles to have the one with the most points on top.

Read the first task.
<*Go ahead and read the first task*>

Before we start changing the spreadsheet let's find out how the answers are currently being used. Select cell B3 – the user's first answer. To see where this cell's value is being used we'll use something called trace dependents. The button is on the auditing toolbar the 4th button from the left, and if you wait for the tooltip it says trace dependents. Click that button.

This brings up two arrows, which show the places in the spreadsheet where the value of this cell will be used in other formulas. So the value in B3 is used in two formulas, C3, and G7.

Let's take a look at how B3 is used in C3's formula. Click on C3 and look at the formula bar at the top of the screen. This formula is determining how many points their answer is worth.

In case you aren't completely comfortable with how "if" formulas work let's go over this one in some detail. In fact, we'll look at two different cases for it.

It starts by comparing B3=E9, it's looking to see if B3 (which has the value often) is equal to the value of cell E9 – the value of E9 in this case, so the two values are equal. If the comparison is true it goes to the part of the formula right after the first comma, the then part. In this case, since they are equal it prints 5 to this cell.

Let's go through another example where we change the value of B3 to seldom. If the arrows are getting in your way you can erase by clicking on the middle button in the audit toolbar called "remove all arrows." Once you've changed the value to seldom let's go back to cell C3.

This time when it checks B3=E9 those two are not true, so it goes into the next if checking if B3 is equal to E10, E10 is sometimes, so that is not true; if those two were equal then it would print 3, but, since they aren't equal it goes into a third if statement where it compares B3 to E11 (the value seldom), and they are equal, so it prints out 1. If each comparison was not true the answer that would show up is 0.

The right most button in the auditing toolbar, called "evaluate formula" can show you what a function is doing. Make sure you have C3 selected and click on that now. Take a minute to explore "evaluate formula," you can try it out on a few different formulas if you would like to see how it works.

<*wait a minute*>

I want to point out one last thing in this formula in case you haven't seen it before, you notice the formula starts as if B3=E9, but E9 has the $ signs around it. That means that if you copy that formula into a cell next to it that E9 will stay the same, it will always refer to the cell E9. When you don't have $ signs, such as the B3, when you copy the formula somewhere else, let's say down one to C4, the B3 will update and refer to cell B4. But, the E9 will always refer to cell E9.

<*pause, breath!*>

The reason we started to look at that formula was because we were trying to find out how the user's answers are used in the spreadsheet. We saw that B3 went into C3. Let's find out what the value from C3 affects.

Making sure you're on cell C3, click the trace dependents button.

That points to cell G3, so let's look at that cell's formula.

It appears to be summing up some of the different learning style statements. Use the "trace precedents" button to see which cells these are. The trace precedents is the second button on the toolbar. It shows which cells contribute to a formula.

So, this formula is looking at those statements which are assessing visual/verbal learning styles.

This might be a good time to add one of our statements from Task1, let's add the statement for row #35 which according to our description is another visual/verbal learning style statement. Select row 35 in order to insert a row above it. To insert the row go into the insert menu, select insert row.

Since we aren't told what the text of the statement should be let's just write something generic like, New visual/verbal learning style statement in column A, and for column B select one value.
<wait>
For column C let's type in the formula that should go into that cell. I'm having you type it out to make sure you are comfortable with doing formula like this one. Ok, so the column C was determining how many points the users answer is worth. Let's work on putting in this formula. First, since it's a formula we have to put in the = sign. Then type "if" and an open paren.

Once you're this far in the formula Excel, as you may know, tells you what it's looking for in this type of formula. It says it's looking for a logical statement, which for our case would be a comparison. We want to compare the answer the user just put in to see if it's = to "often" for example. So, type in (or select) cell B35=E9, this will check if both are equal to often. If they are then that answer is worth 5 points so type a comma (notice now we're on the value if true area) and type in 5. Then another comma, and now we're in the value if false area. So, if it's not equal to often then we should check to see if it's equal to "sometimes" to do this we need to start another "if" so type in "if(" now let's type in B35 again and compare it to E10 this time, the "sometimes" add another comma and this time if they are equal the points should be 3. And again, we have to write one more if (after putting in another comma). So, if(and this time the comparison should be between B35 and E11, comparing to "seldom", and if both are equal to seldom then the points are 1, and if none of those answers are what is in the cell we just want 0 points to go into that cell, so this time after the 1 and the comma put a 0. We now have to put 3 closing parentheses.

Recall before I talked about the $ signs about some of the cells if we always want to refer to specific cells and not have them change if we copy the formula somewhere. Let's put the $ signs into this formula, we need to put on before each of the E's and then before each of the #s of those E cells b/c we want it always to refer to those exact cells.

Let's check on cell C35 to make sure the formula works as we expect.

Remember, when you change the spreadsheet it's important that you check your changes to make sure there are no errors. First, let's turn on the precedent arrows for C35. It refers to B35 & the cells E9-E11, as we expected it would. This looks good.

We're not done checking it yet though, it's also a good idea to check a different value, rather than just checking the formula. To do this let's change the value of cell B35 to sometimes. After doing this we expect 3 to appear in C35 – and it does. Now, change to seldom.

We have a few more places to update, for example, we need to update cell G3 to take this new visual/verbal learning style question into account (just notice before we start this that the value right now is 20 in cell G3). Click on G3 and add + and C35 (or click on that cell). Press enter.

Let's make sure that the change we made to G3 is correct, notice right now that the value of cell G3 is 21, which is one more than the value before we made that change of adding cell C35 to the formula. If we change the value of cell B35 to "often" – which is worth 4 more points than seldom then we would expect cell G3 to be 25 when we change B35. Go ahead and make that change now to see if the value is as you expect it. And in fact the cell G3 does display 25 as expected.

<pause>

Earlier, when we clicked on the trace dependents from cell B3 it also referred to cell G7. Let's go to that cell now and see if we also need to update it. Looking at its formula it's doing something called countA- this function, according to the help I looked up for it, counts any cell in the range you specify which is not blank. So, it's counting the number of questions the user of this spreadsheet has answered.

It should also include cell B35 in its range. Update this formula.

Whoa, did you notice when we made that change that the cells in the lower part of the spreadsheet stopped displaying information!??!

Let's look at the dependents of cell G7 by turning on the trace dependents. We see that this impacts each of the cells that disappeared! Click on cell A37. Is there a problem with this formula? Take a minute to think about that. Feel free to use the evaluate formula button again if you think it would help you to understand what this formula is trying to do.

<wait ~1 minute>

Ok, as you might have noticed the change needs to be in the comparison for this cell. Instead of =32 in the if part of the formula, it should be equal to 33 questions, since we now have 33 questions. Go ahead and make that change now. Remember it needs to be done for each of these cells which display an answer. (This also includes cells in column C which are supposed to display some points.)

<wait a few minutes>

Once you made the change we need to be sure the changes worked as expected. What should we check? First, we know now that all the answers should be appearing right now, so let's make a change where not all the questions are answered. Just delete any one of the user's answers (by selecting the cell and pressing delete). The boxes on the bottom should disappear when you do that.

I wanted to point out a feature before I give you time to finish up the tasks we started.

The first is the error checking button, this looks for the triangles which point out what excel believes is a suspicious formula. To explore this you can change the formula C8 for example, change one of the values, for example the 5 to a 10. Now click on the first button in the auditing toolbar. Take a moment to explore this.

<*wait about 30 seconds*>

Now, take some time exploring the rest of the spreadsheet, completing the task using any features you think may help you complete this. And, as we did during the tutorial make your changes carefully and make sure the spreadsheet works as you expect it should after the change.

Also, there are other features on the audit toolbar that could be helpful in completing your task.

<*wait 5 minutes*>

This next spreadsheet is a Gradebook spreadsheet. Let's take a look at the description and spreadsheet together for a minute.

The first thing to notice is what I have labeled as Box E – which is where all the students' grades have been put into the spreadsheet. All the grades are added together in column E for the total points, and in Column F it's the average over all the possible points they could have gotten. These Average points are then used to help calculate the Letter grade and GPA. The limits for the letter grade and GPA are at the top of the spreadsheet in Box A.

Box B shows the assignment name and the points for that assignment, and finally the Class Summary is under the grades, highlighted by Box F.

Your tasks are on the back of the piece of paper. There are 4 tasks listed, you have to do 2 of those 4, you can choose any 2. Your changes should work even if we have new students with different grades!

Remember, it's important that you make sure you changes work as you expect them to, and also that there are many features in the audit toolbar which could help you in this task.

You have about 45 minutes starting now before I'll ask you to stop.

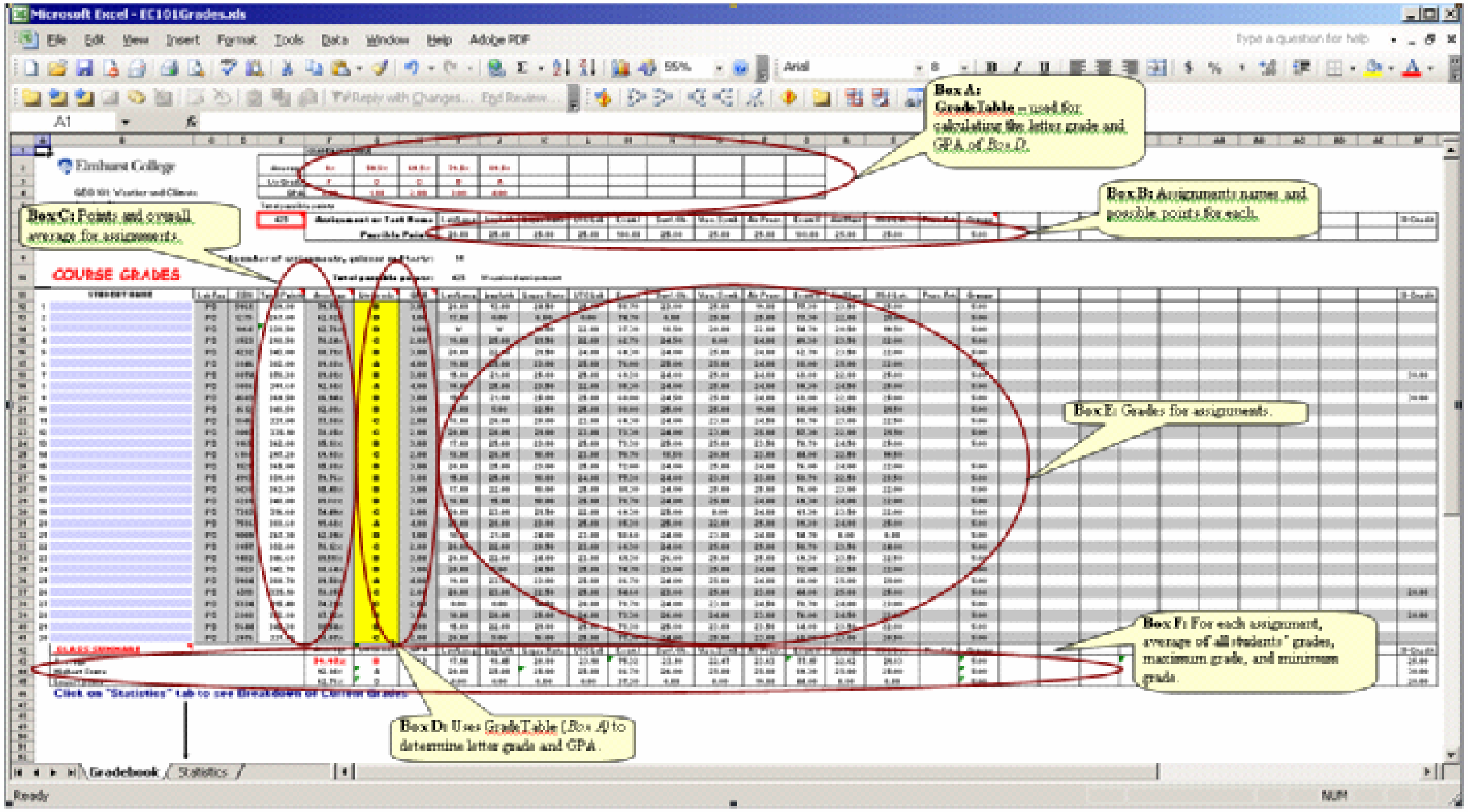
Grading

Task 1: Add 10 lab columns for this course. (See full task description in Figure 32)
Sub-Task1: Add 10 columns for lab (row 11) – requires adding columns & filling this in with lab names.
Sub-Task2: Add 10 columns for assignment points (row 7) assign points for lab (row 8) – this area of the spreadsheet keeps trace of assignment names and total possible points.
Sub-Task3: Determines # of Labs attended – In order to determine if enough labs were attended some sum of the labs attended needed to be determined.
Sub-Task4: Letter grade is F if less than 7 labs – If the "if" statement determined if enough labs were attended.
Sub-Task5: Total Points doesn't count labs 1pt if didn't fix 2 – although the area of assignments are (referred to in the second subtask) – requires undoing some of the. This was a side effect of sub-task2. The task was specific about not counting the labs in the total score, but by default the was counted. If the user did not complete sub-task2 then this would not have been a problem, and they would get 1 point for this problem.
Sub-Task6: Sum of each students doesn't count labs, but counts extra credit. This was also a side effect, depending on how the user went about adding in the labs, the extra credit should still be counted in the grade, but the lab grade should not be included.
Penalties: Grade lookup doesn't work (LtrGrade column): This takes away a point if they modified a cell (and left an error in it), for a cell that they did not need to modify.

Task 2: Waived homework. (See full task description in Figure 32)
Sub-Task1: Add a column for waived points (not required, but a first step in determining the number of points that should not be counted in the final grade).
Sub-Task2: Make a formula to determine the number of waived points (this is an "if" formula).
Sub-Task3: Update F14 to get rid of -45 (This is fixing an existing problem with a formula which was manually excluding 45 points from this student's grade.)
Sub-Task4: Update formula for all scores for column F
Penalties: Start to make modifications for this task, but do not update total points to include all possible points (even if W is a score).

Figure 42. Grading Scheme for Excel Study.

Spreadsheet



Microsoft Excel - EC101Grades.xls
File Edit View Insert Format Tools Data Window Help Adobe PDF
Elmhurst College
COURSE GRADES
Click on "Statistics" tab to see Breakdown of Current Grades
Gradebook
Statistics
Ready
NUM

Box A: GradeTable – used for calculating the letter grade and GPA of Box D.

Box B: Assignments names and possible points for each.

Box C: Points and overall average for assignments.

Box D: Uses GradeTable (Box A) to determine letter grade and GPA.

Box E: Grades for assignments.

Box F: For each assignment, average of all students' grades, maximum grade, and minimum grade.

Figure 43. Excel Task. The annotated screen shot given to part icipants in the study. The tasks were given on the backside of the paper. (Tasks are show in Figure 32.)