Investigating the Impact of Teaching Inclusive Design Concepts on Ecampus Students' Mindsets

by
Rosalinda Garcia

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 18, 2021
Commencement June 2021

AN ABSTRACT OF THE THESIS OF

Rosalinda Garcia for the degree of Honors Baccalaureate of Science in Computer Science presented on May 18, 2021. Title: Investigating the Impact of Teaching Inclusive Design Concepts on Ecampus Students' Mindsets.

Abstract approved: _____ _____

Margaret Burnett Larissa Letaw

Although computer science (CS) education researchers and practitioners have found ways to improve CS classroom inclusivity, few researchers have considered inclusivity of online CS education. We have begun developing a new approach that we term "embedded inclusive design" to address inclusive CS. The essence of the approach is to integrate elements of inclusive design education into mainstream CS coursework. This paper presents three curricular interventions we have developed in this approach, and empirically investigates their efficacy in online CS post-baccalaureate education. Our research questions were: How do these three curricular interventions affect (RQ1) the climate among online CS students and (RQ2) how online CS students honor the diversity of their users? To answer these research questions, we implemented the curricular interventions in four asynchronous online CS classes across two CS courses at Oregon State University and conducted an action research study to investigate the impacts. Results show that online CS students who experienced these interventions reported feeling more included in the major than they had before, reported positive impacts on their team dynamics, and increased their interest in accommodating diverse users.

Key Words: Human-centered computing, gender, computer science education, online learning

Corresponding e-mail address: garciros@oregonstate.edu

Investigating the Impact of Teaching Inclusive Design Concepts on Ecampus Students' Mindsets

by
Rosalinda Garcia

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 18, 2021
Commencement June 2021

Honors Baccalaureate of Science in Computer Science project of Rosalinda Garcia presented on May 18, 2021.

APPROVED:

_____

Margaret Burnett, Mentor, representing Electrical Engineering and Computer Science

_____

Larissa Letaw, Mentor, representing Electrical Engineering and Computer Science

_____

Anita Sarma, Committee Member, representing Electrical Engineering and Computer Science

_____

Toni Doolen, Dean, Oregon State University Honors College

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

_____

Rosalinda Garcia, Author

Acknowledgements

To Margaret Burnett, for all the support and knowledge you have given me. Thank you for helping me find a passion for research and giving me an opportunity to change the world. I am beyond excited to continue working with and learning from you.

To Larissa Letaw, for your guidance and expertise both on this project and in general. Thank you for always being willing to answer questions and help me understand the fundamentals of conducting a research study. Working on this project with you has been incredible.

To all my current and past lab mates, thank you for everything from welcoming me to the group, to teaching me, to laughing with me. I wish you all the best and hope our paths cross in-person again in the future.

And lastly, to my friends and family, for being there every step of the way and listening to me ramble on about this topic every chance I had. I could not have done it without your endless support.

This thesis is an extension of a paper submitted to the 2021 International Computing Education Research Conference. Special thanks to my co-authors: Larissa Letaw, Heather Garcia, Christopher Perdriau, and Margaret Burnett, and to our research co-contributor Aishwarya Vellanki.

Contents

1. Introduction

Although bits and pieces of computing curricula continually change as computer science (CS) technologies advance, the CS major course sequence has changed little in decades: it begins with a few introductory computing courses followed by a sequence of data structures, algorithms, theory, 1-2 programming language fundamentals and/or implementation courses, 1-2 operating systems courses, a collection of software engineering courses, and some electives [1]. These courses reinforce and build upon one another: programming language courses use some theory and data structures taught earlier, operating systems courses use data structures and computer architecture concepts, and so on.

But courses that address how the software students build can affect the people who use it, such as CS ethics, human computer interaction (HCI), or usability courses, are essentially sidelined, with little connection to the rest of the CS education that students receive [62]. The message comes through to CS students loud and clear: concepts in ethics, society, and humans, are unimportant to CS professions. In response to growing evidence of such problems, CS-education researchers have called for not only an increase in teaching ethics and social consequences in CS, but also increasing coverage of such topics in mainstream CS courses [19, 24-26, 44, 62].

One effect that has been called out less often is that CS education shows direct evidence of producing CS professionals who are unable to create inclusive technology. For example, in one recent study of post-secondary computing faculty, 49% of the faculty reported having seen their students struggling to prevent or even recognize how their biases affected the software they were designing [57]. In the same survey, 54% of the faculty also reported having seen students in their courses finding it difficult to design for diverse people's abilities and usage styles [57]. As Putnam summarizes, as long as CS education continues to sideline the concept of designing inclusive software, it perpetuates "the cycle of ignorance among ... developers <that> maintains the status quo of exclusion and marginalization" [61].

We are developing a new approach to mainstreaming inclusivity in CS education, in which we integrate portions of an inclusive design method called GenderMag into a variety of mainstream CS courses. We call our approach "embedded inclusive design". In essence, the approach incrementally embeds bits of inclusive design into the work students already do. The goal is to increase not only the inclusivity of CS education itself, but also students' attitudes toward product inclusivity of the software they are creating. Because students "do equity" as part of their mainstream CS work, we hypothesize that the approach will produce students who (1) are more inclusive to each other and (2) more aware of diverse users.

In this thesis, we focus on inclusivity in asynchronous, online, post-secondary CS education at Oregon State University (OSU). We present our first three interventions of the embedded inclusive design approach and evaluate their effectiveness in four online CS classes. Three were separate offerings of a third-year (junior-level) database course (CS-DB) and one was an offering of a third-year (junior-level) software engineering course (CS-SE). Both are required courses for the online CS major.

At Oregon State, the online program is a post-baccalaureate program, taken by people who previously earned a non-CS post-secondary degree, and now are taking courses in the CS major to earn a second post-secondary degree, namely in CS. Each "class" is entirely asynchronous—there are no synchronous class meetings, and people from different locations and timezones around the world can, within limits, set their own schedules. However, students in a given class commit to starting and ending the course on traditional term-calendar boundaries, to completing the

assignments by certain deadlines and, in some classes, need to work (asynchronously) with other students in that class on a team. Instructors and teaching assistants (TAs) are permanently assigned for the duration of a term, and they answer individual questions (via email or discussion platforms), provide timely feedback on assignments, run discussion forums, and so on. Classes tend to be large—in our investigation, class sizes were 218, 150, 213, and 226 students. A total of 64 of these students opted in to allowing their work to be used for research purposes.

Within this educational setting, we conducted an Action Research investigation. Action Research is a type of longitudinal field study that involves engaging with a community to address some problem and through this problem solving develop scholarly knowledge [34]. As per Action Research's longitudinal focus, our involvement spanned months. Specifically, we had consistent involvement over 9 months (three terms) with four faculty members at the university. We structured our investigation around the following research questions:

- RQ1: How do these curricular interventions affect the climate among online CS students?
- RQ2: How do these curricular interventions affect online CS students' respect for users' diversity?

2. Background

Our approach leverages the GenderMag method's components and foundations. GenderMag [13] is an evidence-based method for avoiding, finding, and fixing inclusivity "bugs" in software. The process aspect of the method is a specialized cognitive walkthrough, but here we describe only its facets and personas, since those are the portions that our curricular interventions leveraged.

GenderMag's cognitive styles (cognitive "facets" in GenderMag) form the core of the GenderMag method. Each facet captures different individuals' diverse cognitive approaches by defining an evidence-based range of possible values. The five facets capture diversity of motivations for using tech; information processing style; computer self-efficacy; learning style (by process or by tinkering); and attitude toward risk. GenderMag defines "inclusivity bugs" as omissions of a technology product to support these five facets' full ranges of values. For example, technology features that support risk-tolerant users but present barriers to risk-averse users have inclusivity bugs.

Such barriers are cognitive style inclusivity bugs because they disproportionately impact people with particular cognitive styles. They are also gender-inclusivity bugs because the facets capture (statistical) gender differences in how people problem-solve [3, 13, 16, 17, 69, 73].

GenderMag uses three personas to bring the facets to life. Abi's and Tim's values for each of these facets lie at opposite ends of the spectrum, and Pat has values within. The Abi persona represents facet values whose proportions disproportionately skew towards women, Tim represents facet values that disproportionately skew towards men, and Pat provides a third set of values [13]. The interventions we describe in this paper include these three personas, shown later with our curricular interventions (Section 4).

Empirical studies have found GenderMag to be effective at identifying inclusivity bugs and at pointing toward effective fixes [11, 13, 21, 35, 58, 64, 72]. However, in the realm of CS education, the only work relating to GenderMag is Oleson et al.'s Action Research investigation into how to teach GenderMag in face-to-face university CS classes [56]. No prior work has investigated incorporating aspects of GenderMag into online CS courses, or the effects of doing so on the inclusivity climate of any CS course.

3. Related Work

Many researchers have reported issues with inclusivity in online CS education's climate. For example, one study reports that instructors are more likely to respond to forum posts by White male students [4]. On one popular online discussion question-and-answer site, Stack Overflow, women tended to ask fewer questions, answer fewer questions, have lower reputation scores than men, and experience barriers to participation such as feeling intimidated and being unable to identify other participants of their gender [28, 72]. Although researchers investigating the Piazza online gathering site for students have reported better experiences by women than in Stack Overflow, a recent study of over 2500 Piazza users reported that Piazza women still feel the need to keep their identities and genders anonymous, and when they did not, to be less likely than men to receive answers to their questions by members of the same gender [71].

Further, Phirangee and Malec identified three "othering" themes—professional, academic, and ethnic—which are experienced by women in online learning. Despite differences in these three themes, each type of othering resulted in students feeling excluded from their online learning communities [59]. Dym et al. likewise reported that LGBTQ+ programmers in online communities expect few women and LGBTQ+ individuals to become CS students without additional support or encouragement because, in these participants' experiences, the field exhibits little diversity and a heterosexist climate [23]. Dym et al.'s results are not unique; similar results have been reported by other investigations of the experiences of individuals with queer gender and/or sexual identities [51, 52, 67].

Non-inclusive academic climates can affect students' performance and completion in these education environments. Kizilcec and Halawa found that women were less persistent with lectures and assessments in online courses. They also found that feelings of social belonging are influenced by success in the classroom, which can negatively impact women when they have higher attrition rates. Similarly, in an online conversion program helping individuals change career paths to computer science, women were much less likely to finish the program than men [38].

Both online and in-person CS education research have investigated factors, including gender differences, that contribute to students' feelings of exclusion. For example, Pournaghshband and Medel point out that much of society embraces a widely accepted "fallacious archetype" of what a successful CS student looks like: a young, White male with at least mid-level socioeconomic status [60]. Kuttal et al. reported differences in women's and men's experiences when completing a remote pair-programming assignment [45]. One of these differences was that their communication and gender awareness differed significantly—women relied more on non-verbal communication, which is difficult in an online setting. Women also preferred co-located pair programming whereas men were comfortable with a remote setting. Several have reported women in in-person CS classes to have less passion about technology per se but more passion about "computing with a purpose", and lower confidence in their computing abilities [2, 7, 20, 50]. Low confidence can become even lower when students compare themselves to others, such as in [38] where women students reported that they constantly compared themselves to more experienced students and became less confident when they saw experienced students struggle. Gender differences in confidence have in turn been linked to gender differences in communication in CS classes; for example, Alvarado's study found that women were less comfortable than men were when communicating with their instructor [2].

A significant body of work has investigated increasing recruitment and/or retention across genders in in-person CS education, and these works have brought about improvements in both

recruitment and retention. Among the especially well-known practices are: pair programming (e.g., [75, 77]), meaningful or socially relevant assignments (e.g., [6, 10, 49]), and leveling the playing field with mechanisms like having everyone start with a language new to all or eschewing programming entirely to instead focus on problem solving (e.g., [43]). Some of these practices, such as giving socially relevant assignments or changing the language used in the course, are not reliant on synchronous or in-person presence in classes and thus can transfer directly to asynchronous online CS classes. However, all of these practices tend to be unidirectional—they aim to make more students feel included, but do not generally aim to improve students' inclusivity toward others.

Closest to our own research is work on using universal design to improve inclusivity for disabled stakeholders in in-person CS classes. For example, the DO-IT project created a web development course that integrates accessibility and universal design into its curriculum [22]. To increase feelings of inclusion by both women and students with disabilities,

Blaser et al. have proposed including universal design principles in engineering courses in order to prepare future engineers better as well as improving representations of disabled users and engineers [8]. This research rests on prior investigations of what diverse students value in their courses and jobs, reporting that women in engineering often value contribution to society more than men do, which suggests that women may be drawn to inclusive and universal design (e.g., [32]). Similarly, Izzo et al. have found that teaching universal design in college courses in order to include people with disabilities helps both students and instructors to improve accessibility, awareness, and instructional flexibility [39]. Putnam et al. [61] and Waller et al. [74] have both experimented with integrating accessibility concepts across multiple face-to-face courses in the major, treating accessibility as an integral part of design and development. Others have investigated including stakeholders with a disability (e.g., a wheelchair-bound user) in design/evaluation team sessions [48, 65]. Our approach applies many of the Putnam and Waller recommendations to our project, and also leverages elements of the stakeholders strategy, but our "stakeholders" are research-based personas instead of actual people. Most important, our education setting is asynchronous, online courses rather than face-to-face courses.

Thus, although there is extensive work on CS education's lack of inclusivity, there are a limited number of previous studies that investigate how to improve inclusivity, and even fewer in online computer science courses. The common themes in the small body of existing work on improving online CS education's inclusivity are ways that the instructors, prerequisites, or course advertising can help.

For example, work from Kizilcec's lab found that women in online learning tend to enroll in courses that are taught by female instructors and are less rigorous [42]. They noted that the preference for less rigor and fewer prerequisites may be due to a preference for meeting all requirements and expectations for success in the course, and pointed out that these preferences may be mitigated by clearly communicating expectations. Kizilcec's lab also found that having two instructors, one man and one woman, helped retain women in online computer programming courses but having only a woman instructor was met with negative reactions from some of the women in the course [41]. Work from that lab also found that adding gender-inclusive elements to course presentation increases women's enrollment in STEM courses [40]. For example, Cheryan et al. found that classroom decoration impacts women's interest and success in computer science—even in virtual classrooms [18]. In particular, having more neutral elements such as nature pictures is better for women than having elements that are perceived as masculine or stereotypical for computer scientists (e.g., action movie posters). All of these studies show ways that incorporating

gender-inclusive elements can help women to feel comfortable in online computer science. However, these studies focus mainly on course advertising or presentation, not on how curriculum changes themselves can improve both feelings of inclusion and acts of inclusion, by students in online CS education.

4. Approach

We are working on an emerging approach we term "Embedded Inclusive Design". The essence is to embed elements of inclusive software design into the curricula of mainstream CS courses.

Toward this end, we have developed three curricular interventions for asynchronous online CS education. *InclusivityFacets*: The first curricular intervention is a set of activities to enable students to learn the GenderMag cognitive styles (termed "facets" in GenderMag literature). *InclusivityHeuristics*: The second is a set of activities to enable students to learn the GenderMag heuristics based on these cognitive styles and use the heuristics to evaluate technology. *InclusivityDesign*: The third is a set of activities to enable students to improve the inclusivity of the technology they create, using the GenderMag heuristics. All interventions included mechanisms to assess student learning of inclusive design concepts from these interventions. Table 1 enumerates each activity and the interventions to which they contributed.

We hypothesize that, because inclusive design will be integrated with what students are learning as part of their major, these curricular interventions will impact online CS education in these primary ways: it will positively impact online CS students' feelings of belonging in the major; it will positively impact online CS students' inclusiveness toward other students in the major; it will positively impact online CS students' attitudes toward diverse users of software products.

4.1 Curricular Materials

Table 1 enumerates the activities our interventions used, and here we describe the curricular materials supporting those activities.

Course	When	#	Activity	Who Involved	Intervention(s)
CS-DB	Week 7	1	Exploration	Individual	InclusivityFacets, InclusivityHeuristics
	Week 7	2	Extra Credit Assignment (reflection / application)	Individual	InclusivityFacets, InclusivityHeuristics
CS-SE	Weeks 1+2	3	Exploration	Individual	InclusivityFacets, InclusivityHeuristics
	Weeks 1+2	4	HW1 (facet reflection)	Individual	InclusivityFacets
	Weeks 1+2	5	HW1 (design / evaluate / revise)	Individual	InclusivityHeuristics, InclusivityDesign
	Weeks 1+2	6	Learning Quiz	Individual	InclusivityFacets, InclusivityHeuristics
	Weeks 1+2	7	Team Facet Discussion	Team	InclusivityFacets
	Weeks 5+6	8	HW3 (integration with others' designs)	Team	InclusivityDesign
	Weeks 5+6	9	Peer Heuristic Evaluation, HW3 (design revisions)	Classmates, Team	InclusivityHeuristics, InclusivityDesign
	Weeks 9+10	10	HW5 (climate and users reflection)	Individual	InclusivityFacets, InclusivityHeuristics
	Weeks 9+10	11	Course Feedback (extra credit cognitive styles reflection)	Individual	InclusivityFacets, InclusivityHeuristics

Table 1. **Summary of curricular interventions**. CS-DB students experienced two interventions (InclusivityFacets and InclusivityHeuristics) implementations through an extra credit assignment and exploration during one week of the course. CS-SE students experienced all three interventions (added InclusivityDesign) through different implementations, spanning the entire course. All activities are available in the Appendix. Activity# serves as an ID; we refer back to these throughout this paper.

Exploration (Activities1,3): The first activity in each course as an interactive exploration to learn the GenderMag Heuristics. We devised the GenderMag Heuristics to fix gender inclusiveness issues in software, deriving them from the evidence-based GenderMag facets [12-14, 36]. At the time of this study, there were nine heuristics to support the cognitive styles of all three GenderMag personas (which we have since reorganized into eight heuristics). The nine were:

(1) Explain what new features do, and why they are useful
(2) Explain what existing features do, and why they are useful
(3) Let people gather as much information as they want, and no more than they want
(4) Keep familiar features available
(5) Make undo/redo and backtracking available
(6) Provide ways to try out different approaches
(7) Communicate the amount of effort that will be required to use a feature
(8) Provide a path through the task
(9) Encourage mindful tinkering

Fig. 1. **Quiz widget from Cognitive Style Heuristics exploration**. Provided to CS-DB and CS-SE students as a low-stakes way to check their understanding of the cognitive styles content.

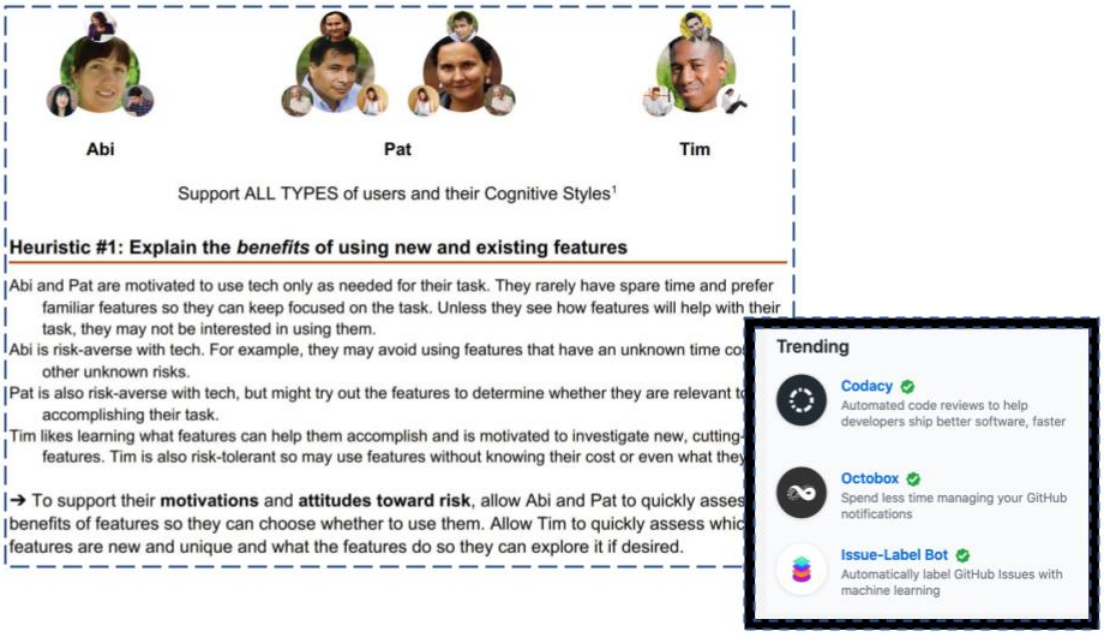


Fig. 2. **One of eight Cognitive Style Heuristics.** (Left): Each Heuristic has a short summarizing title, explanation of how it supports each persona, and a recommendation for how to apply to software design. Current version is shown. (All versions are in the Appendix.) (Lower Right): Example of applying Heuristic #1 by briefly explaining the benefits of each feature.

We created an interactive exploration that students used to learn these heuristics as part of InclusivityFacets and InclusivityHeuristics. This exploration included descriptions of each heuristic and examples of applications to show how facet values may affect software and how to design for diverse users (Figure 2). It also included the GenderMag personas (Figure 3) as part of InclusivityFacets to help students understand the cognitive style spectrum that the heuristics support. Additionally, embedded quiz questions (Figure 1) gave students immediate feedback on their understanding of the reading.

During CS-DB and before CS-SE, we iterated on the exploration. We renamed GenderMag Heuristics to Cognitive Style Heuristics to communicate that GenderMag is about cognitive diversity but still retained the discussion of gender in the exploration. The Cognitive Style Heuristics and GenderMag Heuristics are the same and the names can be used interchangeably but we will continue to call them Cognitive Style Heuristics unless reference to the name change is necessary.

We also replaced the full-length persona documents with abbreviated versions (Figure 3) (personas are not needed for a heuristic evaluation, which is meant to be a "discount" usability evaluation technique [54]). This change was made to help students focus on the personas' facet values for InclusivityFacets by removing extra information. Additionally, in the revised exploration, Pat (middle-spectrum) persona helps to emphasize that individuals often reflect a mixture of Abi and Tim facet values.

Finally, the instructor and teaching assistants added profiles of their cognitive styles (Figure 4 (Right)). One of the primary motivations for this change was to increase social presence which allows individuals to represent their full personality in communication [29]. Building social presence in the course is important to both successful discourse and climate. For example, these profiles provide examples to guide students on later discussions. They also allow the teaching team to share their diversity and create a safe environment with open communication for a better climate [27]. Beyond social presence, the profiles provide further evidence that individuals can identify as varying combinations of Abi, Pat, and Tim.

The first two CS-DB classes offered the GenderMag exploration (Activity1) prior to these revisions. The third CS-DB class and CS-SE included the revised Cognitive Styles Heuristics. Additionally, while the Cognitive Styles Heuristics Exploration was required for CS-SE, the exploration was offered for extra credit to students in CS-DB. CS-DB students also had the option of completing an exploration for the general usability Nielsen's Heuristics [30, 55] instead. The versions of the heuristics and descriptions used for our interventions are in the Appendix.

Extra-Credit Assignment (Activity2): For extra-credit, CS-DB students could complete a 400-word reflection discussing their thoughts on the GenderMag or Nielsen's exploration and giving an example of how they would apply what they learned. The structure of the assignment was defined by a CS-DB instructor, who had used the structure for other extra-credit assignments.

Homework 1 Facet Reflection (Activity4): Within the first homework assignment of CS-SE, students completed an individual reflection to identify their own facet values. Students were asked to self-identify each of their five facet values. The assignment also asked students to identify one way they are like the Abi persona and one way they are like the Tim persona in order to again emphasize that individuals can be a mixture of facet values. Finally, students identified how cognitive styles may impact their software usage.

Fig. 3. **GenderMag persona snippets** as presented to CS-SE students as part of learning exploration (Activity3). Adapted from the full GenderMag personas [13]. Multiple races, genders, and ages represented to help students see they can be like any persona.

Homework 1 Design Process (Activity5): As part of the same homework, students individually completed and evaluated a paper prototype for an application they would create throughout the term. Students were not required to make changes but could lose points if they did not adequately justify how their design reflected three heuristics: (#2) Explain what existing features do, and why they are useful, (#3) Let people gather as much information as they want, and no more than they want, and (#4) Keep familiar features available.

Learning Quiz (Activity6): Students in CE-SE took a learning quiz about applications and aspects of the GenderMag exploration. They were able to take the quiz unlimited times as a low-stakes measure of their learning. The quiz was designed to help students understand the scope and usage of the Cognitive Style Heuristics. Questions on the quiz included: "Applying the Cognitive Style Heuristics will ONLY make software more usable to women. True or False", "Which of these on-screen messages would best help a user understand a new feature?", and "Why might Pat want to see what existing features do and why they are useful?".

Team Facet Discussion (Activity7): As part of a group reflection, students discussed the exploration and shared two of their facets with their term-long team members on a discussion board, as in Figure 4 (right).

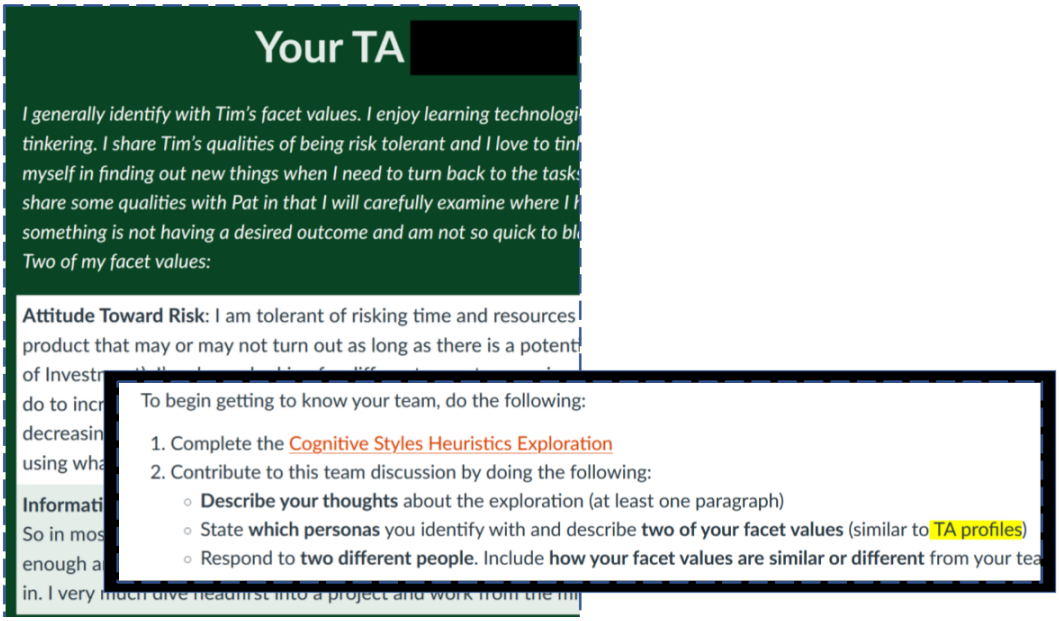


Fig. 4. **(Left): Cognitive style profile of a CS-SE teaching assistant.** Includes TA's persona and facet value identifications. Instructor and four TAs provided profiles like these to help students learn the GenderMag course material, give students examples of talking about cognitive styles, and to help students see the diversity within the instructional team. (**Right): Team discussion assignment from CS-SE.** Students had the first two weeks of the course to discuss their persona identification(s), facet values, and similarities/differences. (See the Appendix for full version.)

Homework 3 Design Integration (Activity8): The individual design process (Activity5) led into a group assignment where students coordinated with their team to combine prototypes (each a different feature) into a GUI prototype for an entire application.

Peer Heuristic Evaluation, Homework 3 Design Revision (Activity9): Students then collaborated with non-teammates in a peer review discussion. On a discussion board, students posted their work for critique by classmates outside the team, evaluated another team's work, and received feedback with which they revised their GUI design. Reviews included evaluations for 8 of the 9 Cognitive Style Heuristics (#2-9) as well as a suggestion and something they liked about the design.

Homework 5 Climate and Users Reflection (Activity10): During Weeks 9 and 10, students individually reflected on the curriculum as part of their final homework assignment. They were asked how the cognitive styles discussion had affected interactions with their teammates and how learning about cognitive styles affected the way they view users.

Extra-Credit Course Feedback (Activity11): For extra-credit at the end of the term, students could leave feedback further reflecting on the cognitive styles content. Students were asked about its positive / negative effects on them and why the content might / might not be useful in the future.

The curricular materials are available in the Appendix.

5. Methods

We investigated our research questions using these three curricular interventions in four online classes spanning nine months (three CS-DB classes, one CS-SE class), via Action Research. Action Research is an iterative and longitudinal form of field research where the goals of solving a problem and obtaining scholarly knowledge about that problem are intertwined. One implication is that the division between researchers and participants is also blurred—researchers can act as participants and participants can act as researchers [34, 46, 68]. In our investigation, the four classes were taught by four different faculty members. All four experienced the interventions from a faculty perspective (participant role) and conducted the interventions (researcher role), and two also contributed to the interventions by creating/improving them (researcher role). Another implication of Action Research is the treatment is not "fixed," but rather is iteratively improved in response to data arriving over time. To achieve rigor, Action Research often makes extensive use of triangulation to affirm credibility and validity of results, providing conclusions only when multiple data sources/perspectives/methods produce the same findings. We return to our triangulation mechanisms in Section 7.

Our participants were students in the four classes, and our data came from their work products (Table 1) and questionnaires. From an education perspective, each assignment was required of all students (or available to all students, if extra-credit). From a research perspective, students who did the assignment could opt in to having their work used in our research, for which they were compensated with a $10 Amazon eGift Card. As per IRB and university policies, the instructors/TAs involved in the research did not know which students' work had been opted-in until after final course grades were turned in.

	Men	Women	Agender	FTM	Total
CS-DB	20	8	1	1	30
CS-SE	22	12	0	0	34
Combined	42	20	1	1	64

Table 2. **Participant count by self-identified gender for each course**. Participants self-identified their gender in an open-ended survey question.

5.1 Data and Analysis

We collected data on the effects of the curricular interventions as described in Section 4 using a post-questionnaire for CS-DB, and pre- and post-questionnaires for CS-SE. The questionnaires, adapted from the NCWIT Student Experience of the Major survey [52], asked students about their perception of the curriculum, feelings of inclusion, and perceptions of the CS major. The questions students in CS-DB were asked included why they liked / disliked Activity1, why they would use / not use the material learned, what course content made them feel included / excluded, and if the assignment increased their interest in computer science. CS-SE students were asked about topics including how likely they are to complete a CS major / minor, how represented they feel in CS, what they would take into consideration when designing software, and what aspects of the course made them feel included / excluded. Both questionnaires can be found in the Appendix.

We also collected data through students' assignment submissions in CS-SE and extra credit submissions in CS-DB. Additionally, in CS-SE, we collected students' course feedback every two weeks. Data collected from CS-DB also includes data from students who completed the extra-

credit assignment on Nielsen's Heuristics. With this information, we can look not only at the impact of the GenderMag curriculum, but also compare these results to those of the Nielsen's participants.

Our quantitative analyses use descriptive statistics only. Inferential statistics would not be appropriate in this investigation because no "controls" were in place to cleanly isolate variables, and the number of students opting into the research from any one class offering was very small.

Although some data presented numerically in this paper was collected quantitatively (e.g., with Likert-style questions), some was free-format, which we qualitatively coded to allow numeric summaries. We segmented the qualitative data (open-ended questionnaire responses and written student assignments) so each response was a segment. Next, as our work crosses through several areas of research (education, online education, HCI, CS, social sciences. . .), we followed Hsieh & Shannon's [37] conventional content analysis approach, which is appropriate for describing phenomena that cannot be well-encompassed by existing research, to develop our codeset. Working bottom-up, categories of phenomena emerge from the data (we used affinity diagramming [31] for this process). Two researchers independently coded small portions of the data to check agreement. We achieved >=80% agreement on >=20% of the data for each dataset (Jaccard method) [66]. Given this level of consensus, one coder coded the remainder of the data. All code sets can be found in the Appendix.

6. Results

To investigate how our new approach affected different diversity outcomes, we analyzed students' coursework (Activities1-10 in Table 1) and questionnaire responses to investigate their feelings of inclusion and inclusivity attitudes toward others.

6.1 Feeling Included and Including Others

6.1.1 Learning About Me. In CS-SE, students were asked to self-identify their facet values (listed in Table 4) in their first homework assignment. Students also reflected on their cognitive styles in Activity4 and Activity10 (the two self-reflection activities). For our analysis, we labeled participants to be Abi if they reported at least 3 facet values like Abi, Tim if they reported at least 3 facet values like Tim, and Pat otherwise. The result of this labelling system can be seen in Table 3.

These activities helped students of varying facet values report new understanding of or even appreciation for their cognitive styles and software use.

> P30774-SE-Tim*: "I actually bring to software a bigger set of values than just being good at it or "getting it." That is the biggest take-away from this reflection on my facet values."*
> P31369-SE-Abi: *"Learning how strongly I identify with Abi makes me realize why I get annoyed when I use software that doesn't have clear explanations for things or labels showing what features do and are useful for."*
> P30683-SE-Tim: *"I never thought of myself as having some of the aspects I read about in Abi and Pat, but I am definitely somewhat resistant/have trouble with learning new technologies"*
> P30097-SE-Tim: *"Identifying my facet values helped me [understand which features of technology] are most helpful [for my learning. . .] the most successful I have been. . .was when I just jumped right in. . . <Facet: Tech learning style (tinkering)>"*

	n	Abi	Tim	Pat
Women	12	42%	8%	50%
Men	22	18%	41%	41%
Overall	34	27%	29%	44%

Table 3**. Persona self-identifications by gender for the CS-SE students.** Women skewed closer to Abi, and men skewed toward Tim. Almost half the participants were Pat's, and the other half was almost evenly divided between Abi's and Tim's. Abi: Participant self-identified with >=3 of Abi's 5 facet values. Tim: >=3 of Tim's facet values. Pat: Everyone else. The frequencies of specific facet values with which they identified are shown in Table 4.

Facet	Facet Value	Frequency
Attitude toward risk	Averse (Abi)	16
Information processing	Selective (Tim)	16
Learning style	Tinkering (Tim)	15
Information processing	Comprehensive (Abi)	14
Motivations	Task (Abi)	14
Computer self-efficacy	High (Tim)	13
Attitude toward risk	Tolerant (Tim)	11
Learning style	Process (Abi)	8
Motivations	Tech Interest (Tim)	7
Computer self-efficacy	Low (Abi)	2

Table 4. **How many CS-SE students self-identified as having each GenderMag facet value.** Part of Activity4 (Table 1)**.** Students self-identified with Abi and Tim facet values about as often; they were cognitively diverse. Students identifying as having both values of a facet are not counted here.

Following Activity4, CS-SE students completed a team discussion of facet values (Activity7) where these realizations continued to appear.

> P30683-SE-Tim: *"I agree that the module helped me gain a different perspective on being tech-savvy. Honestly, I used to think maybe it was just stubbornness or giving up too easily… But each user gets to decide how they want to interact with a technology, and it's okay that some people might feel intimidated by a new technology or just decide that it isn't worth the effort to them."*
>
> P30774-SE-Tim: *"I have always felt uncomfortable with the idea of my students and younger generations as "digital natives," and I think this exploration helped to solidify just why."*

Further, when asked how the content positively impacted them during their end-of-term evaluation, students recognized the benefit of learning their cognitive styles and better understanding themselves.

> P30018-SE-Abi: *"I have been taking CS classes for 3 years now and like many of my classmates felt like an imposter because I wasn't a tinkerer. These cognitive styles point out the benefits of my caution as well as validate them to myself and amongst my peers"*
>
> P30683-SE-Tim: *"[The cognitive styles content] has made me aware of my usual patterns and now I can be reflective about them and maybe try different things (like tinkering)."*

When students shared their facet values and insights with others, they also were able to validate their experiences through interacting with students who had similar facet values. This occurred even when students had differences in facet values. For instance, although only about 27% of participants were Abi's, 77% identified as having at least one Abi facet value.

> P30018-SE-Abi: *". . . excited to work on a team with a fellow Abi. . . [anticipating] running through the whole process start to finish. <Facets: Information processing, Tech-learning style>"*
>
> P33731-SE-Abi: *". . . our [facet values] are nearly identical! Like yourself I can be very timid about new programs and apps when I don't fully understand what all is going on. <Facets: Computer self-efficacy, Risk>"*

6.1.2 Exploring Each Other's Differences. Besides finding commonalities, students also explored facet value differences within their teams. For example, P30683-SE, who self-identified as having three Tim facet values and two Abi facet values, chose to point out their Abi learning style.

> P30683-SE-Tim: *"Wow! It's crazy to read about your willingness to use LaTeX . . .I really hate tinkering. <Facet: Tech learning style>"*

This also resulted in recognizing benefits of other approaches or even a desire to change one's own facet values.

> P36676-SE-Abi: *"I think Tims have some enviable qualities. I imagine the drive to learn new technologies has many positive outcomes."*
>
> P31766-SE-Pat wanting to become more Abi-like: *"...working on being more comprehensive...<Facet: Information processing>"*
>
> P35173-SE-Tim with Abi's Information processing style, wanting to become even more Tim-like: *"[My comprehensive style is] probably often a detriment <Facet: Information processing>"*

From these discussions, strong "includedness" results emerged. As Table 6's leftmost section (bottom row) shows, our implementations of the InclusivityFacets intervention (Activities3,4,6,7,10,11) made 88% of students feel included, and made nobody feel excluded. Disaggregating into Abi-like, Tim-like, and Pat-like CS-SE students (top rows), and by gender (middle rows) shows very high results for every persona-type and every gender. In summary, the InclusivityFacets intervention helped everyone, but Abi and women especially seemed to benefit.

Note the rightmost section, regarding simply learning about cognitive styles (e.g., in Activity3) (as opposed to discussing them, e.g., in Activity7). As that section shows, for Tim's, Pat's, and men, learning alone was nearly as effective as discussing them. However, for Abi's and women, discussing cognitive styles in addition to learning them had a pronounced positive effect.

	n	Very included	Somewhat included	Neither	Somewhat excluded	Very excluded	Included	Excluded
Abi	9	56%	22%	22%	0%	0%	78%	0%
Tim	10	70%	20%	10%	0%	0%	90%	0%
Pat	15	53%	27%	20%	0%	0%	80%	0%
Women	12	58%	25%	17%	0%	0%	83%	0%
Men	22	59%	23%	18%	0%	0%	82%	0%
Overall	34	59%	23%	18%	0%	0%	82%	0%

Table 5. **Effects of LEARNING COGNITIVE STYLES on CS-SE participants feeling included/excluded. Right**: Almost everyone felt included and nobody felt excluded. **Left**: Results leaned more toward very included than somewhat included. Tim's felt especially included. Maximum row values within section are highlighted.

	n	Included (Discuss)	Excluded (Discuss)	Very inclu.	Some inclu.	Neither	Some exclu.	Very exclu.	Included (Learn)	Excluded (Learn)
Abi	9	100%	0%	67%	33%	0%	0%	0%	78%	0%
Tim	10	90%	0%	40%	50%	10%	0%	0%	90%	0%
Pat	15	80%	0%	27%	53%	20%	0%	0%	80%	0%
Women	12	92%	0%	50%	42%	8%	0%	0%	83%	0%
Men	22	86%	0%	36%	50%	14%	0%	0%	82%	0%
Overall	34	88%	0%	41%	47%	12%	0%	0%	82%	0%

Table 6. **Effects of COGNITIVE STYLES on CS-SE participants' feelings of inclusion.** Students' post-questionnaire responses after the term showed almost everyone felt included by discussing cognitive styles with teammates, but Abi and women especially seemed to benefit. (Recall that only two genders participated in the CS-SE investigation.) **Left**: Summary, where "Included" = Somewhat/very included, and "Excluded" = Somewhat/very excluded. **Middle**: Detailed breakout for discussing cognitive styles. **Right**: Summary of students' responses for simply learning cognitive styles, as per Table 5. Maximum row values within section are highlighted.

6.1.3 Teamwork: Toward "I'm OK, You're OK" [33]. Learning about their teammates also helped students to understand others and create more inclusive teams. Students noted instances of being included by and inclusive of teammates with different cognitive styles:

> P36170-SE-Pat: *"[My teammates understand that we] tend to work differently [and thus we were] less demanding on each other."*
>
> P32624-SE-Pat: *"my teammates' information process style and learning style were a 180 degree pivot from the way I processed information … I think the biggest change in my interactions happened in sprint 4 where I became more understanding of teammates that tend to tinker"*

In essence, students began to understand that their teammates' facet value differences did not equate to differences in ability.

> P30683-SE-Tim: *"[I'm] resistant [to] new technologies...[but my team understands that I'm not] being lazy...some people just aren't really tinkerers."*

These shifts in understanding helped teams be more efficient and cooperative. P30018-SE-Abi explained that they were able to use cognitive styles to better communicate with their team:

> P30018-SE-Abi: *"Learning about the cognitive styles made me see people by their strengths and what they brought to the table. I enjoyed acknowledging our cognitive styles at team meeting when issues became more complex. I liked that they introduced language that gave us the confidence to admit when we were in an area that felt overwhelming or something came easy to us"*
>
> P30018-SE-Abi: *"[Cognitive styles] were a non-intimidating way to neutrally express our confidences and capabilities."*

And it helped them to balance the design work and solve problems:

> P37307-SE-Abi: *"if we were looking to improve our project with respect to a Tim-type user, we already had an understanding of which teammates could relate to that user the most."*

	n	Team			Non-Team		
		Included	Neither	Excluded	Included	Neither	Excluded
Abi	9	89%	11%	0%	45%	33%	22%
Tim	10	90%	0%	10%	40%	60%	0%
Pat	15	80%	7%	13%	27%	46%	27%
Women	12	66%	17%	17%	17%	50%	33%
Men	22	95%	0%	5%	45%	45%	10%
Overall	34	85%	6%	9%	35%	47%	18%

Table 7. **Effects of team/class interactions on CS-SE students feeling included/excluded.** Students' responses to post-questionnaire showed that students' feelings of inclusion were very high in interactions with teammates, and far more so than with their other classmates. **Left**: Effects of TEAM interactions. **Right**: Effects of NON-TEAM CLASSMATE interactions. Maximum row values within section are highlighted.

	n	Represented?		Belong?		Likely to complete		Cognitive Styles increased interest?
		Pre	Post	Pre	Post	Pre	Post	
CS-DB	8							0.59
CS-SE	34	0.67	0.68	0.81	0.88	0.91	0.99	0.75

Table 8: **CS major/minor climate. Left:** Feelings of representation, belonging, and likelihood to complete CS major/minor increased for CS-SE participants between the beginning and end of course (not measured for CS-DB). **Right:** Both CS-DB participants and CS-SE participants were more interested in the CS major/minor after using GenderMag Heuristics. (Note: For CS-DB, 8 of 30 participating students used GenderMag Heuristics; the others used Nielsen's). Maximum row values within section are highlighted. *Units*: Position on Likert scale from 0 (least) to 1 (most).

6.1.4 Climate by the Numbers. In terms of numbers, when looking at the CS-SE post-questionnaire, students of every persona and every gender reported feeling included by their teammates (Table 7). These numbers are in stark contrast to their feelings of being included by their non-team classmates, which did not reach even 50% for any persona-type or any gender. Women felt the least included, but their inclusion rate among their teammates was still a dramatic increase over their inclusion rate by other classmates. In combination with Table 6, this suggests that the InclusivityFacets intervention was directly responsible for these effects.

Further, as Table 8 shows, the implementation of InclusivityFacets positively impacted the climate. For both CS-DB and CS-SE, students reported that learning cognitive styles increased their interest in CS and increase the likelihood of them completing their CS major/minor. This result is particularly interesting as the CS-SE course is a 3rd year or junior level course. While students in the course can be earlier in their CS education, students who usually take this course are nearing the end of their degree program. Despite students nearing graduation, a time when they are expected to be fully committed to their program, likelihood to complete the major/minor showed the greatest increase: nearly 9%.

6.2 Did GM affect feelings of exclusion?

In addition to feelings of *inclusion*, we also investigated if the interventions impacted feelings of *exclusion*. In CS-SE we found little indication that the interventions contributed to feelings of

exclusion. Tables 5 and 6 show that CS-SE students reported only being included by learning and discussing cognitive style heuristics. The highest levels of exclusion reported by students in CS-SE was in relation to interactions with non-team classmates (Table 7). In fact, this gap between teammates and non-team classmates may be another result of the intervention as sharing and discussing cognitive styles improved in-team dynamics as discussed previously.

However, in CS-DB, some students did feel excluded by discussions of gender in the GenderMag exploration. One participant who self-identified as agender said that GenderMag excluded non-binary participants.

> P0514201915-DB: *"I didn't like how stereotyping and nonbinary-exclusive the heuristics were. It kind of irritated me."*

Similarly, another student commented on the need for gender inclusivity:

> P0513202057-DB: *"The heuristics might be dated now that discourse and acceptance of gender fluidity and non-binary people is the norm."*

In the first version of the CS-DB course, we had neglected to include the Pat persona, and this may have contributed to the feelings of stereotyping/"binaryness" by some students. Based on this feedback, we began including the Pat persona (middle-spectrum) into the exploration and renamed the exploration (as discussed in Section 4.1). After these changes, students no longer reported feeling excluded.

6.3 Considering Diverse Users

Beyond students' feelings of inclusion and inclusivity within the classroom, we also hypothesized that the three interventions would increase participants' awareness and inclusion of diverse users.

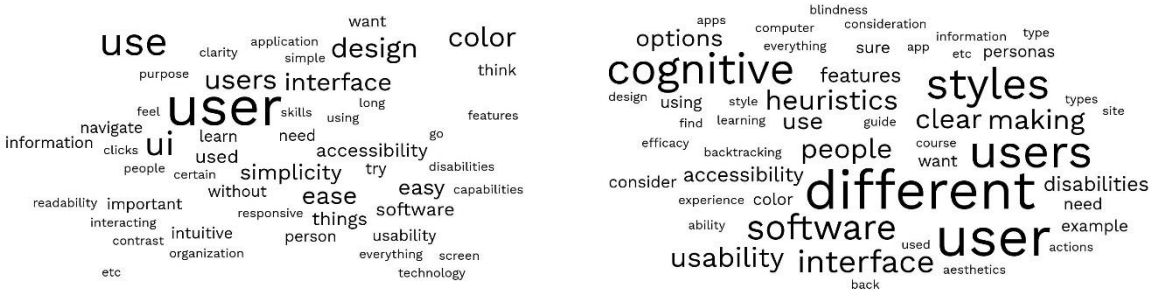
During the three interventions and afterward, students began to talk about the need to support diverse users. In fact, as shown in Table 9, *every one* of the students in both courses who learned the cognitive style heuristics commented on this in one way or another. In CS-SE, the majority of students began to talk about this with their groups in Activity 7 and carried this idea through to the post questionnaire. Students began to realize that users are not all the same and that creating software for a generic "the user" can create bias.

> P33965-SE-Tim: *"All of us were more aware of the different types of users that would be using our website... We never had a debate about whether such a focus was necessary, and that was definitely because of the discussion we had on cognitive styles."*
>
> P0513202057-DB: *"I would refer to [Cognitive Style Heuristics] as a software developer to ensue that the software I crease is user-friendly for all kind of users. I would want to make sure I am not being biased"*

Figure 5 shows the different words that students used when discussing what they would take into account when designing software based on the pre- and post-questionnaires. Before the interventions, common words were "user", "ease", and "color". After the interventions, students used words such as "different", "usability", and "cognitive".

Course	Source	% of students mentioning user diversity
CS-DB	Extra Credit Assignment (Activity 2)	88%
CS-DB	Post-Questionnaire	63%
CS-DB	Course Total	100%
CS-SE	Team Facet Discussion (Activity 7)	91%
CS-SE	Feedback Assignments (including Activity 11)	74%
CS-SE	HW5 (Activity 10)	74%
CS-SE	Post-Questionnaire	68%
CS-SE	Course Total	100%

Table 9. **CS-DB and CS-SE students whose work products mentioned user diversity.** From all data sources throughout the term. Bottom row for each course computes union of students counted in that course's rows above. By the end of the term, 100% of the participating students in all three CS-DB classes and in CS-SE had brought up this point. Activities refer to Table 1.



(a) Before learning GenderMag Heuristics (b) After learning GenderMag Heuristics

Fig. 5 **GUI design priorities of CS-SE students** at the beginning of the course (left) versus the end (right). Students went from talking about "color" (7 counts), "ease" (6), and "simplicity" (5), to "different" (13), "styles" (10), and "cognitive" (10).

In comparison, CS-DB participants who completed the Nielsen's exploration talked very little about user diversity or the need to support diversity. Out of the 22 Nielsen's participants, about two thirds did not mention user diversity at all. If they discussed users, it was a generic user and those who discussed diversity discussed only users' levels of experience with technology. The following quotes detail a few of these differences.

> P0512200012-DB-Nielsen: *"...tailoring a program to... beginners [and also] advanced users"*
>
> P0514201819-DB-GenderMag: *"...people with different problem-solving instincts and software-interaction behaviors"*

The interventions also taught students that users are not just diverse but different from themselves. In addition to discussing different users, 53% of CS-SE participants began to explicitly comment on how they are different from users.

> P33695-SE-Tim: *"Now when I think of users using a piece of software I don't picture them as I am… I am much more aware that there are others like Abi and Pat."*
>
> P33842-SE-Pat: *"Now, I feel more considerate of others and the way they interact with software too, and I think I'm going to be more patient and understanding of those who are more risk-averse than I am."*

In CS-DB, 13% of GenderMag participants commented on this point. Similar to the results for discussions of diversity, fewer Nielsen's participants acknowledged these differences – only 5%.

By the end of the term, many CS-DB and CS-SE students were expressing a sense of responsibility for technology problems diverse users might experience. For example, upon learning that low-self-efficacy users might blame themselves [13] when technology does not perform as expected (recall Figure 3), P30683-SE was vehement that the blame lay with the developer, not the user.

> P30683-SE-Tim: *"That's not right! I felt a sense of responsibility to users like these. <Facet: Computer self-efficacy>"*
>
> P37987-SE-Pat: *"if you do not fit [the user's] cognitive type, then you may not fully understand how they interpret [a feature]"*
>
> P30774-SE-Tim: *"[when I see others] struggle with technology, especially in this context of the pandemic, I will view them with more compassion"*

7. Discussion

7.1 Results Triangulation

Consistent with Action Research methods, we safeguarded the reliability of our results through extensive use of triangulation, which we enumerate in Table 10. As the table shows, every result was cross-confirmed across multiple sources of evidence.

Each major result occupies a column. The cells under the first two columns show the activities and/or questionnaires that showed evidence for each Research Question; the last three columns do so for each curricular intervention. Summing up the evidence from CS-DB and CS-SE, the total results for each research question are evidenced by 5 data sources, and for each intervention were evidenced by 2 to 7 data sources. In summary, evidence from multiple courses and data sources pointed to the same results for all research questions and interventions.

	Results by RQ		Results by Curricular Intervention		
	RQ1-climate	RQ2-users	InclusivityFacets	InclusivityHeuristics	InclusivityDesign
CS-DB	Q	Q	2,Q	2,Q	N/A
CS-SE	10,11,Q,Q	5,10,Q,Q	4,7,10,11	5,9,10,11	5,9

Table 10. **Results triangulation**. Row 1: Triangulation of results within the CS-DB course. (Note that InclusivityDesign is not applicable to CS-DB). Row 2: Triangulation of results within the CS-SE course. Each number refers to an activity number whose work product produced the result for the column header above it (Q refers to a questionnaire result instead of an activity result). **Left:** Triangulation of results by Research Question. **Right:** Triangulation of results by curricular intervention.

7.2 Perspectives on Gender, Feminism, and Responsible CS—and Lessons Learned

The impetus behind our approach is the goal of supporting pluralism in the technologies that CS students, as future CS practitioners, create. As Bardzell explained, supporting pluralism means creating technologies that "resist any single, totalizing, or universal point of view" [5], i.e., emphasizing attention to individual differences within genders rather than universality. The concept of pluralism is a central tenet of feminist HCI as per Bardzell [5]. Under differing vocabularies, the same concept is also widely espoused in others' views on HCI feminism, queer feminism, and queer theory (e.g., [15, 47, 63, 70]; summarized in [9]). The concept of pluralism is also a central idea behind work in universal or inclusive design. All of these ideologies embrace the common theme of avoiding making technology that attempts to force diverse individuals to all problem-solve and work with technology in one and only way.

Like researchers in the "Responsible CS" and "critical CS" schools of thought (e.g., [19, 26, 44]), we view integrating such issues into modern CS education to be important and necessary. CS students, as future CS practitioners, cannot build a digital world that honors diversity and pluralism if CS education does not offer them skills to do so. The interventions we presented attempted to do so, and our results so far are encouraging.

Even touching upon gender equity issues can seem controversial, and three students explicitly shared their views of the inadvisability of our interventions. As Table 11 shows, one expressed discomfort with the inclusivity of the interventions themselves, three said that the interventions invoked gender stereotyping, and one implied that talking about gender at all was unnecessary. However, several others applauded the interventions, for reasons ranging from feeling from more included to being able to build better software to becoming a more responsible computing professional; the table shows a few of the positive responses.

From these views we derived three lessons.

Integrate throughout the term: First, the interventions worked best when integrated throughout the course as in CS-SE, not just in a single assignment as in CS-DB. Not only did the integration enable CS-SE students to learn more skills for software creation, the term-long experience also produced very strong climate impacts (e.g., recall Table 8).

Incorporate in team discussions: Although too late for data gathering, another class of CS-SE using these interventions has just finished. In this post-study offering, team discussions were not part of the interventions, and the students did not seem to have gained as much comfort with the materials as a result.

Emphasize the research behind the interventions: Over time, we have become more explicit about the evidence behind these interventions, with several pointers and citations pointing the way, and this has been positively received. However, we have not emphasized the research investigating whether GenderMag encourages stereotyping. (In fact, it reduces stereotyping [36].) Future terms will bring this evidence forward, to see whether that helps allay stereotyping concerns.

Student and View	Pro/Con	Why Pro	Why Con
Woman-DB: There are probably developers ... not using these heuristics limiting the ...effectiveness of their software.	→	Better CS	
Man-SE: [when I see others] struggle with technology...I will view them with more compassion... I am happy...this class did more to diversify representation.	→	Responsible CS, Better at CS	
FTM (Transgender)-DB: I noticed the gender inclusion right away... I appreciate that the industry is finally addressing these issues... I want to see that reflected in my education... [It's] good to... increase your user base...	→←	Responsible CS, Better at CS, Includes me	
Agender-DB: ...bring to mind the gender stereotype orientedness of the heuristics [but I'll] disassociate them from the idea of gender and just think about different user types... nonbinary-exclusive... irritated me.	←	Better at CS	Stereotyping, Excludes me
<unknown gender>-SE: I feel like we have a tendency to tie gender to things unnecessarily... I wish we would stop fueling stereotypes.	←		Unnecessary, Stereotyping
<unknown gender>-SE: I think the cognitive styles content can be harmful...the concept may be used to stereotype...	←		Stereotyping

Table 11. **Students' pro (→) vs. con (←) views on these interventions**. We gathered these statements from our data (Section 5) and from anonymous course evaluations. Thus, these views represent those of >800 students in all course offerings, not just the opting-in students. This table shows all three CON comments from these >800 students, and a subset of the PRO comments.

7.3 Limitations

No empirical study is perfect. One reason is the inherent trade-off among different types of validity [76]. Field studies, including Action Research studies, achieve real-world applicability, whereas controlled studies achieve isolation of variables. Our Action Research study therefore had many uncontrolled variables, such as multiple courses with multiple instructors in multiple terms.

This leads to limitations in generalizability. Our four classes were offerings of only two courses, and our educational setting was (1) post-bacc CS students, who are different than post-secondary CS students; and (2) asynchronous online courses, which are different from face-to-face courses. Another generalizability limitation is that fewer than 10% of students in these classes opted in (out of >800 students). Thus, interpretations we made from our data might be different had we studied different students.

Limitations like these can only be addressed additional empirical studies using a variety of empirical methods in a variety of educational settings. Given these limitations, we do not view our results as being generalizable beyond the particular context of our investigation, but rather as encouraging evidence of the potential of these interventions.

8. Conclusion

In this paper, we have presented three new curricular interventions for online CS courses. Our educational setting was two courses (four classes) over nine months in an asynchronous online CS education program for post-bacc students. We then evaluated whether these interventions increased class climate and led students to honor their users' diversity. Among our results were:

- RQ1 (climate): Students gained new acceptance of themselves, reported positive impacts on team dynamics, and felt more included in the major.
- RQ2: (users, tech): Students came to recognize and respect their users' diversity.

Students' attitudes toward these interventions were generally quite positive, although a few raised issues that will require more improvement in our implementation of the interventions. Among their reasons for positive responses were the feeling that the interventions helped them to both create better software and to be a more responsible CS professional. Perhaps this is the most important outcome of all—accepting responsibility for the impacts one's software creations can have on diverse users.

References

[1] ACM/IEEE Joint Task Force on Computing Curricula. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. (20 December 2013). https://www.acm.org/education/curricula-recommendations Accessed: Feb. 2, 2020.

[2] Christine Alvarado, Yingjun Cao, and Mia Minnes. 2017. Gender Differences in Students' Behaviors in CS Classes throughout the CS Major. In *Proceedings of the 2017 acm sigcse technical symposium on computer science education.* 27–32.

[3] Manon Arcand and Jacques Nantel. 2012. Uncovering the nature of information processing of men and women online: The comparison of two models using the think-aloud method. *Journal of theoretical and applied electronic commerce research 7, 2* (2012), 106–120.

[4] Rachel Baker, Thomas Dee, Brent Evan, and June John. 2018. *Bias in Online Classes: Evidence from a Field Experiment.* Technical Report. Stanford Center for Education Policy Analysis. https://cepa.stanford.edu/sites/default/files/wp18-03-201803.pdf

[5] Shaowen Bardzell. 2010. Feminist HCI: taking stock and outlining an agenda for design. In *Proceedings of the SIGCHI conference on human factors in computing systems.* 1301–1310.

[6] Lecia J. Barker, Charlie McDowell, and Kimberly Kalahar. 2009. Exploring Factors That Influence Computer Science Introductory Course Students to Persist in the Major. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, TN, USA) (SIGCSE '09). Association for Computing Machinery, New York, NY, USA, 153–157. https://doi.org/10.1145/1508865.1508923

[7] Sylvia Beyer. 2014. Why are women underrepresented in Computer Science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. *Computer Science Education* 24, 2-3 (2014), 153–192.

[8] Brianna Blaser, Katherine M. Steele, and Sheryl Elaine Burgstahler. 2015. Including Universal Design in Engineering Courses to Attract Diverse Students. In *2015 ASEE Annual Conference & Exposition*. ASEE Conferences, Seattle, Washington. https://www.jee.org/24272.

[9] Samantha Breslin and Bimlesh Wadhwa. 2014. Exploring Nuanced Gender Perspectives within the HCI Community (*IndiaHCI '14*). Association for Computing Machinery, 45–54. https://doi.org/10.1145/2676702.2676709

[10] Michael Buckley, Helene Kershner, Kris Schindler, Carl Alphonce, and Jennifer Braswell. 2004. Benefits of using socially-relevant projects in computer science and engineering education. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*. 482–486.

[11] M. Burnett, R. Counts, R. Lawrence, and H. Hanson. 2017. Gender HCl and microsoft: Highlights from a longitudinal study. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).* 139–143. https://doi.org/10.1109/VLHCC.2017.8103461

[12] Margaret Burnett, Anicia Peters, Charles Hill, and Noha Elarief. 2016. *Finding Gender-Inclusiveness Software Issues with GenderMag: A Field Investigation*. Association for Computing Machinery, New York, NY, USA, 2586–2598. https://doi.org/10.1145/2858036.2858274

[13] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A Method for Evaluating

Software's Gender Inclusiveness. *Interacting with Computers* 28, 6 (10 2016), 760–787. https://doi.org/10.1093/iwc/iwv046arXiv:https://academic.oup.com/iwc/article-pdf/28/6/760/7919992/iwv046.pdf

[14] Margaret M. Burnett, Laura Beckwith, Susan Wiedenbeck, Scott D. Fleming, Jill Cao, Thomas H. Park, Valentina Grigoreanu, and Kyle Rector. 2011. Gender pluralism in problem-solving software. *Interacting with Computers* 23, 5 (07 2011), 450–460. https://doi.org/10.1016/j.intcom.2011.06.004arXiv:https://academic.oup.com/iwc/article-pdf/23/5/450/1875429/iwc23-0450.pdf

[15] Judith Butler. 2011. Gender trouble: Feminism and the subversion of identity. routledge.

[16] Shuo Chang, Vikas Kumar, Eric Gilbert, and Loren G Terveen. 2014. Specialization, homophily, and gender in a social curation site: Findings from Pinterest. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 674–686.

[17] Gary Charness and Uri Gneezy. 2012. Strong evidence for gender differences in risk taking. *Journal of Economic Behavior & Organization* 83, 1 (2012), 50–58.

[18] Sapna Cheryan, Andrew N. Meltzoff, and Saenam Kim. 2011. Classrooms matter: The design of virtual classrooms influences gender disparities in computer science classes. *Computers & Education* 57, 2 (2011), 1825–1835. https://doi.org/10.1016/j.compedu.2011.02.004

[19] Lena Cohen, Heila Precel, Harold Triedman, and Kathi Fisler. 2021. A New Model for Weaving Responsible Computing Into Courses Across the CS Curriculum. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 858–864.

[20] Mathilde Collain and Deborah Trytten. 2019. You don't have to be a white male that was learning how to program since he was five. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 968–974.

[21] Sally Jo Cunningham, Annika Hinze, and David M. Nichols. 2016. Supporting Gender-Neutral Digital Library Creation: A Case Study Using the GenderMag Toolkit. In *Digital Libraries: Knowledge, Information, and Data in an Open Access Society,* Atsuyuki Morishima, Andreas Rauber, and Chern Li Liew (Eds.). Springer International Publishing, Cham, 45–50.

[22] DO-IT. 2015. *Web design & development I*. http://www.uw.edu/accesscomputing/webd2/

[23] Brianna Dym, Namita Pasupuleti, Cole Rockwood, and Casey Fiesler. 2021. "You don't do your hobby as a job": Stereotypes of Computational Labor and their Implications for CS Education. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 823–829.

[24] Rodrigo Ferreira and Moshe Y Vardi. 2021. Deep Tech Ethics: An Approach to Teaching Social Justice in Computer Science. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 1041–1047.

[25] Casey Fiesler, Mikhaila Friske, Natalie Garrett, Felix Muzny, Jessie J Smith, and Jason Zietz. 2021. Integrating Ethics into Introductory Programming Classes. *In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE'21)*. New York, NY, USA: ACM.

[26] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What do We teach when We teach tech ethics? A syllabi analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 289–295.

[27] Holly Fiock. 2020. Designing a community of inquiry in online courses. *The International Review of Research in Open and Distributed Learning* 21, 1 (2020), 135–153.

[28] Denae Ford, Justin Smith, Philip J Guo, and Chris Parnin. 2016. Paradise unplugged: Identifying barriers for female participation on stack overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 846–857.

[29] D Randy Garrison, Terry Anderson, and Walter Archer. 1999. Critical inquiry in a text-based environment: Computer conferencing in higher education. *The internet and higher education* 2, 2-3 (1999), 87–105.

[30] Emily Gonzalez-Holland, Daphne Whitmer, Larry Moralez, and Mustapha Mouloua. 2017. Examination of the use of Nielsen's 10 usability heuristics & outlooks for the future. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 61. SAGE Publications Sage CA: Los Angeles, CA, 1472–1475.

[31] Elizabeth Goodman, Mike Kuniavsky, and Andrea Moed. 2012. *Observing the User Experience, Second Edition: A Practitioner's Guide to User Research* (2nd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[32] Jerilee Grandy. 1994. Gender and ethnic differences among science and engineering majors: Experiences, achievements, and expectations. *ETS Research Report Series* 1994, 1 (1994), i–63.

[33] Thomas A Harris. 1967. *I'm OK, you're OK*. Random House.

[34] Gillian R Hayes. 2014. Knowing by doing: action research as an approach to HCI. In *Ways of Knowing in HCI*. Springer, 49–68.

[35] C. Hilderbrand, C. Perdriau, L. Letaw, J. Emard, Z. Steine-Hanson, M. Burnett, and A. Sarma. 2020. Engineering Gender-Inclusivity into Software: Ten Teams' Tales from the Trenches. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. 433–444.

[36] Charles G. Hill, Maren Haag, Alannah Oleson, Chris Mendez, Nicola Marsden, Anita Sarma, and Margaret Burnett. 2017. *Gender-Inclusiveness Personas vs. Stereotyping: Can We Have It Both Ways?* Association for Computing Machinery, New York, NY, USA, 6658–6671. https://doi.org/10.1145/3025453.3025609

[37] Hsiu-Fang Hsieh and Sarah E Shannon. 2005. Three approaches to qualitative content analysis. *Qualitative health research* 15, 9 (2005), 1277–1288.

[38] Hui-Ching Kayla Hsu and Nasir Memon. 2021. Crossing the Bridge to STEM: Retaining Women Students in an Online CS Conversion Program. *ACM Transactions on Computing Education (TOCE)* 21, 2 (2021), 1–16.

[39] Margaretha Vreeburg Izzo and William M Bauer. 2015. Universal design for learning: enhancing achievement and employment of STEM students with disabilities. *Universal Access in the Information Society* 14, 1 (2015), 17–27.

[40] René Kizilcec and Andrew Saltarelli. 2019. Psychologically Inclusive Design: Cues Impact Women's Participation in STEM Education. In *Proceedings of the 2019 CHI Conference on human factors in computing systems (CHI '19)*. ACM, 1–10.

[41] Rene Kizilcec, Andrew Saltarelli, Petra Bonfert-Taylor, Michael Goudzwaard, Ella Hamonic, and Rémi Sharrock. 2020. Welcome to the Course: Early Social Cues Influence Women's Persistence in Computer Science. In *Proceedings of the 2020 CHI Conference on human factors in computing systems (CHI '20)*. ACM, 1–13.

[42] Rene F Kizilcec and Anna Kambhampaty. 2020. Identifying course characteristics associated with sociodemographic variation in enrollments across 159 online courses from 20 institutions. *PloS one* 15, 10 (2020), e0239766–e0239766.

[43] Maria Klawe. 2013. Increasing female participation in computing: The Harvey Mudd College story. *Computer* 46, 3 (2013), 56–58.
[44] Amy J Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Commun. ACM* 63, 11 (2020), 31–33.
[45] Sandeep Kaur Kuttal, Kevin Gerstner, and Alexandra Bejarano. 2019. Remote Pair Programming in Online CS Education: Investigating through a Gender Lens. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing* (VL/HCC). IEEE, 75–85.
[46] Kurt Lewin. 1952. Group decision and social change. Readings in social psychology. *Newcombe and Hartley (Eds.), Henry Holt, New York* (1952).
[47] A. Light. 2011. HCI as heterodoxy: Technologies of identity and the queering of interaction with computers. *Interacting with Computers* 23, 5 (2011), 430–438. https://doi.org/10.1016/j.intcom.2011.02.002
[48] Stephanie Ludi, Matt Huenerfauth, Vicki Hanson, Nidhi Rajendra Palan, and Paula Garcia. 2018. Teaching inclusive thinking to undergraduate students in computing programs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 717–722.
[49] Jennifer Mankoff. 2006. Practical service learning issues in HCI. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 201–206.
[50] Jane Margolis and Allan Fisher. 2002. *Unlocking the clubhouse: Women in computing*. MIT press.
[51] Allison Mattheis, Daniel Cruz-Ramírez De Arellano, and Jeremy B Yoder. 2019. A model of queer STEM identity in the workplace. *Journal of homosexuality* (2019).
[52] Ryan A Miller and Megan Downey. 2020. Examining the STEM Climate for Queer Students with Disabilities. *Journal of Postsecondary Education and Disability* 33, 2 (2020), 169–181.
[53] National Center for Women and Information Technology (NCWIT). 2015. *Survey-in-a-Box: Student Experience of the Major*. https://www.ncwit.org/resources/survey-box-student-experience-major-0
[54] Jakob Nielsen. 1994. Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. Academic Press, Inc., USA, 245–272.
[55] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 249–256.
[56] Oleson, Christopher Mendez, Zoe Steine-Hanson, Claudia Hilderbrand, Christopher Perdriau, Margaret Burnett, and Amy J. Ko. 2018. Pedagogical Content Knowledge for Teaching Inclusive Design. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (Espoo, Finland) (ICER '18)*. Association for Computing Machinery, New York, NY, USA, 69–77. https://doi.org/10.1145/3230977.3230998
[57] Alannah Oleson, Meron Solomon, and Amy J Ko. 2020. Computing Students' Learning Difficulties in HCI Education. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
[58] Susmita Hema Padala, Christopher John Mendez, Luiz Felipe Dias, Igor Steinmacher, Zoe Steine Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Dale Simpson, Margaret Burnett, et al. 2020. How gender-biased tools shape newcomer experiences in OSS projects. *IEEE Transactions on Software Engineering* (2020).
[59] Krystle Phirangee and Alesia Malec. 2017. Othering in online learning: An examination of social presence, identity, and sense of community. *Distance Education* 38, 2 (2017), 160–172.

[60] Vahab Pournaghshband and Paola Medel. 2020. Promoting Diversity-Inclusive Computer Science Pedagogies: A Multidimensional Perspective. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. 219–224.
[61] Cynthia Putnam, Maria Dahman, Emma Rose, Jinghui Cheng, and Glenn Bradford. 2016. Best practices for teaching accessibility in university classrooms: cultivating awareness, understanding, and appreciation for diverse users. *ACM Transactions on Accessible Computing (TACCESS)* 8, 4 (2016), 1–26.
[62] Inioluwa Deborah Raji, Morgan Klaus Scheuerman, and Razvan Amironesei. 2021. You Can't Sit With Us: Exclusionary Pedagogy in AI Ethics Education. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 515–525.
[63] Jennifer A Rode and Erika Shehan Poole. 2018. Putting the gender back in digital housekeeping. In *Proceedings of the 4th Conference on Gender & IT*. 79–90.
[64] Arun Shekhar and Nicola Marsden. 2018. Cognitive Walkthrough of a Learning Management System with Gendered Personas. In *Proceedings of the 4th Conference on Gender & IT (Heilbronn, Germany) (GenderIT '18)*. Association for Computing Machinery, New York, NY, USA, 191–198. https://doi.org/10.1145/3196839.3196869
[65] Kristen Shinohara, Cynthia L Bennett, and Jacob O Wobbrock. 2016. How designing for people with and without disabilities shapes student design thinking. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 229–237.
[66] Steven E Stemler. 2004. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research, and Evaluation* 9, 1 (2004), 4.
[67] Jane G Stout and Heather M Wright. 2016. Lesbian, gay, bisexual, transgender, and queer students' sense of belonging in computing: An Intersectional approach. *Computing in Science & Engineering* 18, 3 (2016), 24–30.
[68] Ernest T Stringer. 2007. *Action Research*. Sage.
[69] Simone Stumpf, Anicia Peters, Shaowen Bardzell, Margaret Burnett, Daniela Busse, Jessica Cauchard, and Elizabeth Churchill. 2020. Gender-inclusive HCI research and design: A conceptual review. *Foundations and Trends in Human–Computer Interaction* 13, 1 (2020), 1–69.
[70] Lucy Suchman. 2009. Agencies in technology design: Feminist reconfigurations. In Proceedings of 5th European Symposium on Gender & ICT, Digital Cultures: Participation–Empowerment–Diversity.
[71] Adrian Thinnyun, Ryan Lenfant, Raymond Pettit, and John R Hott. 2021. Gender and Engagement in CS Courses on Piazza. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 438–444.
[72] Jan H Van Driel, Nico Verloop, and Wobbe De Vos. 1998. Developing science teachers' pedagogical content knowledge. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 35, 6 (1998), 673–695.
[73] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511.
[74] Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, and Margaret Burnett. 2019. From Gender Biases to Gender-Inclusive Design: An Empirical Investigation. In *Proceedings of the 2019 CHI Conference on Human Factors in*

Computing Systems (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300283

[75] Annalu Waller, Vicki L Hanson, and David Sloan. 2009. Including accessibility within and beyond undergraduate computing courses. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*. 155–162.

[76] Linda L Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)* 4, 1 (2004), 4–es.

[77] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[78] Kimberly Michelle Ying, Lydia G Pezzullo, Mohona Ahmed, Kassandra Crompton, Jeremiah Blanchard, and Kristy Elizabeth Boyer. 2019. In their own words: Gender differences in student perceptions of pair programming. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1053–1059.

Appendices

Appendix A: GenderMag, Cognitive Styles, and Nielsens Explorations (Activities1,3)

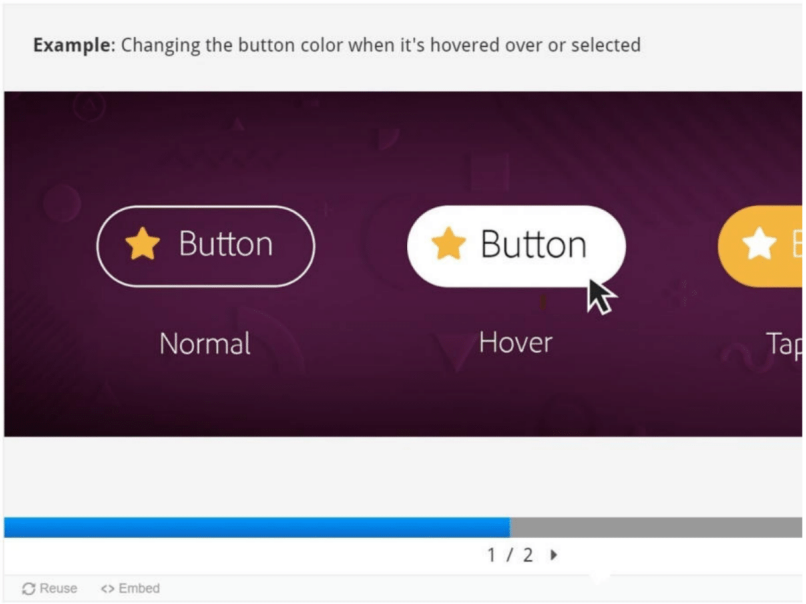
Activity1 CS-DB GenderMag Heuristics Exploration:

Does not include interactive widgets (for displaying images and quiz questions).

Exploration: GenderMag Heuristics

How to Complete This Module

This module will teach you about the 9 GenderMag Heuristics. Included with descriptions of this topic are example photos and quizzes. Note that the photos are presented as multiple slides. You can use the arrows or toolbar at the bottom of the photo widget to navigate between photos (shown below).



Additionally, there is an end quiz with multiple-choice questions for you to complete at the end of the module.

Continue reading below to start learning!

Introduction: What are Software Usability Heuristics?

Software usability heuristics are guidelines for making software user interface (UI) designs more usable. Software designers go through heuristics one-by-one when evaluating the usability of a

UI. Software usability heuristics can also guide the creation of a new UI.

An advantage of software usability heuristics is they are an inexpensive way to make a UI more usable: they do not require testing with actual users.

What are the GenderMag Heuristics?

One set of software usability heuristics is the GenderMag Heuristics. Margaret Burnett, a leader in usability research, published this set of 9 "heuristics for avoiding gender-inclusiveness 'bugs' in software" in 2019.

The GenderMag Heuristics are used to find and fix gender bias bugs in software. They are based on research that has found that the ways people problem-solve in software tends to cluster by gender. However, many software features are inadvertently designed to work best for problem-solving styles favored mainly by men.

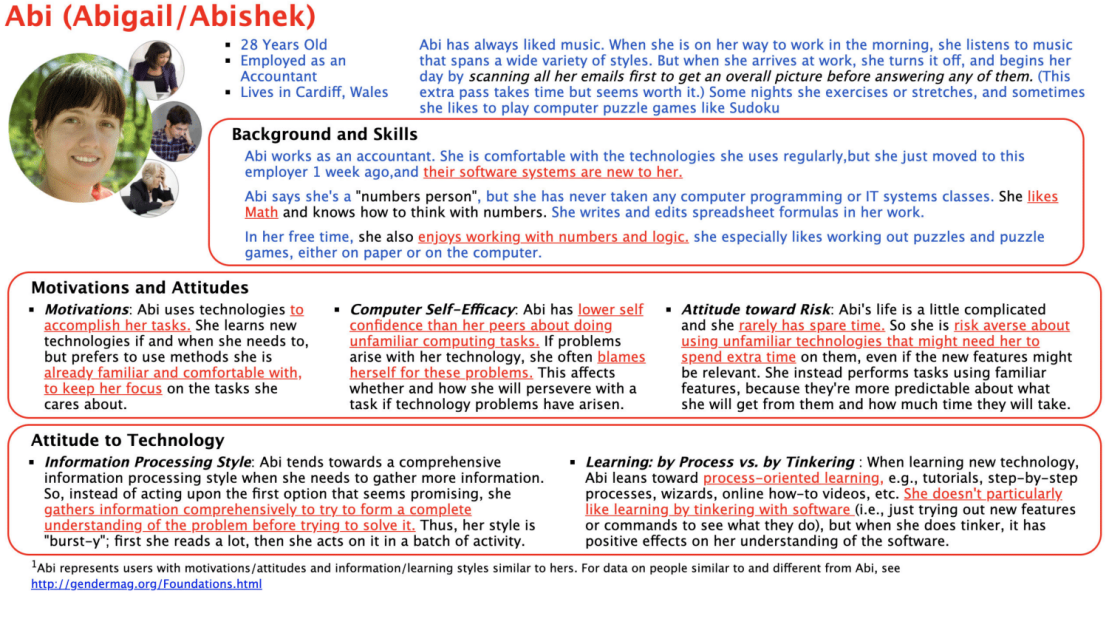
At the core of the GenderMag Heuristics are five *cognitive problem-solving facets*:

- a user's *motivations* for using software
- their *information processing style*
- their *computer self-efficacy*
- their *attitude towards risk*
- their *style of learning* new technology.

The GenderMag Heuristics are guided by the GenderMag *personas*: The Abi persona represents how women tend to problem-solve in software, and the Tim persona represents how men tend to problem-solve in software. Abi and Tim are detailed in the images below.

▶

Note: The blue text is customizable; the blue text shown is only an example. In addition, Abi and Tim may be any gender.



Tim (Timothy/Timara)

- 28 Years Old
- Employed as an Accountant
- Lives in Cardiff, Wales

Tim loves public transportation. He knows several routes to get there from home and he's always exploring ways to optimize his trips into the office. *Work starts with email, Which he answers one at a time, as soon as he reads them.* (Sometimes this backfires, if there is a second related message he hasn't read yet, but he doesn't mind sending a follow-up email.) Some nights he plays computer games with his online friends.

Background and Skills

Tim works as an accountant. He just moved to this employer 1 week ago,and their software systems are new to him. For Tim, technology is a source of fun, and he is always on the lookout for new computer software. He likes to make sure he has the latest version of all the software with all the new features.

Tim says he's a "numbers person" , but he has not taken any computer programming or IT classes. Tim likes Math and knows how to think in terms of numbers. He writes and edits spreadsheet formulas for his work.

He plays the latest video games, has the newest smart phone and hybrid car. He downloads and installs the latest software, and experiments with its settings. He is comfortable and confident with technology and he enjoys learning about it and using new technologies.

Motivations and Attitudes

- ***Motivations***: Tim likes learning all the available functionality on all of his devices and computer systerms he uses, even when it may not be necessary to help him achieve his tasks. he sometimes finds himself exploring functions of one of his gadgets for so long that he loses sight of what he wanted to do with it to begin with.
- ***Computer Self-Efficacy***: Tim has high confidence in his abilities with technology, and thinks he's better than the average person at learning about new features.If he can't fix the problem, he blames it on the software vendor it's not his fault if he can't get it to work.
- ***Attitude toward Risk***: Tim doesn't mind talking risks using features of technology, that haven't been proven to work. When he is presented with challenges because he has tried a new way that doesn't work, it doesn't changes his attitudes toward technology.

Attitude to Technology

- ***Information Processing Style***: Tim leans towards a selective information processing style or "depth first" approach. That is, he usually delves into the first promising option, pursues it, and if it doesn't work out he backs out and gathers a bit more information until he sees another option to try. Thus, his style is very incremental.
- ***Learning: by Process vs. by Tinkering*** : Whenever Tim uses new technology, he tries to construct his own understanding of how the software works internally. He likes tinkering and exploring the menu items and functions of the software in order to build that understanding. Sometimes he plays with features too much, losing focus on what he set out to do originally, but this helps him gain better understanding of the software.

[1]Tim represents users with motivations/attitudes and information/learning styles similar to his. For data on people similar to and different from Tim, see http://gendermag.org/Foundations.html

▶ summarize, the GenderMag Heuristics are a set of principles based on the GenderMag personas and cognitive problem-solving facets. They allow developers to quickly identify parts of a UI that potentially have gender bias bugs. Keep reading below to learn about each of the heuristics.

Heuristic #1 (of 9): Explain what ***new*** features do, and why they are useful

Abi uses software only as needed for their task. They prefer familiar features to keep focused on the task and may be wary of new features.

Tim likes using software to learn what new features can help them accomplish.

Allow Abi to quickly assess new features so they can choose whether to start using them. Allow Tim to quickly assess what a feature is for so they can move on to finding out what the rest of the software's features do.

▶ euristic #2 (of 9): Explain what ***existing*** features do, and why they are useful

Abi is risk-averse and so may avoid using features that have an unknown time cost and an unknown benefit.

Tim is risk-tolerant so may use features without knowing their cost or even what they do. Tim is also motivated to investigate new, cutting-edge features.

Allow Abi to determine whether existing features are familiar to them. Allow Tim to more efficiently determine which features are known and which are new and unique.

Heuristic #3 (of 9): Let people gather as much information as they want, and no more than they want

Abi gathers and reads relevant information comprehensively before acting.

Tim likes to delve into the first option and pursue it, backtracking if need be.

Allow Abi to find the information they want, but don't force them to spend excessive time or effort gathering that information. Allow Tim to find correct information immediately, so that they don't go down the wrong path or get distracted from their task.

Heuristic #4 (of 9): Keep familiar features available

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software, such as features having changed.

Tim has high computer self-efficacy, so often blames the software if a problem arises, such as features having changed.

To encourage Abi and Tim to continue using the software, allow them to interact with it in ways they expect.

Heuristic #5 (of 9): Make undo/redo and backtracking available

Abi is risk-averse, so prefers not to take actions in software that might not be easy to reverse.

Tim is risk-tolerant, so is willing to take actions in software that might be incorrect and need to be reversed.

Provide undo/redo and backtracking to allow Abi to feel comfortable proceeding, and to allow Tim to recover from mistakes.

Heuristic #6 (of 9): Provide ways to try out different approaches

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software. This can discourage Abi from persevering in the software.

Tim has high computer self-efficacy, so often blames the software if a problem arises but is willing to try numerous ways of addressing the problem.

Allow Abi to try a different approach when they feel unable to proceed with the current one. Allow Tim to try multiple ways of solving a problem.

Heuristic #7 (of 9): Communicate the amount of effort that will be required to use a feature

Abi is risk-averse, so may want to avoid features with high effort costs.

Tim is risk-tolerant, so may begin using features that require extra effort and time, and that are unrelated to the task at hand.

Allow Abi to decide whether or not a feature will require too much effort to use. Allow Tim to understand that a feature may take extra effort, and thus more time, to help them stay on track with their task.

Heuristic #8 (of 9): Provide a path through the task

Abi is a process-oriented learner, so prefers to proceed through tasks step-by-step.

Tim learns by tinkering, so prefers not to be constrained by rigid, pre-determined processes.

Provide Abi a way to go through tasks using a clear process. Provide Tim a way to bypass step-by-step processes and tutorials if those are not required for learning the software.

Heuristic #9 (of 9): Encourage mindful tinkering

Abi is a process-oriented learner so usually prefers not to tinker. Because of this, they might miss useful or important parts of the software.

Tim learns by tinkering, but sometimes tinkers addictively and gets distracted from their task.

Encourage Abi to tinker in ways that lead to them discovering task-relevant functionality. Encourage Tim not to over-tinker (e.g., by adding an extra click), so that they make fewer mistakes, have time to absorb important information, and stay on-task.

Wrap Up

The GenderMag Heuristics are a set a 9 software usability heuristics for evaluating and improving the usability of UIs for different genders of users.

Quiz

How might Tim send an email from an unfamiliar site without a undo button?

- ○ They would try out some of the settings and text formatting only to get frustrated with the site when there was no undoing the changes
- ○ They would notice the lack of undo button, be worried about making mistakes, and find a different email service
- ○ They would navigate to the compose email text area, type their message, and send (perhaps with mistakes)

Check ›

References

Zoe Steine-Hanson, Claudia Hilderbrand, Lara Letaw, Jillian Emard, Christopher Perdriau, Christopher Mendez, Margaret Burnett, and Anita Sarma. 2019. *Fixing Inclusivity Bugs for Information Processing Styles and Learning Styles*. arXiv:1905.02813 [cs.HC].

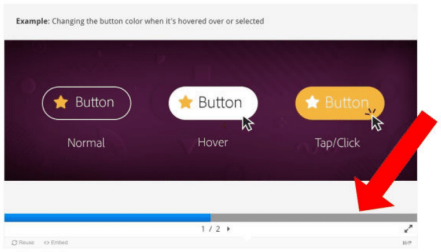
Activity1 CS-DB Cognitive Styles Heuristics Exploration:

Final iteration of the CS-DB GenderMag/Cognitive Style Heuristics Exploration (differing sections only). See Section 4 for details on changes made. Does not include interactive widgets (for displaying images and quiz questions).

Exploration: Cognitive Style Heuristics

How to Complete This Module

This module will teach you about the **9 Cognitive Style Heuristics**. For each heuristic, there's a text description and example images. **Note**: The images are presented as multiple slides. You can **use the arrows or toolbar at the bottom of the image widget to navigate** between images.



Additionally, there's an end quiz with multiple-choice questions for you to complete at the end of the module.

Continue reading below to start learning!

Introduction: What are Software Usability Heuristics?

Software usability heuristics are guidelines for making software user interface (UI) designs more usable. Software designers go through heuristics one-by-one when evaluating the usability of a UI. Software usability heuristics can also guide the creation of a new UI.

An advantage of software usability heuristics is they are an inexpensive way to make a UI more usable: they **do not require testing with actual users**.

What are the Cognitive Style Heuristics?

The Cognitive Style Heuristics are one set of software usability heuristics. They were **created by a research team** headed by Margaret Burnett, a leader in usability research.

The Cognitive Style Heuristics are **used to find and fix usability bugs in software**. They are **based on research about different ways people problem-solve in software** they're using for

the first time.

At the core of the Cognitive Style Heuristics are five **cognitive problem-solving facets**:

- a user's **motivations** for using software (possible **facet values**: task completion vs. interest)
- their **information processing style** (comprehensive vs. selective)
- their **computer self-efficacy** (low vs. high)
- their **attitude toward risk** (risk-averse vs. risk-tolerant)
- their **style of learning** new software (by process vs. by tinkering)

The Cognitive Style Heuristics are guided by **cognitive style personas**: Abi, Pat, and Tim. Abi's cognitive style consists of a set of facet values at one end of the cognitive style spectrum. Tim is at the other end. Pat is somewhere in between. **Most people are like Pat**---a mixture of facet values. In addition, **an individual's facet values can change with time and circumstances**.

Note: The blue text is customizable; the blue text shown is only an example. Abi, Pat, and Tim may be any gender.

Researchers have shown that **designing with cognitive styles in mind can make software less gender-biased** [Vorvoreanu 2019].

To summarize, the Cognitive Style Heuristics are a set of principles based on the cognitive style personas and cognitive problem-solving facets. They allow developers to quickly identify parts of a UI that potentially have usability bugs. Designing for cognitive styles can make software less gender-biased. Keep reading below to learn about each of the heuristics.

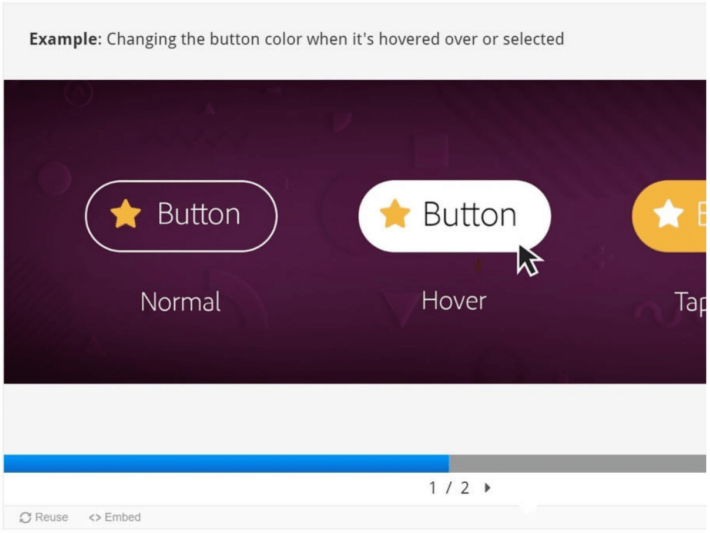
CS-DB Nielsen's Heuristics Exploration:

Nielsen's Heuristics exploration provided to students. Does not include interactive widgets (for displaying images and quiz questions). Version used for all three CS-DB classes.

Exploration: Nielsen's Heuristics

How to Complete This Module

This module will teach you about Nielsen's 10 Usability Heuristics. Included with descriptions of this topic are example photos and quizzes. Note that the photos are presented as multiple slides. You can use the arrows or toolbar at the bottom of the photo widget to navigate between photos (shown below).



Additionally, there is an end quiz with multiple-choice questions for you to complete at the end of the module.

Continue reading below to start learning!

Introduction: What are Software Usability Heuristics?

Software usability heuristics are guidelines for making software user interface (UI) designs more usable. Software designers go through heuristics one-by-one when evaluating the usability of a

UI. Software usability heuristics can also guide the creation of a new UI.

An advantage of software usability heuristics is they are an inexpensive way to make a UI more usable: they do not require testing with actual users.

What are Nielsen's Heuristics?

One set of software usability heuristics is Nielsen's Heuristics. Jakob Nielsen, a leader in usability research, published this set of "10 general principles for interaction design" in 1994. Keep reading, below, to learn about each of the heuristics.

Heuristic #1 (of 10): Visibility of system status

Always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time. Allow users to feel in control by providing communication between the system and the user.

Heuristic #2 (of 10): Match between system and real life

Use words, phrases and concepts familiar to the user, rather than system-oriented terms. Do this by following real-world conventions, making information appear in a natural and logical order.

Heuristic #3 (of 10): User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. To help them, support undo and redo and let users handle accidents or changed minds.

▶ euristic #4 (of 10): Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. According to Jakob's Law, people spend most of their time on sites other than yours. Thus, maintaining consistency both within a UI (*internal* consistency) and across UIs in general (*external* consistency) increases learnability of the UI as there are fewer new things for users to learn.

▶ euristic #5 (of 10): Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. This can be critical for software where errors may have significant impact on critical parts of people's lives (e.g., medical, financial, etc.).

▶ euristic #6 (of 10): Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. Users should not have to remember information from one part of the UI to another. Provide users instructions for using the system that are visible or easily retrievable.

Example of using recognition: Is the [redacted]
Example of using Recall: What is the [redacted]

▶ euristic #7 (of 10): Flexibility and efficiency of use

Accelerators, unseen/ignorable by the novice user, may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

▸ euristic #8 (of 10): Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. "Communicate, don't decorate."

▸ euristic #9 (of 10): Help users recognize, diagnose, and recover from errors

Express error messages in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. You can even provide an immediate course of action for solving the problem when the error is shown.

Example: "Error 404" vs. "The page you're looking for wasn't found"
Example: "Page not found, click 'here' to return to the previous page"

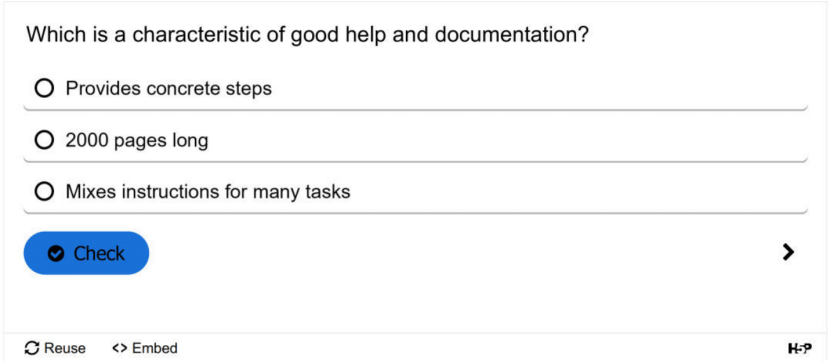
▸ euristic #10 (of 10): Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. To do this, give straightforward, scannable steps for the user and use screenshots when possible.

▸ rap Up

Nielsen's Heuristics are a set of 10 software usability heuristics for evaluating and improving general UI usability.

Quiz



References

Jakob Nielsen. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

Return to Extra Credit

Activity3 CS-SE Cognitive Styles Heuristics Exploration:

Converted from HTML. Examples and quizzes converted to text/image, but in the course were interactive H5P widgets: https://h5p.org/

Exploration: Cognitive Style Heuristics

Image source: Solution Tree

Overview

Your team will use the Cognitive Style Heuristics to make your software more usable.

In this exploration you will:

- Learn about each of the nine Cognitive Style Heuristics and their relevance to software design
- Learn about Cognitive Style Personas and their cognitive styles
- Learn the background and research behind Cognitive Style Heuristics and Personas
- Have the opportunity to self-assess your understanding of the material
- Learn about the cognitive styles of your instructional team

If you are uncomfortable with this material, contact me for a substitute reading.

What are the Cognitive Style Heuristics?

The Cognitive Style Heuristics are nine principles of interaction design framed around how different people use software in different ways. They were **created by a research team** headed by Margaret Burnett, a leader in usability research.

The Cognitive Style Heuristics are **used to find and fix usability bugs in software**. They are **based on research about different ways people problem-solve in software** they're using for the first time.

At the core of the Cognitive Style Heuristics are five **cognitive problem-solving facets**:

- a user's **motivations** for using software (possible **facet values**: task completion vs. interest)
- their **information processing style** (comprehensive vs. selective)
- their **computer self-efficacy** (low vs. high)
- their **attitude toward risk** (risk-averse vs. risk-tolerant)
- their **style of learning** new software (by process vs. by tinkering)

The Cognitive Style Heuristics are guided by **Cognitive Style Personas**.

Cognitive Style Personas

The Cognitive Style Personas are Abi, Pat, and Tim. Abi's cognitive style consists of a set of facet values at one end of the cognitive style spectrum. Tim is at the other end. Pat is somewhere in between. **Most people are a mixture of Abi and Tim**---a mixture of facet values. In addition, **an individual's facet values can change with time and circumstances**.

Note: Abi, Pat, and Tim may be any gender.

Background

The Cognitive Style Heuristics are derived from the GenderMag Method. The GenderMag Method is a process for finding and fixing gender-inclusivity bugs in software. It uses the Abi, Pat, and Tim personas and their cognitive styles.

What do cognitive styles have to do with gender? Software tends to be biased against the cognitive styles often favored by women. **Designing with cognitive styles in mind can make software less gender-biased** [Vorvoreanu 2019].

In addition, "designing software so that it works for diverse populations matters to software companies' **profitability**, to **equity** in the workplace and at home, and to **anyone in a situation that changes the way they think**, such as when under deadline pressure." [Mendez et al., 2019]

Keep reading below to learn about each of the Cognitive Style Heuristics.

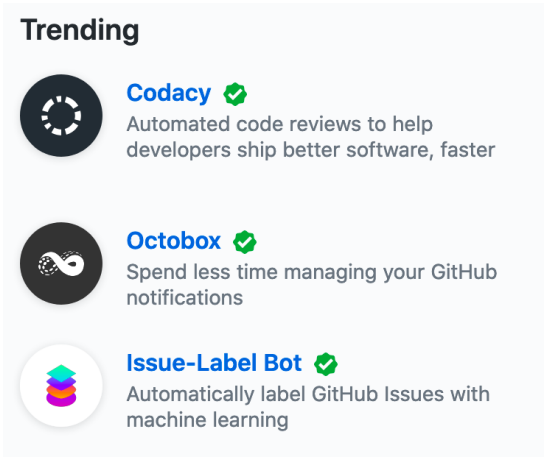
Heuristic #1 (of 9): Explain what *new* features do, and why they are useful

Allow Abi and Pat to quickly assess new features so they can choose whether to start using them. Allow Tim to quickly assess what a feature is for so they can move on to finding out what the rest of the software's features do.

Abi and Pat use software only as needed for their task. They prefer familiar features to keep focused on the task and may be wary of new features.

Tim likes using software to learn what new features can help them accomplish.

Example: Each trending extension has a brief description that says what the extension does and why somebody would use it



Example: This Keybase announcement briefly describes a new feature and how to use it

Heuristic #2 (of 9): Explain what *existing* features do, and why they are useful

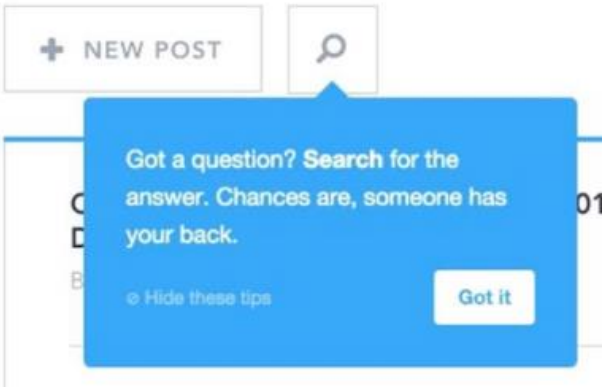
Allow Abi to determine whether existing features are familiar to them. Allow Tim and Pat to more efficiently determine which features are known and which are new and unique.

Abi is risk-averse and so may avoid using features that have an unknown time cost and an unknown benefit.

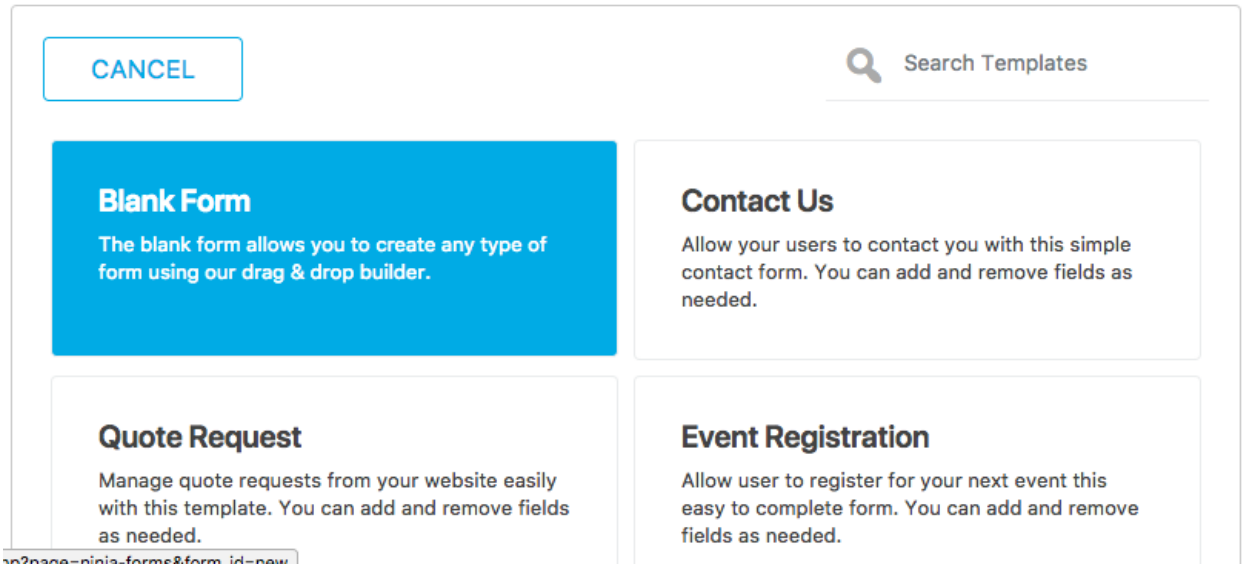
Pat is also risk-averse, but might tinker with features to determine whether they are relevant to accomplishing their task.

Tim is risk-tolerant so may use features without knowing their cost or even what they do. Tim is also motivated to investigate new, cutting-edge features.

Example: The tooltip explains what the search is for and why someone might use it



Example: Each tile explains a feature and the benefit of using the feature



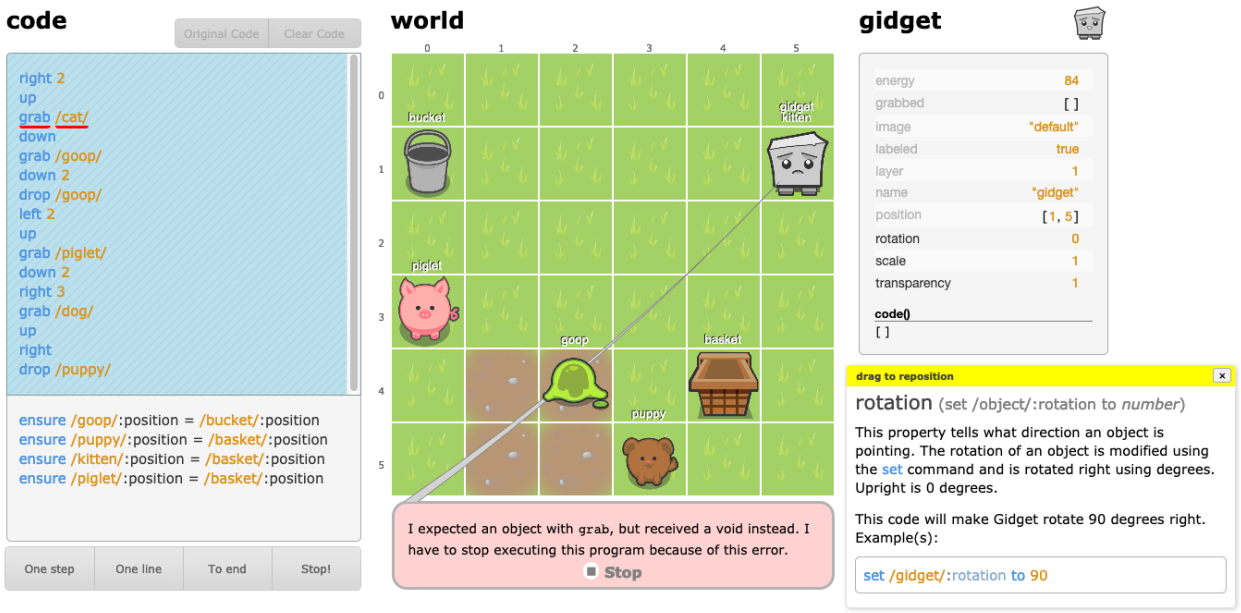
Heuristic #3 (of 9): Let people gather as much information as they want, and no more than they want

Allow Abi and Pat to find the information they want, but don't force them to spend excessive time or effort gathering that information. Allow Tim to find correct information immediately, so that they don't go down the wrong path or get distracted from their task.

Abi and Pat gather and read relevant information comprehensively before acting.

Tim likes to delve into the first option and pursue it, backtracking if need be.

Example: Users can choose to view code documentation while still viewing their code, and the code's output



Example: Users can quickly see the contents of the webpage and jump to the section they're interested in

Printing

- Removing a printer on Windows
- Adding Printers on Windows
- Print Server (for university owned, on domain computers)
- Hostname (personal computers and printers that are not on the print server)
- Adding a printer MacOS
- Ricoh Code - Windows
- Ricoh Code - MacOS

Quiz

Question 1

Which of the following is a good way to help users assess new features and their usefulness?

A. Forcing them to use each feature.
B. Briefly describing each new feature and why it's useful. [correct answer]
C. Don't tell users about the new features; They'll find them on their own.

Question 2

Why might Abi want to see what existing features do, and why they are useful?

A. **Abi likes to know what's going on before investing effort. [correct answer]**
B. **So they can avoid unfamiliar features that might make the task take longer. [correct answer]**
C. Abi always likes exploring without considering time or utility risks.

Question 3

Which are potential disadvantages of showing Tim seven options at once?

A. **Tim might select the first promising option without reading the rest. [correct answer]**
B. **If the options are cool-looking features, Tim might get distracted. [correct answer]**
C. No disadvantages; Tim will read through everything before deciding.

Heuristic #4 (of 9): Keep familiar features available

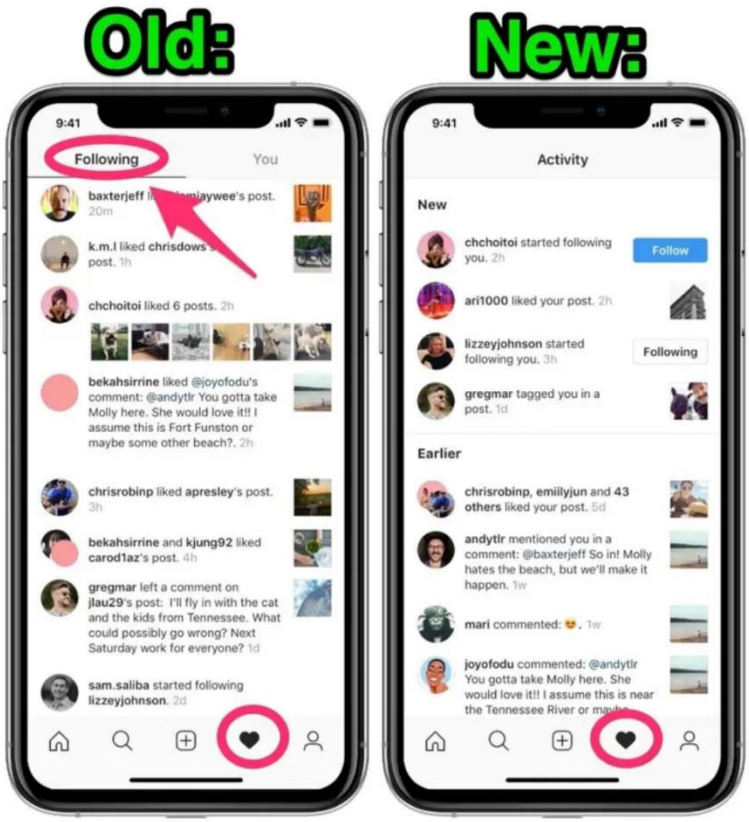
To encourage Abi, Pat, and Tim to continue using the software, allow them to interact with it in ways they expect.

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software, such as features having changed.

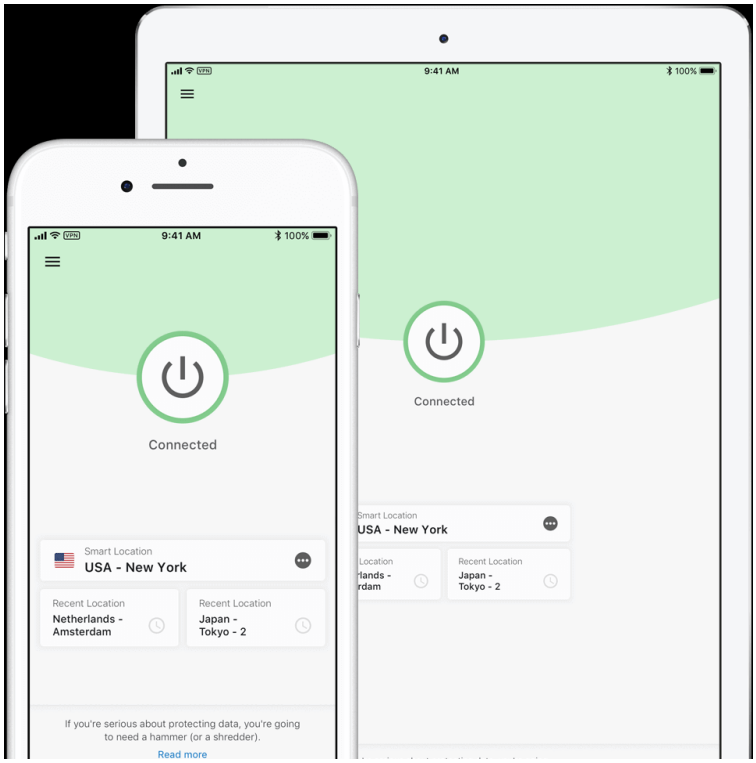
Pat has medium self-efficacy with technology, and will keep on trying for a while if a problem arises, such as features having changed.

Tim has high computer self-efficacy, so often blames the software if a problem arises, such as features having changed.

Example: Although the "following" page is gone, the new Instagram update looks similar to the previous version so that users are still familiar with the app



Example: The iPhone and iPad versions of this VPN app offer the same features which makes switching between the two easy



Heuristic #5 (of 9): Make undo/redo and backtracking available

Provide undo/redo and backtracking to allow Abi and Pat to feel comfortable proceeding, and to allow Tim to recover from mistakes.

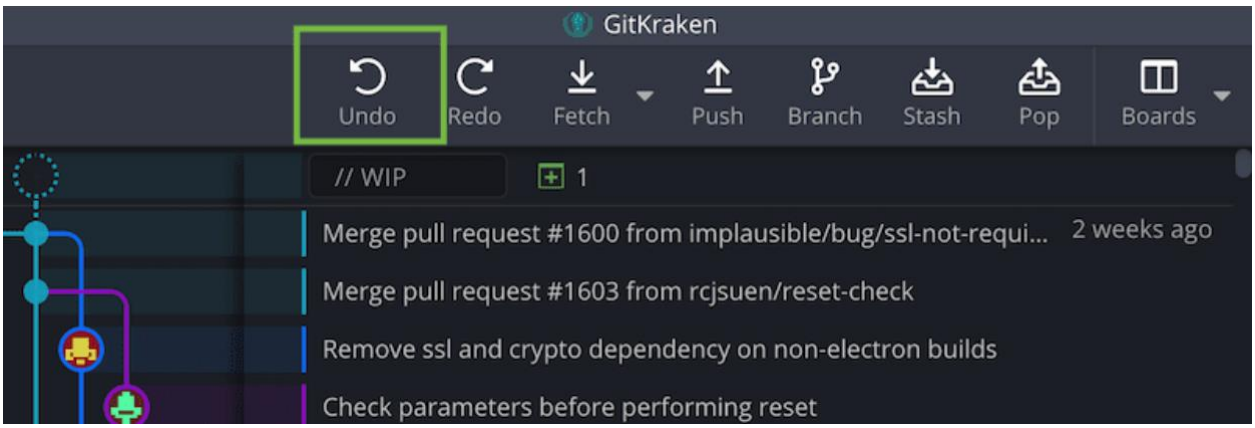
Abi and Pat are risk-averse, so they prefer not to take actions in software that might not be easy to reverse.

Tim is risk-tolerant, so is willing to take actions in software that might be incorrect and need to be reversed.

Example: Browser back/forward buttons allow users to backtrack through their browsing history

[omitted for anonymization]

Example: An undo button allow users to make and recover from mistakes. Also, version control control systems allow users to revert to any previously-committed code state.



Heuristic #6 (of 9): Provide ways to try out different approaches

Allow Abi to try a different approach when they feel unable to proceed with the current one. Allow Tim and Pat to try multiple ways of solving a problem.

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software. This can discourage Abi from persevering in the software.

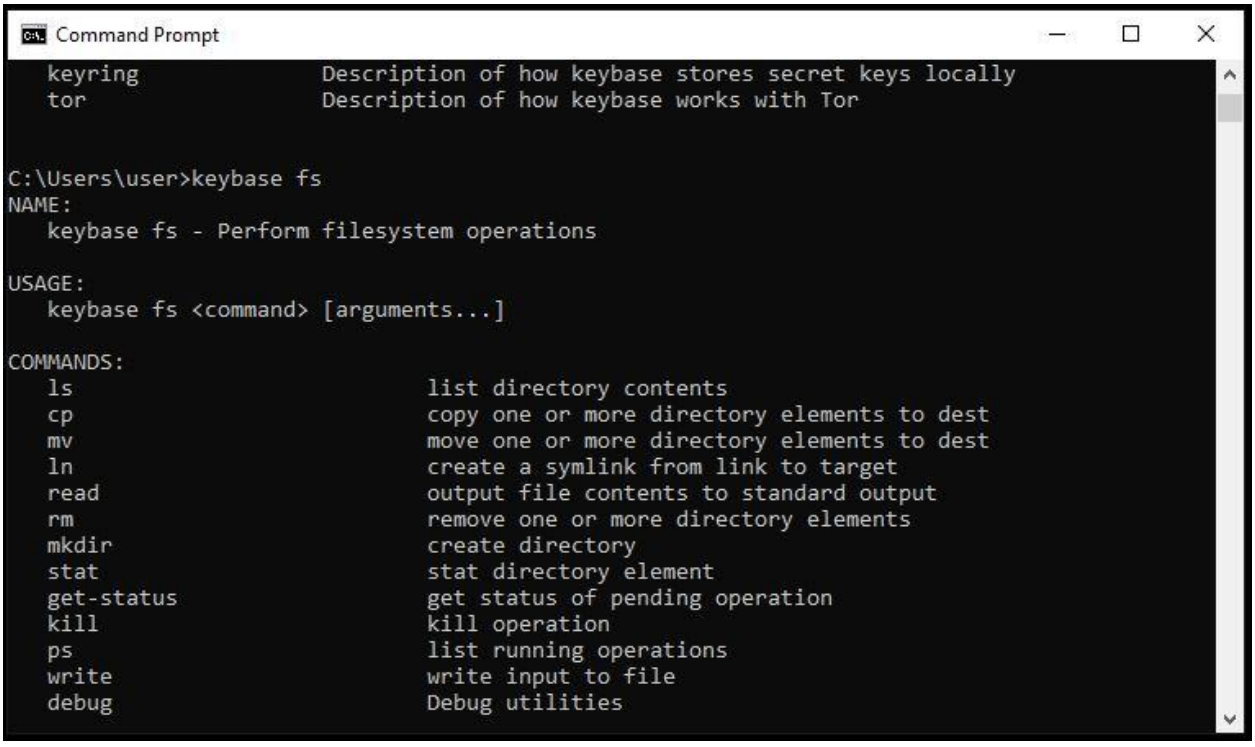
Pat has medium self-efficacy with technology, and will keep on mindfully tinkering for a while if a problem arises, to find the solution.

Tim has high computer self-efficacy, so often blames the software if a problem arises but is willing to try numerous ways of addressing the problem.

Example: If users don't find what they need on the "Choose a Question" drop-down menu, they have multiple other ways of finding the information they need, including through a chatbot

[omitted for anonymization]

Example: If users encounter a problem using the Keybase UI, they can attempt the same operations using the command line interface



Quiz

Question 1

How might a user with low computer self-efficacy benefit from having familiar features available after a software update?

A. No benefit; It deters them from tinkering with new features.
B. They might feel reassured they know what they're doing. [correct answer]
C. They will become bored with the software because it's not changing.

Question 2

Only risk-averse users (Abi and Pat) can benefit from having undo/redo and backtracking options.

A. True
B. False [correct answer]

Question 3

How might providing different ways to accomplish a task affect Abi?

A. Abi always likes to explore and tinker; They will feel encouraged.
B. Abi might appreciate having an alternate approach if another isn't working out. [correct answer]

C. Abi will always feel overwhelmed by this.

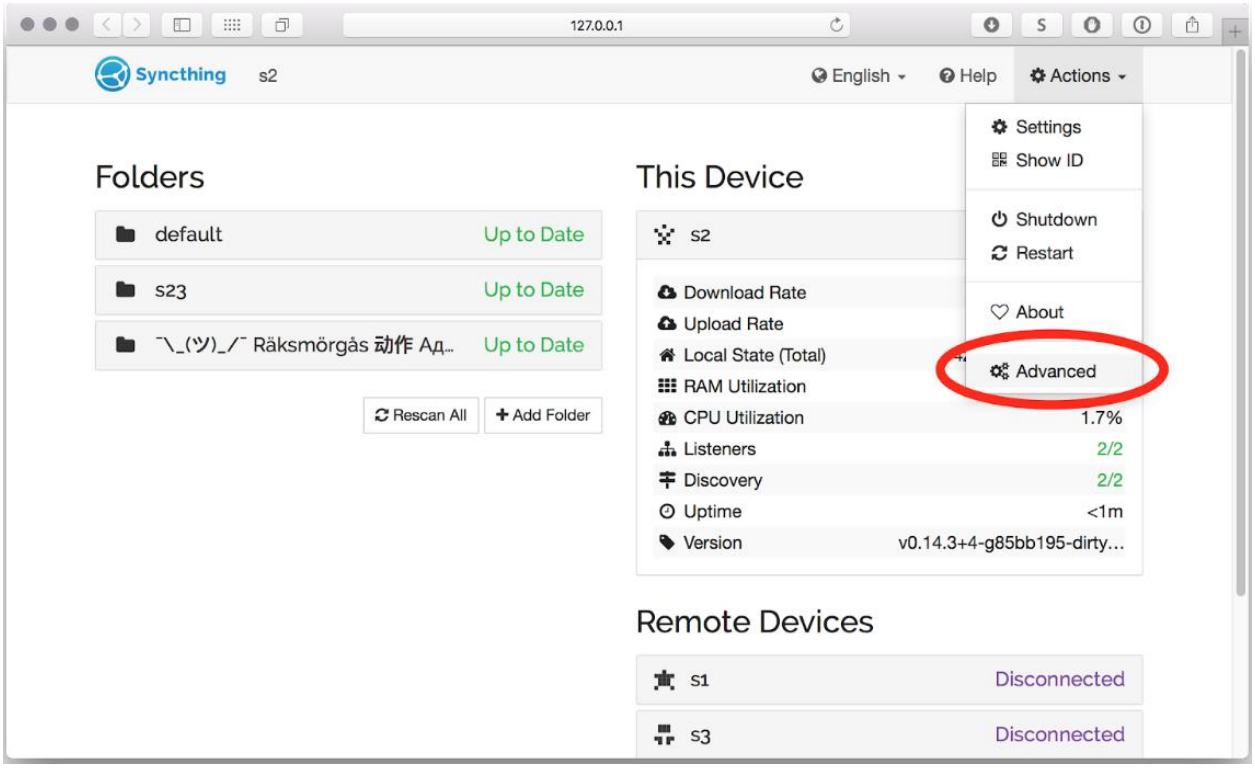
Heuristic #7 (of 9): Communicate the amount of effort that will be required to use a feature

Allow Abi and Pat to decide whether or not a feature will require too much effort to use. Allow Tim to understand that a feature may take extra effort, and thus more time, to help them stay on track with their task.

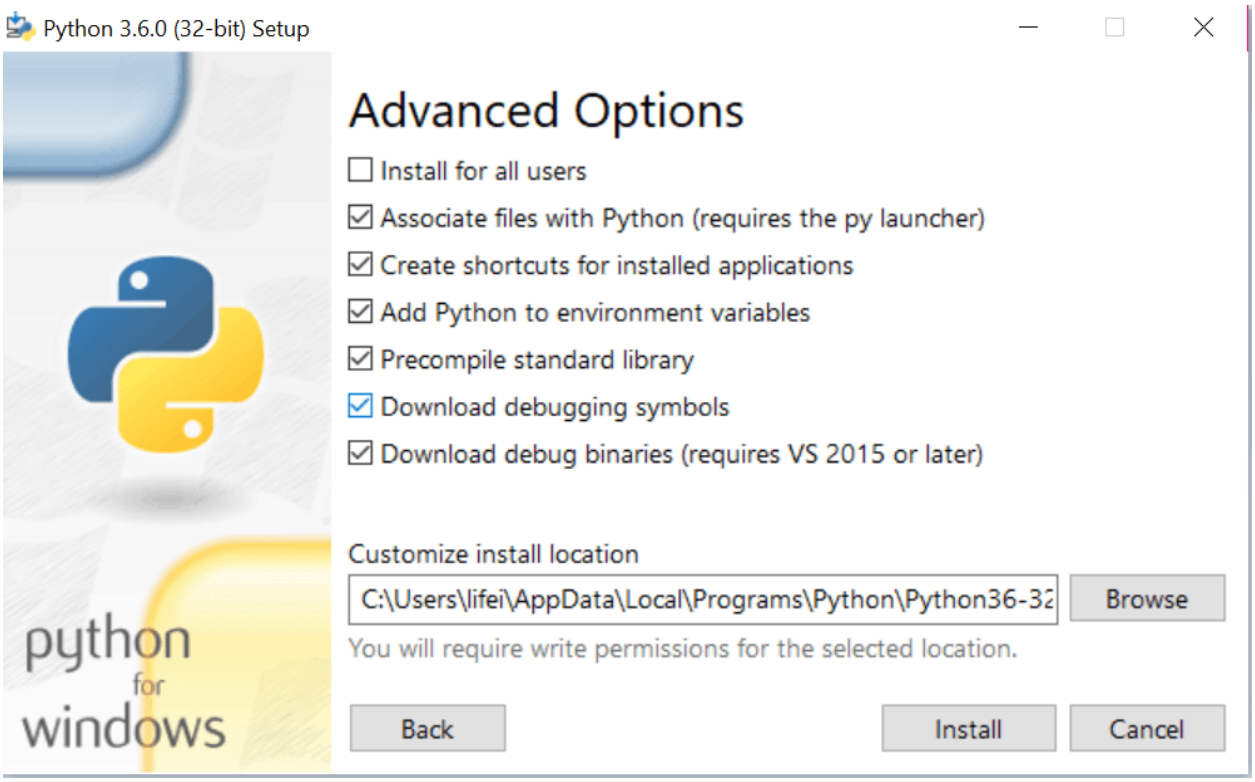
Abi and Pat are risk-averse, so they may want to avoid features with high effort costs.

Tim is risk-tolerant, so may begin using features that require extra effort and time, and that are unrelated to the task at hand.

Example: Placing the "advanced" option far down in a menu helps indicate that "advanced" features may take more effort



Example: The dialog indicates that "py launcher" will be needed to "associate files with Python"



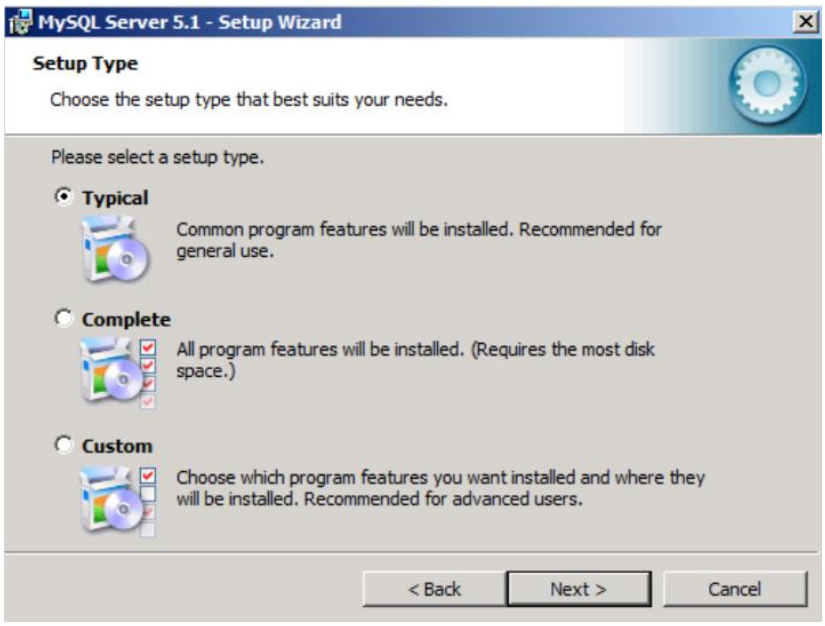
Heuristic #8 (of 9): Provide a path through the task

Provide Abi a way to go through tasks using a clear process. Provide Tim and Pat a way to bypass step-by-step processes and tutorials if those are not required for learning the software.

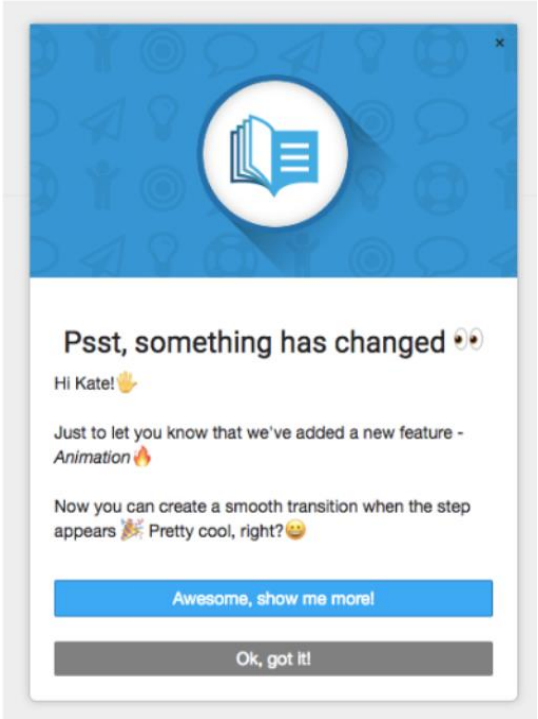
Abi is a process-oriented learner, so prefers to proceed through tasks step-by-step.

Tim and Pat learn by tinkering, and therefore prefer not to be constrained by rigid, pre-determined processes.

Example: Users are given an entry point to their path, with each path explained



Example: Users are given a path to either learning more or dismissing the dialog



Heuristic #9 (of 9): Encourage mindful tinkering

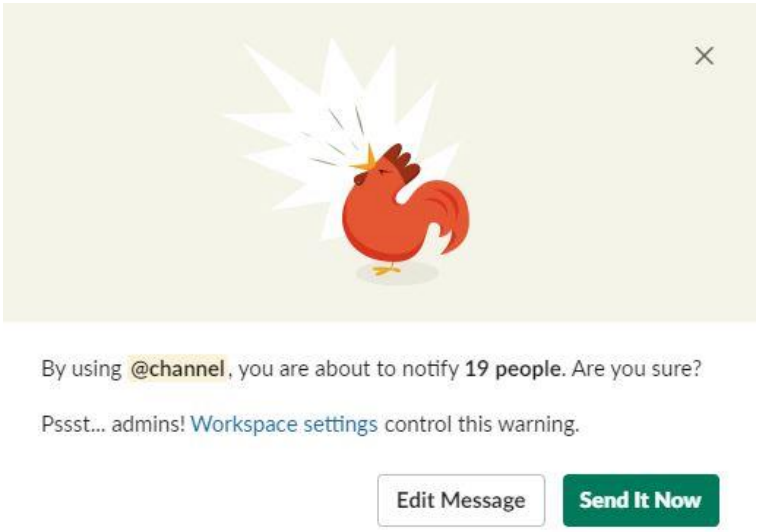
Encourage Abi to tinker in ways that lead to them discovering task-relevant functionality. Encourage Tim not to over-tinker (e.g., by adding an extra click), so that they make fewer mistakes, have time to absorb important information, and stay on-task.

Abi is a process-oriented learner so usually prefers not to tinker. Because of this, they might miss useful or important parts of the software.

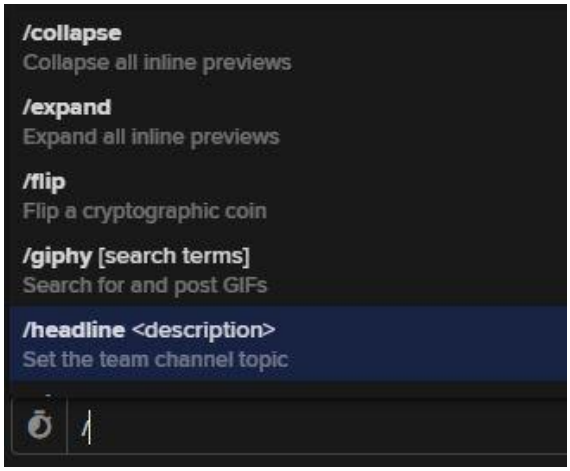
Pat learns by trying out new features but does so mindfully, reflecting on each step.

Tim learns by tinkering, but sometimes tinkers addictively and gets distracted from their task.

Example: Slack encourages mindful tinkering by demonstrating they will notify users when impactful actions are about to take place, such as sending an email to 19 people



Example: Keybase encourages users to try out new "slash" commands by showing all the commands when a user types "/", and explaining what each does and how to use it



Quiz

Question 1

If Tim is told a task will require high effort, how might that affect them?

A. Tim might re-consider addictively tinkering with the task and save time.
B. Tim always likes high-effort tasks; They will feel motivated.
C. Tim will always discontinue the task if it seems high-effort; Tim is risk-averse. [correct answer]

Question 2

Why provide a path through a task?

A. There's no reason to.
B. To help process-orientated learners proceed step-by-step. [correct anwers]
C. To help the user stay focused.

Question 3

Which of the following are benefits of helping Tim slow down or reduce their tinkering?

A. Helps Tim have time to absorb important information [correct answer]
B. Helps Tim stay on-task [correct answer]
C. Helps Tim make fewer mistakes [correct answer]

Cognitive Styles of Your Instructional Team

I and your teaching assistants identified some of our own facet values, below.

Your Instructor <Instructor name>

<Description of which persona(s) instructor identifies with>

First Facet: <Description of facet value identification for any facet>
Second Facet: <Description of facet value identification for any other facet>

Your TA <TA name>

<Description of which persona(s) the TA identifies with>

First Facet: <Description of facet value identification for any facet the TA chooses>

Second Facet: <Description of facet value identification for a second facet the TA chooses>

Wrap Up

The Cognitive Style Heuristics are a set a nine software usability heuristics for evaluating and improving the usability of UIs across users with different cognitive styles. If you would like to learn more, please visit: https://gendermag.org/.

References

Margaret Burnett, Anita Sarma, Claudia Hilderbrand, Zoe Steine-Hanson, Christopher Mendez, Christopher Perdriau. July 2018. *The GenderMag Heuristics (Beta Version)*. https://gendermag.org/flyers_handouts.php.

Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, William Jernigan. 2016. *GenderMag: A Method for Evaluating Software's Gender Inclusiveness*. Interacting with Computers, Volume 28, Number 6, October 2016, pp. 760-787

Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, Margaret Burnett. 2019. *From Gender Biases to Gender-Inclusive Design: An Empirical Investigation*. CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4-9, 2019, Glasgow, Scotland, UK. ACM, New York, NY, USA.

Appendix B: Additional Curricular Materials (Activity2, Activities4-11)

Activity2 CS-DB Extra Credit Assignment:

Extra Credit: Software Usability Heuristics

Complete **ONE** of the following software usability explorations:

1. **Exploration: Nielsen's Heuristics** (general guidelines for making software usable)
2. **Exploration: Cognitive Style Heuristics** (guidelines for making software usable and inclusive)

Turn in a discussion of the exploration you chose. Please follow the requirements below.

Requirements:

- **Reflect** upon what you learned (e.g., what were your thoughts while going through the exploration? what is the exploration about?)
- Give **your own example** of how you might apply one or more of the software usability heuristics.
- Write **at least 400 words**.
- Mention **which exploration** you chose.

Activity4 CS-SE HW1 Facet Reflection:

Cognitive Styles Reflection (30pts)

If you are uncomfortable with this portion of the assignment, contact me for an alternative.

Instructions

- Identify your own facet values and reflect on the Cognitive Style Heuristics exploration. This can help you better understand how to apply the heuristics.

What are **your facet values** when using software? One or more sentences each.

Facet	**Your facet value** *Ex: I prefer to tinker with most software and usually skip tutorials.*
Motivations	*MotivationsFacetValue*
Attitude Toward Risk	*AttitudeTowardRiskFacetValue*
Computer Self-Efficacy	*ComputerSelfEfficacyFacetValue*
Information Processing Style	*InformationProcessingStyleFacetValue*
Learning Style	*LearningStyleFacetValue*

How are you **like Abi**? Two or more sentences. Be specific.

HowYoureLikeAbi

How are you **like Tim**? Two or more sentences. Be specific.

HowYoureLikeTim

What's **one situation when your facet values might change**? Two or more sentences. Be specific.

SituationWhenFacetValuesChange

How did identifying your facet values affect your understanding of how you use software? Two or more sentences. Be specific.

HowIdentificationAffectedUnderstanding

Activity5 CS-SE HW1 Design/Evaluate/Revise:

Paper Prototypes First Draft (70pts)

Instructions

- Create a first draft of your paper prototypes for the feature you chose.
- Low or medium fidelity.
- Show how the feature will look in all states. You may need to create multiple drawings.
- Indicate how the feature moves between states.
- Provide screenshots or scans of your draft.

Where is your **paper prototyping task on Asana**?

AsanaURL

First draft of paper prototypes:

ImagesOfPaperPrototypes

Paper Prototypes Usability Evaluation (15pts)

Instructions

- Evaluate your paper prototypes based on Heuristics #2, #3, and #4. Two or more sentences for each heuristic. Be specific.

Heuristic	How your user interface design does or does not reflect the heuristic
Heuristic #2: Explain what existing features do, and why they are useful	*Evaluation*
Heuristic #3: Let people gather as much information as they want, and no more than they want	*Evaluation*
Heuristic #4: Keep familiar features available	*Evaluation*

Paper Prototypes Second Draft (15pts)

Instructions

- Revise your paper prototypes so they reflect Heuristic #2, #3, and #4.
- Low or medium fidelity.
- Explain what you did. One or more sentences each. Be specific.
- Provide screenshots or scans of your revised paper prototypes. Clearly indicate what has changed.

Heuristic	Change you made based on evaluation
Heuristic #2: Explain what existing features do, and why they are useful	*DescriptionOfChange*
Heuristic #3: Let people gather as much information as they want, and no more than they want	*DescriptionOfChange*
Heuristic #4: Keep familiar features available	*DescriptionOfChange*

Revised paper prototypes:

ImagesOfPaperPrototypes

Activity6 CS-SE Learning Quiz (Question Bank):

*Note: Correct answered are **bolded***

1. Applying the Cognitive Style Heuristics will ONLY make software more usable to women. True or false?
    a. **True**
    b. False
2. Which are the cognitive problem-solving facets? Select ALL correct answers.
    a. **Motivations (task completion vs. interest)**
    b. Gender (man vs. woman vs. non-binary)
    c. **Information Processing Style (comprehensive vs. selective)**
    d. **Style of Learning (by process vs. by tinkering)**
    e. Age (young vs. old)
    f. **Computer Self-Efficacy (low vs. high)**
    g. **Attitude Toward Risk (risk-averse vs. risk-tolerant)**
3. Al Abi's are women and all Tim's are men. True or false?
    a. True
    b. **False**
4. A person's facet values can change with time and circumstances. True or false?
    a. **True**
    b. False
5. Cognitive Style Heuristics are… Select the MOST CORRECT answer.
    a. Not based on research, but still useful.
    b. **Based on research, and proven to be useful.**
    c. Based on research, but results show they aren't very useful.
    d. Not based on research, and not useful.
6. Which of these on-screen messages would best help a user understand a new feature? Select the BEST answer.
    a. **"We added a way to set multiple alarms so you don't have to keep re-created them" <short animation of where to find the feature>**
    b. "New features are now included!"
    c. "We've updated our settings page go take a look to see what's new!"
    d. "Now you can add alarms in multiple ways!"
7. How is Tim most likely to respond when their Wi-Fi stops working? Select the MOST LIKELY answer.
    a. Call tech support (potentially having to wait a long time to get help)
    b. **Turn the Wi-Fi on and off again. If that doesn't work, reset the router (potentially losing settings/configurations)**
    c. Become frustrated and stop working immediately
8. Why is it important to provide options for process-oriented learners? Select the MOST CORRECT answer.
    a. So that they will feel comfortable with the software
    b. To support their progress
    c. So that they can choose the path they want
    d. **All of these options are correct**

9. How might Tim send an email from an unfamiliar site without an undo button? Select the MOST LIKELY answer.
    a. They might notice the lack of undo button, be worried about making mistakes, and find a different email service
    b. They might navigate to the compose email text area, type their message, and send (perhaps with mistakes)
    c. They might try out some of the settings and text formatting only to get frustrated with the site when there was no undoing the changes
10. Which is an example of a good way to explain to users why they would use a feature? Select the BEST answer.
    a. "Tap here!"
    b. "Click here to learn more"
    c. "Clicking this button will create a new post"
11. How is Abi most likely to respond when they update their new alarm clock app and find it's in military time instead of 12-hour time?
    a. Delete the app and download one with 12 hour time
    b. Pull up Google to figure out what time their alarm will be in military time
    c. Set and alarm for the time they think their alarm should be set for
12. Some users prefer jumping to appropriate sections to only read what is relevant to them. True of false?
    a. True
    b. False
13. Which approach is Tim most likely to use when learning new software? Select the MOST LIKELY answer.
    a. Click around the software to figure out how it works
    b. Search YouTube for an instructional tutorial
    c. Call a friend who has used the software before
14. If Abi is unsure of the functionality of a new feature on a familiar platform, what are they most likely to do? Select the MOST LIKELY answer.
    a. Stick to using features they're familiar with
    b. Be interested in learning the feature and try to use it
    c. Tinker with the feature to see if it's relevant
15. Why might Pat want to see what existing feature do and why they are useful? Select ALL correct answers.
    a. So they can avoid unfamiliar features that might make the task take longer.
    b. Pat always likes exploring without considering time or utility risks.
    c. Pat likes to know what's going on before investing effort.
16. Which are potential advantages of showing task relevance? Select ALL correct answers.
    a. Helps Tim stay on-task
    b. Helps Abi decide which features to use
    c. Helps Pat decide which features to use
17. Which of the following is a good way to help users assess new features and their usefulness? Select the BEST answer.
    a. Forcing them to use each feature.
    b. Briefly describing each new feature and why it's useful.
    c. Don't tell users about the new features; they'll find them on their own.

18. Why might Abi want to see what existing features do and why they are useful? Select ALL correct answers.
    a. So they can avoid unfamiliar features that might make the task take longer.
    b. Abi always likes exploring without considering time or utility risks.
    c. Abi likes to know what's going on before investing effort.
19. What are potential disadvantages of showing Tim seven options at once? Select ALL correct answers.
    a. Time might select the first promising option without reading the rest.
    b. If the options are cool-looking features, Tim might get distracted.
    c. No disadvantages; Tim will read through everything before deciding.
20. If a feature that Abi uses often is changed or removed, they will most likely: Select the MOST LIKELY answer.
    a. Try to figure out how to use the new version despite any problems
    b. Blame the software for any problems with the new version
    c. Blame themself for any problems with the new version
21. Undo/redo and backtracking options help Tim because: Select the MOST CORRECT answer.
    a. Knowing that actions are reversible helps them be comfortable proceeding
    b. They don't help Tim
    c. Reversible actions help them backtrack and recover from mistakes
22. Which personas benefit from having undo/redo and backtracking options? Select the BEST answer.
    a. Only Abi and Pat
    b. Only Abi
    c. Only Abi and Tim
    d. Abi, Pat, and Tim
23. How might providing different ways to accomplish a task affect users with high computer self-efficacy? Select the BEST answer.
    a. They will always feel overwhelmed by this.
    b. They might appreciate having an alternate approach with another isn't working out.
    c. They might be able to solve problems in multiple ways.
24. How might Abi benefit from having familiar features available after a software update? Select the BEST answer.
    a. They might feel reassured they know what they're doing.
    b. No benefit; It deters them from tinkering with new features.
    c. They will become bored with the software because it's not changing.
25. How might a user with low computer self-efficacy benefit from having familiar features available after a software update? Select the BEST answer.
    a. They might feel reassured they know what they're doing.
    b. No benefit; It deters them from tinkering with new features.
    c. They will become bored with the software because it's not changing.
26. Only risk-averse users (Abi and Pat) can benefit from having undo/redo and backtracking options. True or False?
    a. True
    b. False

27. How might providing different ways to accomplish a task affect Abi? Select the BEST answer.
    a. Abi will always feel overwhelmed by this.
    b. Abi might appreciate having an alternate approach if another isn't working out.
    c. Abi always likes to explore and tinker; they will feel encouraged.
28. Which of the following is true about communicating the effort needed to use a feature? Select the MOST CORRECT answer.
    a. Tim does not benefit from it because they are risk-tolerant
    b. It can help Abi decide whether or not to use the feature
    c. Pat is risk-tolerant so it will help them stay on track with their task
29. A good example of encouraging mindful tinkering is: Select the BEST answer.
    a. Adding important hidden features that are only found by constant tinkering
    b. Adding information that encourages Abi to tinker with relevant features
    c. Removing barriers to allow Tim to tinker regardless of the relevance of the feature
    d. Removing opportunities to tinker with the software
30. If Abi is told a task will require high effort, how might that affect them? Select the BEST answer.
    a. Abi always likes high-effort tasks; they will feel motivated.
    b. Abi may discontinue the task if it seems high-effort; they are risk averse.
    c. Abi might re-consider addictively tinkering with the task and save time.
31. Process-oriented learners are likely to appreciate: Select the MOST LIKELY answer.
    a. A provided path through the task
    b. A way to bypass step-by-step processes
    c. They don't appreciate anything
32. Which of the following are benefits of encouraging Abi to tinker? Select ALL correct answers.
    a. Helps Abi stay on-task
    b. Helps Abi have time to absorb important information
    c. Helps Abi to not miss important and relevant functionality
33. If Tim is told a task will require high effort, how might that affect them? Select the MOST LIKELY answer.
    a. Tim always likely high-effort tasks; they will feel motivated
    b. Tim will always discontinue the task if it seems high-effort; Tim is risk-averse
    c. Tim might re-consider addictively tinkering with the task and save time
34. Why provide a path through a task? Select the MOST CORRECT answer.
    a. To help the user stay focused
    b. To help process-oriented learners proceed step-by-step
    c. There's no reason to.
35. Which of the following are benefits of helping Tim slow down or reduce their tinkering? Select ALL correct answers.
    a. Helps Tim make fewer mistakes
    b. Helps Tim have time to absorb important information
    c. Helps Tim stay on-task

Activity7 CS-SE Team Facet Discussion:

Team Discussion: Cognitive Styles

To begin getting to know your team, do the following:

1. Complete the **Cognitive Styles Exploration**
2. Contribute to this team discussion by doing the following:
    - **Describe your thoughts** about the exploration (at least one paragraph)
    - State **which personas** you identify with and describe **two of your facet values**
    - Respond to **two different people**. Include **how your facet values are similar or different** from your teammate's.

Activity8 CS-SE HW3 Design Integration:

UI Design (40pts)

Instructions:

- With your team, **combine your UI design ideas** together
- As a team, **revise** the UI design as needed **based on Sprint 2**
    - **Spread the work** so that everyone helps
    - Put them on Asana (or put a link from Asana)
- Individually do the **Peer Review** assignment
- As a team, **revise** the UI design again **based on the peer reviews**
    - **Spread the work** so that everyone helps
    - The revisions **should have labels / callouts / comments** to show…
        - **What** changed
        - **Who** made the change
        - **Why** the change was made
        - **Which** heuristics the change reflects
    - Put the revised UI design on Asana (or put a link from Asana)
- Your team's UI design may be paper prototypes, or may have gone beyond that---either is fine

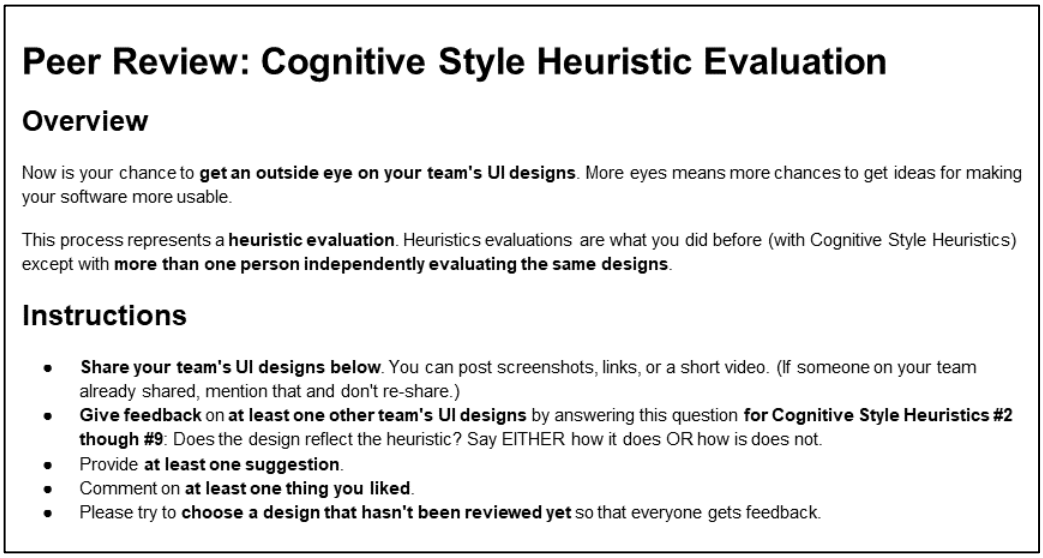
Where are your team's UI designs from **BEFORE the peer review**?

UIDesignURL

Where are your team's UI designs from **AFTER the peer review**? Remember to put your name next to the revisions you made.

UIDesignURL

Activity9 CS-SE Peer Heuristic Evaluation:



Peer Review: Cognitive Style Heuristic Evaluation

Overview

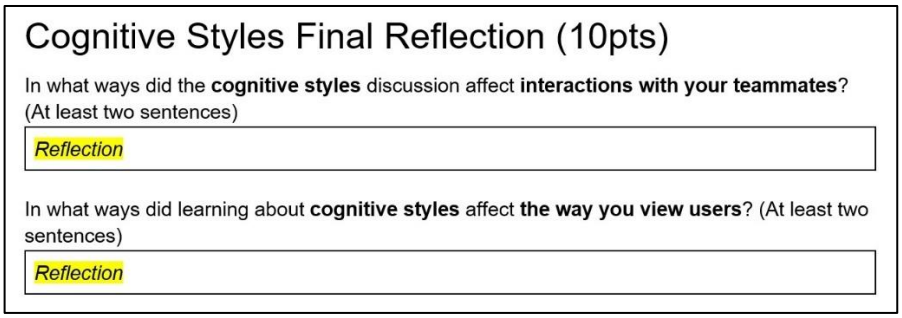
Now is your chance to **get an outside eye on your team's UI designs**. More eyes means more chances to get ideas for making your software more usable.

This process represents a **heuristic evaluation**. Heuristics evaluations are what you did before (with Cognitive Style Heuristics) except with **more than one person independently evaluating the same designs**.

Instructions

- **Share your team's UI designs below**. You can post screenshots, links, or a short video. (If someone on your team already shared, mention that and don't re-share.)
- **Give feedback** on **at least one other team's UI designs** by answering this question **for Cognitive Style Heuristics #2 though #9**: Does the design reflect the heuristic? Say EITHER how it does OR how is does not.
- Provide **at least one suggestion**.
- Comment on **at least one thing you liked**.
- Please try to **choose a design that hasn't been reviewed yet** so that everyone gets feedback.

Activity10 CS-SE HW5 Climate and Users Reflection:



Cognitive Styles Final Reflection (10pts)

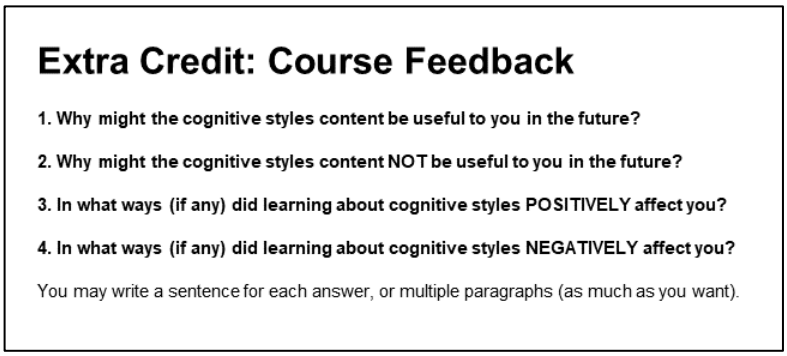
In what ways did the **cognitive styles** discussion affect **interactions with your teammates**? (At least two sentences)

Reflection

In what ways did learning about **cognitive styles** affect **the way you view users**? (At least two sentences)

Reflection

Activity11 CS-SE Course Feedback:



Extra Credit: Course Feedback

1. Why might the cognitive styles content be useful to you in the future?

2. Why might the cognitive styles content NOT be useful to you in the future?

3. In what ways (if any) did learning about cognitive styles POSITIVELY affect you?

4. In what ways (if any) did learning about cognitive styles NEGATIVELY affect you?

You may write a sentence for each answer, or multiple paragraphs (as much as you want).

Appendix C: Questionnaires

CS-DB Post-Questionnaire:

Note: Questions are open answer unless noted otherwise in square brackets (e.g. [Multiple choice])

- "Your major" [Multiple choice with 'Other' field]
- "Your minor (if you have one)" [Multiple choice with 'Other' field]
- "Your degree program" [Multiple choice with 'Other' field]
- "What COURSE CONTENT made you feel EXCLUDED? (e.g., unwelcome, unheard, not accommodated, treated inequitably, like you didn't belong, etc.) List each thing that comes to mind."
- "Your gender identification (e.g., Woman)"
- "Your race/ethnicity (e.g., Hispanic/Latino, Asian, Black/African American)"
- "While you were taking [CS-DB], did you complete an extra credit assignment about Nielsen's or GenderMag heuristics? (If you completed the assignment but did not turn it in, still answer YES.)" [Yes/No]
- "Which software usability heuristics topic did you choose?" [CSH/Nielsen's]
- "Before you did the assignment, had you already learned about [QID101-ChoiceGroup-SelectedChoices]?" [Yes/No/Maybe]
- "What things did you LIKE about the assignment? List each thing that comes to mind."
- "What things did you DISLIKE about the assignment? List each thing that comes to mind."
- "Why might you USE what you learned from the assignment? List each reason that comes to mind."
- "Why might you NOT use what you learned from the assignment? List each reason that comes to mind."
- "In what ways (if any) did the assignment NEGATIVELY affect how you feel about computer science? List each way that comes to mind."
- "In what ways (if any) did the assignment POSITIVELY affect how you feel about computer science? List each way that comes to mind."
- "The assignment increased my interest in computer science." [Likert]
- "What COURSE CONTENT made you feel INCLUDED? (e.g., welcome, heard, accommodated, treated equitably, like you belonged, etc.) List each thing that comes to mind."
- "How likely are you to complete a computer science major/minor?" [Likert]
- "I feel I belong in the computer science major/minor." [Likert]
- "Attach your completed assignment here (up to 100MB, any file type is acceptable)." [File Upload]

CS-SE Pre- and Post- Questionnaires:

Note: Questions are marked with which survey they were included in
Note: Questions are open answer unless noted otherwise in square brackets (e.g. [Multiple choice])

- "Your Major" [Multiple choice with 'Other' field]
- "Your Minor" [Multiple choice with 'Other' field]
- "Your degree program" [Multiple choice with 'Other' field]
- "Your gender identification"
- "Your race/ethnicity"
- PRE/POST: "How likely are you to complete a CS major/minor?" [Likert]
- POST: "Why do you feel that way?"
- PRE/POST: "I feel I belong in the CS major/minor" [Likert]
- POST: "Why do you feel that way?"
- PRE/POST: "I feel represented in the CS major/minor" [Likert]
- POST: "Why do you feel that way?"
- PRE/POST: "If you were designing a software user interface, what would you take into consideration?"
- PRE: "When I work in [online] CS teams, I..." [Likert]
- POST: "Compared to past [online] CS teams, in this course I felt..." [Likert]
- POST: "What has been your professional or personal path leading up to now? Please include numbers of years where you can. (e.g., previous BS degree in Biology, worked as technical program manager for 20 years, stay-at-home parent for last 7 years, etc.)"
- POST: "TA cognitive style profiles incl/excl?" [Likert]
- POST: "Peer review discussion assignments incl/excl?" [Likert]
- POST: "Learning about cognitive styles incl/excl?" [Likert]
- POST: "Discussing cognitive styles with others in the class incl/excl?" [Likert]
- POST: "TA client videos incl/excl?" [Likert]
- POST: "Written feedback from TAs (grading) incl/excl?" [Likert]
- POST: "Instructor videos incl/excl?" [Likert]
- POST: "Announcements incl/excl?" [Likert]
- POST: "¬Gender assumptions in course content incl/excl?" [Likert]
- POST: "Representations of race in course content incl/excl? [Likert]
- POST: "Difficulty of course content incl/excl? [Likert]
- POST: "Options for accessing course content in the way I NEEDED to incl/excl? [Likert]
- POST: "Interaction with classmates ON my team incl/excl?" [Likert]
- POST: "Interaction with classmates OUTSIDE my team incl/excl?" [Likert]
- POST: "Interaction on Piazza or Slack incl/excl?" [Likert]
- POST: "Is there anything you want to clarify about your selections above"
- POST: "Was there anything else during the course that made you feel INCLUDED?"
- POST: "Was there anything else within the course that made you feel EXCLUDED?"
- POST: "Learning about cognitive styles increased my interest in CS." [Likert]

Appendix D: Codesets

CS-DB Codeset:

Code	Description	Examples (researcher generated, no quotes)
Exploration format +	Liked (+) / disliked (-) FORMAT of the exploration	"I liked how the module was structured", "the way the heuristics were laid out helped me learn"
Exploration format -		"the module was too long", "I wishes there were more quizzes interspersed"
Exploration application	Correctly applied the exploration (application = using what was learned on something outside the exploration)	"based on the heuistics, I could improve my mobile app by not only announcing, but explaining how to use the new game boards"
Exploration content +	Got something positive (+) / negative (-) out of the content exploration (e.g., (positive): they found it useful, relatable, helpful, interesting, thought-provoking)	"I found the module really uplifting", "reading through the module made me want to learn more"
Exploration content -		"this module had nothing to do with the course and that annoyed me", "I feel like there's no reason to concentrate on human factors this much"
GenderMag application	Mentioned (by name) AND correctly applied a GenderMag facet	"process-oriented users would really appreciate a step-by-step here"
Want usability in education +	Want (+) / do not want (-) usability concepts in education (in general, or in their own education)	"everyone should learn these heuristics", "I want a whole class on this stuff"
Want usability in education -		"this material has no place in a CS degree", "learning this was a waste of time"
Implied users not like me	Thinking about OTHER diverse USERS (users not like me)	"this module made me realize that some people use software in totally different ways compared to me" *ANTI-EXAMPLE: "some people in the class are more advanced than me"* *ANTI-EXAMPLE: "Abi and Tim are different personas"*
Implied users are different	Acknowledged there are DIFFERENT TYPES of USERS	"this module made me realize that some people use software in totally different ways" ANTI-EXAMPLE: "Abi and Tim are different personas"
Explicit users not like me	Acknowledged there are USERS different from THEM	"not all users sit down and start tinkering with the software like I do" ANTI-EXAMPLE: "some people in the class are more advanced than me" ANTI-EXAMPLE: "users have different levels of technical skill"
Gender	Mentioned gender	"men and women", "non-binary genders", "gender identification"
Included by intervention +	Seemed to feel included (+) / excluded (-) because of intervention content (included = feeling welcomed, heard, accommodated, treated equitably, and/or feeling a sense of belonging)	"when I read the module, I really saw my own styles in there", "I struggle with technology and felt acknowledged when I read the module", "I use a screen reader and was happy the widgets were compatible with it"
Included by intervention -		"I'm a woman and the module made me feel like women are bad a computers", "I'm black and all the examples showed white people", "after doing the assignment, someone started teasing me about being risk averse and I don't like that"
Included by non-intervention +	Seemed to feel included (+) / excluded (-) because of non-intervention content (included = feeling welcomed, heard, accommodated, treated equitably, and/or feeling a sense of belonging)	"the weekly emails from the instructor helped me feel welcomed", "I could tell the TAs cared by how responsive they were"
Included by non-intervention -		"I'm a woman and the textbook kept using 'he' to refer to programmers"

CS-SE Codeset:

Code	Description	Examples (Researcher generated, no quotes)
Validation	Feeling it's OK to be who they are (as a result of learning GenderMag concepts), when they DIDN'T feel OK about themselves before	"going through the course material helped me see there are other people like me" "I've been trying to hide or change who I am to fit in, now I realize the benefits of my cognitive styles"
Peer awareness	Understood something new about fellow CS STUDENTS or CS COLLEAGUES (as a result of learning GenderMag concepts) and mentioned a BENEFIT of that understanding	"The cognitive styles content helped me understand some of my teammates are selective processors like Tim, which helped me understand how they got through the readings so fast" "After doing the team discussion, our team found out we overlap on our cognitive styles in different ways, which made us feel we had things in common" ANTI-EXAMPLE: "I found out my teammate grew up as a Tim but became an Abi" ANTI-EXAMPLE: "Lots of teammates were Tims, so they didn't read the instructions properly"
Implied users are different	Acknowledged there are DIFFERENT TYPES of USERS	"this module made me realize that some people use software in totally different ways" ANTI-EXAMPLE: "Abi and Tim are different personas" ANTI-EXAMPLE: "Accessibility" ANTI-EXAMPLE: "My teammates approached development different because of their cognitive styles"
Explicit users not like me	Acknowledged there are USERS different from THEM	"not all users sit down and start tinkering with the software like I do" ANTI-EXAMPLE: "some people in the class are more advanced than me" ANTI-EXAMPLE: "users have different levels of technical skill"
Preferred	Said they preferred the GenderMag content over some other content	"I can't really get into these UML diagrams as much as the design heuristics" ANTI-EXAMPLE: "I didn't like the UI stuff as much as the design principles" ANTI-EXAMPLE: "I liked the cognitive styles exploration format better"
Cognitive styles	Mentioned words related to GenderMag curriculum (e.g., personas, persona facet values, cognitive/learning styles, cognitive heuristics)	