Inclusivity Bugs in Online Courseware: A Field Study

Amreeta Chatterjee
Oregon State University
Corvallis, Oregon, USA
chattera@oregonstate.edu

Lara Letaw
Oregon State University
Corvallis, Oregon, USA
letawl@oregonstate.edu

Rosalinda Garcia
Oregon State University
Corvallis, Oregon, USA
garciros@oregonstate.edu

Doshna Umma Reddy
Oregon State University
Corvallis, Oregon, USA
ummaredd@oregonstate.edu

Rudrajit Chaudhuri
Oregon State University
Corvallis, Oregon, USA
choudhru@oregonstate.edu

Sabyatha Satish Kumar
Oregon State University
Corvallis, Oregon, USA
sathishs@oregonstate.edu

Patricia Morreale
Kean University
Union, New Jersey, USA
pmorreal@kean.edu

Anita Sarma
Oregon State University
Corvallis, Oregon, USA
anita.sarma@oregonstate.edu

Margaret Burnett
Oregon State University
Corvallis, Oregon, USA
burnett@eecs.oregonstate.edu

ABSTRACT

Motivation: Although asynchronous online CS courses have enabled more diverse populations to access CS higher education, research shows that online CS-ed is far from inclusive, with women and other underrepresented groups continuing to face inclusion gaps. Worse, diversity/inclusion research in CS-ed has largely overlooked the online *courseware*—the web pages and course materials that populate the online learning platforms—that constitute asynchronous online CS-ed's only mechanism of course delivery.

Objective: To investigate this aspect of CS-ed's inclusivity, we conducted a three-phase field study with online CS faculty, with three research questions: (1) whether, how, and where online CS-ed's courseware has inclusivity bugs; (2) whether an automated tool can detect them; and (3) how online CS faculty would make use of such a tool.

Method: In the study's first phase, we facilitated online CS faculty members' use of GenderMag (an inclusive design method) on two online CS courses to find their own courseware's *inclusivity bugs*. In the second phase, we used a variant of the GenderMag Automated Inclusivity Detector (AID) tool to automatically locate a "vertical slice" of such courseware inclusivity bugs, and evaluated the tool's accuracy. In the third phase, we investigated how online CS faculty used the tool to find inclusivity bugs in their own courseware.

Results: The results revealed 29 inclusivity bugs spanning 6 categories in the online courseware of 9 online CS courses; showed that the tool achieved an accuracy of 75% at finding such bugs; and revealed new insights into how a tool could help online CS faculty uncover assumptions about their own courseware to make it more inclusive.

Implications: As the first study to investigate the presence and types of cognitive- and gender-inclusivity bugs in online CS courseware and whether an automated tool can find them, our results reveal new possibilities for how to make online CS education a more inclusive virtual environment for gender-diverse students.

CCS CONCEPTS

• **Human-centered computing → Field studies**; • **Applied computing → E-learning**; • **Social and professional topics → Adult education**; **Gender**.

KEYWORDS

Inclusivity bugs, GenderMag, online CS education

ACM Reference Format:
Amreeta Chatterjee, Lara Letaw, Rosalinda Garcia, Doshna Umma Reddy, Rudrajit Chaudhuri, Sabyatha Satish Kumar, Patricia Morreale, Anita Sarma, and Margaret Burnett. 2022. Inclusivity Bugs in Online Courseware: A Field Study. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.1 (ICER 2022), August 7–11, 2022, Lugano and Virtual Event, Switzerland*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3501385.3543973

1 INTRODUCTION

Fueled by both universities' economic motivations and practical constraints imposed by the pandemic, asynchronous online education has gone from being considered an inferior second-choice to playing an essential role in the higher-education landscape [1, 49, 65]. In tandem with its growing importance, standards have been created to keep online course quality high, and models have been developed for how best to motivate online students and support their learning [24, 28, 49, 71]. During fall 2019, 37% of U.S. post-secondary students were taking at least one online course and 18% were enrolled in entirely online programs [60].

Online education seems to be particularly attractive to diverse students. For example, as of fall 2019, students enrolled at primarily

ICER 2022, August 7–11, 2022, Lugano and Virtual Event, Switzerland

ACM ISBN 978-1-4503-9194-8/22/08…$15.00
https://doi.org/10.1145/3501385.3543973

online institutions are 67% women and 45% racial or ethnic minorities (Black, Hispanic, Asian, Pacific Islander, American Indian, Alaska Native, or more than one race), compared to 57% and 43%, respectively, at other institutions [61].

However, for online education in STEM fields, especially CS, the picture is not so favorable for diverse students, especially when it comes to gender [33]. Research has shown that women and individuals with diverse gender and/or sexual identities join and remain in online CS and STEM education to a much lower extent than men, due to a variety of negative experiences with their online classes [5, 22, 23, 25, 34, 36, 41, 43, 45, 58, 66, 74, 76, 78], as we discuss in detail in Section 2.

A small body of research has begun to address portions of this problem. For example, Kizilcec et al.'s research showed that changing how online CS courses are advertised can help attract more diverse students [42]. Gaucher et al. recommended removing gendered pronouns from online content [29]. Outside of academia, tools have emerged for detecting "male-coded" and "female-coded" text [26, 77]. Other recommendations include improving representation and using neutral visual designs, which we discuss further in Section 2.

However, none of this work has looked into whether online *courseware*—the course web pages and other asynchronous, online materials that, for online students, are the "student-course interface"—is itself a purveyor of inclusivity barriers for diverse students.

To investigate this question, we conducted a three-phase field study. In the first phase, our research question was:

- RQ1: Do gender-inclusivity bugs lurk in online computer science courseware? If so, what kinds, and where are they?

We investigated this question by recruiting experienced online CS faculty at Oregon State University and then helped them do gender-inclusivity evaluations of their own courseware. The evaluation approaches the online CS faculty used were variants of the GenderMag (*Gender* Inclusiveness *Mag*nifier) method [13] for finding, fixing, and/or averting these inclusivity bugs. GenderMag variants have been used in a variety of domains, including digital libraries [19], machine learning interfaces [11], robotics [70], search engines [79], open-source software projects [57, 64], and learning tools and websites [11, 13, 35, 69]. Besides its wide applicability, we chose GenderMag because of its accuracy (low false-positives rate): at least 95% of inclusivity bugs detected with GenderMag also arose with actual users in lab studies or in the field (e.g., [13, 64, 79]).

If the answer to RQ1 is "yes," the next question that arises is whether it is possible for a tool to help online CS faculty find the inclusivity bugs, even if the faculty don't have time to perform GenderMag evaluations manually. To find out, we started with the GenderMag AID (Automated Inclusivity Detector) tool [16] and used it to investigate our remaining two research questions during the second and third phases of our field study:

- RQ2: Can an automated tool based on GenderMag be used to find these inclusivity bugs in online CS courseware?
- RQ3: How do online CS faculty use this tool on their own courseware?

As the first study to investigate the presence and types of cognitive- and gender-inclusivity bugs in online CS courseware and whether an automated tool can find them, the results of our investigation stand to reveal new possibilities for how to make online CS education a more inclusive virtual environment for gender-diverse students.

2 BACKGROUND AND RELATED WORK

2.1 Existing research about inclusivity in online courses

Several factors have been shown to contribute to underrepresentation of certain populations in tech fields. In online CS communities, LGBTQ+ programmers anticipated that, because of the heterosexist climate, few women and LGBTQ+ people would join [23]. Similar problems exist in online CS education too. Studies have shown that women often face "othering" in online learning [66], are less persistent with lectures and assessments [41], have lower retention earlier in CS MOOCs (Massive Open Online Courses) [22], and are less likely than others to complete an online career change program for CS [36]. Krause-Levy et al. reported that, when universities shifted to online learning as a result of COVID-19, first-generation and women students felt a lack of sense of belonging [45]. Research has also shown gender differences in student experiences with CS learning platforms, such as Stack Overflow and Piazza [5, 25, 76, 78]. However, no work has considered whether gender barriers are embedded in the courseware used in online CS education.

Although there is extensive work on CS education's lack of inclusivity, only a few previous studies have looked into how to *improve* inclusivity in CS courses, and even fewer in *online* CS courses. Studies of in-person CS courses have reported that people-oriented tasks and creative expression improve inclusion of women [6, 17, 52]. Pair programming [80, 83], meaningful or socially relevant assignments [7, 9, 51], and leveling the playing field with mechanisms like having everyone start with a language new to all [44] are just a few of the well-known practices increasing recruitment and/or retention across genders in in-person CS education. In online CS, researchers have reported that including gender-inclusive elements in course presentation, improving representation, and using neutral visual designs improve experiences for women in CS [40].

Toward creating inclusive courseware, a number of organizations have created general standards. The Online Learning Consortium's (OLC) five Pillars of Quality Online Education is used by institutions to identify ways to support successful online learning, including access for all [18]. Another well known set of standards is Quality Matters, a set of 50 standards specifically for online and blended higher education courses [55]. This set includes standards for accessibility and usability, but is intended for course designers and requires a membership to access [54]. However, we could not find any online education standards specifically for supporting gender-inclusivity from a cognitive perspective.

2.2 GenderMag: What we wanted online CS faculty to use

Our field study made use of the GenderMag method [13], an evidence-based method for finding, fixing, and averting gender-based cognitive "inclusivity bugs" in technology. GenderMag utilizes extensive

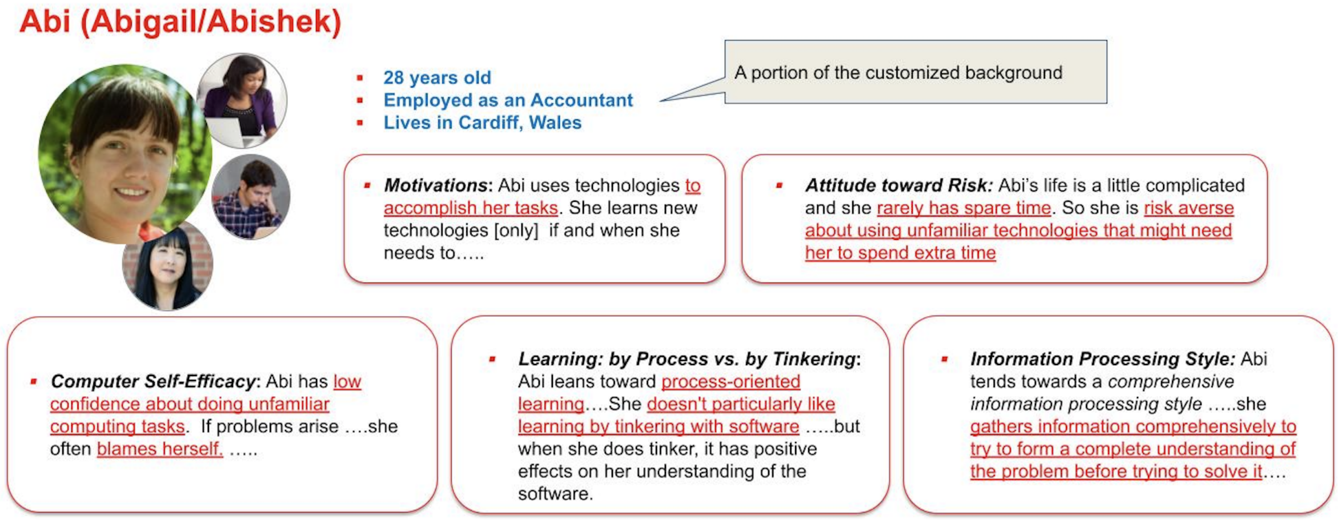


Figure 1: Key portions of the GenderMag Abi persona. Women are statistically skewed toward Abi's facet values (shown here) in how they interact with technology. However, Abi can represent users of any gender. The blue portions of the persona are customizable.

research on how users of different genders tend to interact differently with user-facing technology, using different cognitive styles (cognitive "facets" in GenderMag). The five facets used in GenderMag capture diversity of *motivations* for using tech; *information processing style*; *computer self-efficacy*; *learning style* (by process or by tinkering); and *attitude toward risk*.

The resulting "inclusivity bugs" that GenderMag finds are failures of a technology product to support these five facets' full ranges of values, disproportionately impacting people with particular cognitive styles. They are also gender-inclusivity bugs because the facets capture (statistical) gender differences in how people problem-solve [3, 13–15, 75, 79].

GenderMag uses three personas to bring the facets to life: Abi (Abigail/Abishek), Pat (Patricia/Patrick), and Tim (Timara/Timothy). Each of the five facets has a range of values; Abi's and Tim's values lie at opposite ends, and Pat has values within. The Abi persona represents facet values which disproportionately skew towards women, Tim represents facet values that disproportionately skew towards men, and Pat provides a third set of values [13]:

- Motivations: Abi and Pat are motivated to use tech only as needed for their task. They rarely have spare time and prefer familiar features so they can focus on the task. Tim is motivated to investigate new, cutting-edge features.
- Information processing Style: Abi and Pat gather relevant information comprehensively before acting. Tim likes to delve into the first option and pursue it, backtracking if needed.
- Attitude towards risk: Abi and Pat are risk-averse with tech. They may avoid using features with an unknown time cost and risks. Tim is risk-tolerant so may use unknown features.
- Computer self-efficacy: Abi has lower computer self-efficacy, so if a problem arises when they are trying to use an unfamiliar feature, they blame themselves and stops using the tech. Pat has medium self-efficacy and will try alternative ways of succeeding for a while. Tim has higher computer self-efficacy, so if a problem arises, they'll blame the tech, and may spend extra time finding a solution.
- Learning style: 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.

A portion of the Abi persona is shown in Figure 1; see [10] for the full Abi, Pat, and Tim personas.

There are currently three variants of the GenderMag method:

(1) GenderMag Walkthrough: A systematic, specialized Cognitive Walkthrough (CW) process into which the GenderMag faceted personas are set as follows. While channelling a persona, evaluators "walk through" each step of carrying out a use-case and answer questions about the persona's subgoals and actions for completing the use-case. For example: *Will <Abi/Pat/Tim> have formed this subgoal as a step to their overall goal? (Yes/no/maybe, why, what facets are involved in your answer).* Identifying bugs using this process includes identifying the corresponding facets, which are key to making fixes—an inclusivity bug's fix can be designed around the facet that raised the issue.
(2) GenderMag Moments: An in-the-moment way to utilize the personas and facets to address a particular design question as it comes up [35].
(3) GenderMag Heuristic Evaluation: Similar to Nielsen's Heuristics [62], the GenderMag Heuristics are a set of design principles that can be used to check for gender-inclusivity bugs [73].

In all of these variants, an evaluator draws upon the GenderMag personas and/or individual facets to (manually) perform the evaluation.

The output of a GenderMag evaluation is a set of inclusivity bugs—usability bugs that have been shown to disproportionately impact users with the same cognitive styles as those of the persona used in the evaluation session. The cognitive styles statistically cluster by gender, with Abi's styles statistically more common among women than other people and Tim's styles statistically more common among men than other people [13]. For example, a study with men and women using a search product showed that women's action failure rates were over twice as high as men's. However, after the product owners fixed the gender-inclusivity bugs GenderMag revealed using the Abi and Tim personas, failure rates of both the participating genders went down and the difference between these two genders' failure rates completely disappeared [79].

In the realm of CS education, the only works relating to GenderMag investigate how to teach GenderMag content. Oleson et al.'s action research investigation produced 11 elements of inclusive PCK (Pedagogic Content Knowledge) design that can be leveraged to teach GenderMag in courses [63]. Letaw et al. made use of some of Oleson et al.'s results in teaching GenderMag content in two junior-level online CS courses [46]. Their results showed that the act of learning GenderMag-based content helped students feel more included and more likely to complete the CS major. However, studies have not yet investigated how online CS faculty might use GenderMag to evaluate their own online courseware.

2.3 AID: An automated GenderMag analyzer

Our work builds upon Automated Inclusivity Detector (AID) [16], an open source tool that automates "a vertical slice" of the GenderMag Walkthrough—a portion of its scope (Open Source Software (OSS) projects hosted on GitHub), one of its personas (Abi), and one of its cognitive styles (the information processing style). We further describe this tool in Section 5.

There are multiple usability checkers to automatically evaluate mobile apps [53], webpages [20, 37] and handheld devices [4] based on usability guidelines. For example, the AutoCWW automates the CW to identify website navigation problems. Additionally, there are tools that can automatically detect software accessibility, such as WAVE [38], which checks for accessible media. Similarly, AATT [39] provides an accessibility API to test web applications for conformance to the Web Content Accessibility Guidelines, while Coblis [81] and Vischeck [21] allow developers to simulate colorblindness. However, these tools do not account for gender inclusivity.

Another set of existing tools, such as the gender bias decoder and the gender bias calculator, aim to check for gendered language and biases in text [26, 77]. While these tools improve bodies of text and create inclusive communication, they account for differences in language usage as opposed to differences in how users approach software. Finally, the GenderMag Recorder's Assistant [56] tool assists humans performing the GenderMag Walkthrough and organizing the output, that is, their answers to the Walkthrough questions, additional notes, and screenshots relevant to each question. However, it does not automate the process.

3 FIELD STUDY EDUCATIONAL CONTEXT AND PARTICIPANTS

3.1 Educational context: Online post-baccalaureate CS program

The educational context of our field study was an online post-baccalaureate CS degree program at Oregon State University, a U.S. university. In this program students already have an undergraduate degree in another field, and would like to pursue a second undergraduate degree in CS.

The program is online and asynchronous, in which in-person lectures are replaced with course materials offered through the Canvas course management system. Ten weeks of instruction—which might include videos, readings, active learning exercises, assignments, discussion boards, quizzes, exams, etc.—are grouped together and are made available to students on dates set by the faculty. The courses while fully asynchronous are not self-paced: a cohort of students begins the same week, proceeds through the course according to deadlines on that term's calendar, and completes the course by the same end-date. These courses were previously designed and created in partnership with an instructional designer. Some of the required courses within this program have high enrollment, with over 200 students per offering per term.

The faculty member manages assignment deadlines and discussion forums, teaching assistants, and grading; updates course content as needed; holds office hours; and responds to student queries. Faculty in this program have a Master's or Ph.D. degree in CS. In cases of large courses, the course might have multiple faculty members who share duties and co-facilitate the same students through the same course content.

Nine courses were evaluated as part of the field study: (a) introduction to Object Oriented Programming (CS2), (b) junior-level introductory software engineering (SE1) course covering requirements, design, and project management, (c) another junior-level introductory (SE2) course covering testing and debugging, (d) junior-level algorithms (Algs.), (e) junior-level CS projects (Proj.), (f) junior-level CS Discrete Structures (Disc.Struc.), (g) non-CS-major computer applications (Comp.App.), (h) ethics in CS (Ethics), and (i) a graduate-level non-CS-major course on programming and data structures (Adv.DS). The courses: CS2, SE1, SE2, Algs., Proj., and Disc.Struc. are required for the online CS post-baccalaureate degree at Oregon State University; Comp.App. and Ethics are electives; Adv.DS is required for data analytics majors.

3.2 Participants: Online computer science faculty

Our field study had three phases. In Phase RQ1, several CS faculty (manually) detected inclusivity bugs in their own courseware using variants of GenderMag; in Phase RQ2, researchers used a tool on those courseware to see if it could detect the same bugs; and in Phase RQ3, seven CS faculty used the tool themselves on their own courseware. Phases RQ1 and RQ3 had participants and Phase RQ2 did not; there was one common participant in Phase RQ1 and RQ3.

We recruited participants via email and through CS department professional contacts. Some participants contacted us to participate after hearing about the project through a university mailing list or word of mouth. In total, nine online CS faculty members participated. Participants signed an IRB-approved informed consent form and provided demographic information.

As Table 1 shows, five participants identified as women and four as men. All participants were CS faculty with multiple years experience teaching online CS courses (mean: 6.9 years). Six of the participants were already familiar with GenderMag: four had taught it, one of whom (P1) considered themself an expert. The participants' prior familiarity with GenderMag provided an "acid test[1]" for our field study—their familiarity with the GenderMag facets could have previously influenced them to craft their courseware around GenderMag concepts, without needing a GenderMag evaluation to alert them to their courseware's inclusivity bugs.

4 PHASE RQ1: INCLUSIVITY BUGS IN ONLINE CS COURSEWARE

4.1 Phase RQ1's Methodology

We facilitated three participants (P1, P2, P3) in detecting gender-inclusivity bugs in the courseware of two online CS courses offered through Canvas. They all previously used at least one GenderMag method variant, two of them had previously taught one variant of the GenderMag method (Heuristic Evaluation), and one was a GenderMag expert.

The courses the participants considered during this phase were: (a) an introduction to Object Oriented Programming (CS2) and (b) a junior-level introductory software engineering (SE1) course covering requirements, design, and project management, both of which are required for the online CS post-baccalaureate degree at Oregon State University. Both sets of courseware had been designed to reflect the Quality Matters standards for digital teaching and learning environments [55] but Quality Matters does not address gender-inclusivity. Thus, the evaluations the participants performed in our study aimed to detect cognitive–and gender–inclusivity bugs on courseware that was *already developed under the Quality Matters standards.*

As shown in Table 2, these participants evaluated five use-cases from the perspective of one or more of the GenderMag personas (Section 2). The participants chose these use-cases based on problems they thought their own students might encounter or problems they had already seen among their students.

To evaluate these use-cases, they chose from three variants of the GenderMag method:

- **GenderMag Walkthrough**: P1 and P3 formed a team and used this approach for the first and second use-cases when evaluating CS2 (Table 2). As described in Section 2, a GenderMag Walkthrough session involves answering a set of evaluation questions as the evaluator "walks through" a user interface as one of the GenderMag personas (Abi, in this case). The participants chose to use the Abi persona because Abi has been shown to provide the strongest inclusivity lens [35]. They customized Abi to match some of the characteristics of online students, such as their average age being around 30 (the participants chose 31), being employed while in school, and having previously taken post-secondary courses [27]. During their two evaluation sessions, the participants used the GenderMag Walkthrough forms to write down the gender-inclusivity bugs that Abi would encounter.

- **GenderMag Heuristic Evaluation**: The two participants (P1 and P3) then used this approach for the third use-case (when evaluating SE1). In contrast to the Walkthrough, the GenderMag Heuristic Evaluation [12, 46] involves going through heuristics one by one and identifying inclusivity bugs in software for a chosen use-case and GenderMag persona. The participants chose to use the Abi and Tim personas because they represent opposite ends of the GenderMag cognitive style spectra. They covered three (out of eight) heuristics:
  - Heuristic #1: Explain the benefits of using new and existing features.
  - Heuristic #3: Let people gather as much information as they want, and no more than they want.
  - Heuristic #6: Provide an explicit path through the task

  The participants chose these heuristics based on how relevant they seemed for their courseware. For the purpose of taking notes, one of the participants created the GenderMag Heuristic Evaluation form similar to the GenderMag Walkthrough evaluation form (both available in the Supplemental Document [2]).

- **GenderMag Moments**: P2 used this approach for the fourth and fifth use-cases. As described in Section 2, GenderMag Moments are small pieces of the full GenderMag Walkthrough. During the session, a researcher observed while the participant talked through bugs an Abi-like student might face while attempting the use-case. Unlike with the GenderMag Walkthrough, the participant did not customize the Abi persona; they instead only considered Abi's facet values.

Since not all the participants were GenderMag experts, a researcher with GenderMag expertise was present at each session to provide guidance as needed.

We collected the GenderMag session materials from participants at the end of each session. Included in the session materials were the participants' use-cases, customized Abi persona (Walkthrough only), filled out GenderMag forms (Walkthrough and Heuristic Evaluation; forms included in the Supplemental Document [2]), and observation notes from the GenderMag Moments session. To analyze these data, two researchers collaboratively used affinity diagramming to cluster *how* the bugs could obstruct a student from completing the use-case. The affinity diagrams revealed six categories of gender-inclusivity bugs these participants found in their online CS courseware, which we enumerate in the next section.

4.2 Phase RQ1's Results: Online Courseware's Gender-Inclusivity Bugs—Where and What Kinds?

Participants P1 and P3 found that CS2 students would face inclusivity bugs as soon as they opened the online courseware. The first-day student interaction use-case that P1 and P3 had chosen was, "Figure out what to do during Week 1 and what the Week 1 deadlines are."

[1] A severe or crucial test, as per Merriam-Webster Dictionary.

ID	Gender	Teaching experience	GenderMag experience	Phase(s)	Course evaluated
P1	Woman	3 yrs	Expert+taught	RQ1	CS2 and SE1
P2	Man	4.5 yrs	Used+taught	RQ1	CS2
P3	Woman	2 yrs	Used	RQ1,RQ3	CS2, SE1 and Algs.
P4	Woman	18 yrs	Taught	RQ3	Comp.App.
P5	Woman	8 yrs	Taught	RQ3	Ethics
P6	Man	5 yrs	Used	RQ3	SE2
P7	Man	3 yrs	None	RQ3	Proj.
P8	Man	10 yrs	None	RQ3	Adv.DS
P9	Woman	9 yrs	None	RQ3	Disc.Struc.
		Mean: 6.9 yrs			

Table 1: Field study participants. Nine online CS faculty members participated in one or more phases of the field study. ID = Participant ID; Gender = Self-identified gender; Teaching experience = Years of online teaching experience; GenderMag experience = GenderMag experience (expert, previously taught GenderMag concepts, previously used a GenderMag method variant); Phase(s) = The field study phases for which the participant provided data. Courses evaluated include the 9 CS courses (CS2, SE1, SE2, Algs., Comp.App., Ethics, Proj., Adv.DS, Disc.Struc.)

Use-case	Crs.	Persona(s)	GM variant	ID(s)	Eval. len.
[P1/P3 GenderMag Walkthrough 1 form]: "[Online CS2] student wants to figure out what to do during Week 1 and what the Week 1 deadlines are"	CS2	**Abi**	GM Walk.	P1+P3	60min
[P1/P3 GenderMag Walkthrough 2 form]: "[Online CS2] student wants to figure out what to do to complete the course to be able to plan their term"	CS2	**Abi**	GM Walk.	P1+P3	60min
[P1/P3 GenderMag Heuristic Evaluation form]: "What should a student do to complete Assignment #6"	SE1	**Abi**, **Tim**	GM HE	P1+P3	71min
[P2 GenderMag Moments notes taken by researcher]: "Abi wants to submit Assignment 7"	CS2	**Abi**	GM Moments	P2	30min
[P2 GenderMag Moments notes taken by researcher]: "Abi wants to get started with the course"	CS2	**Abi**	GM Moments	P2	30min

Table 2: Use-cases evaluated by participants during five GenderMag evaluation sessions. Participants evaluated from the perspectives of Abi-like and Tim-like students attempting the task for the first time in their online courseware. Crs = Course; GM variant = Variant of the GenderMag method used for the evaluation. GM Walk. = GenderMag Walkthrough; GM HE = GenderMag Heuristic Evaluation; ID(s) = Participant ID's; Eval. len. = Approximately duration of the GenderMag session during which the use-case was evaluated.

To complete the use-case tasks, these participants wanted students to first navigate to a page named "Syllabus" because that is the page containing the Week 1 deadlines. However, they realized that this expectation was problematic in multiple ways.

First, P1 and P3 realized that a syllabus is often not where weekly deadlines actually are located (e.g., instead deadlines are often in a separate "course schedule" page). Thus, to find the syllabus' Week 1 deadlines, students would have to already be familiar with the course structure or have to "tinker" in the course navigation. P1 and P3 realized that, since students who learn by process (like Abi does) are disinclined to tinker, these students would be at a disadvantage.

> [P1/P3, GenderMag Walkthrough 1 form] "Abi has never seen an online Canvas course at Oregon State University before, so she wouldn't be looking for the syllabus page"
> <Facets: *motivations, information processing style, attitude toward risk, learning by process vs. tinkering*>
> <Category: *3: "newcomer-unfriendly" (Table 3)*>

P1 and P3 then found an even worse problem when they considered what Abi-like students might do instead of following the "Syllabus" link. Since Abi processes information comprehensively, they predicted that such students would read through the entire contents of the course homepage—but still not find a path toward their goal. Contained on the homepage was, among other elements, a welcome message, buttons for "Start Here," "Syllabus," "Announcements," and a "Jump to Module" navigation, but nowhere was there a mention of "Week 1," "weekly deadlines," or the like. The "Start Here" button, which might seem like a promising next step for Abi, led to a long list of pages that also did not contain "Week 1."

> [P1/P3 GenderMag Walkthrough 1 form] "[S]he'd instead want to...read the intro paragraphs...Confusing: There are a lot of options."
> <Facets: *motivations, information processing style, attitude toward risk, learning by process vs. tinkering*>
> <Category: *3: "newcomer-unfriendly" (Table 3)*>

After students introduced themselves to the course—or decided to skip that step— P2 had hoped they would use a "Coming Up" sidebar (like an auto-generated to-do list) to navigate to course assignments. However, the participant realized that Abi-like students were unlikely to click an assignment link on that sidebar without having more information about the assignment first.

> [P2 GenderMag Moments observation notes] "Abi will want to read more first. [Clicking the assignment link without more information] would be tinkering."
> <Facets: *information processing style, learning by process vs. tinkering*>
> <Category: *6: "blocked by information" (Table 3)*>

Once students finally navigated to the assignments, P1 and P3 found more inclusivity bugs—this time in the "assignment requirements" write-ups. For example, a write-up for a "minimum viable product" software project had a list of steps for students to complete, but no indication of the order in which the steps needed to be completed, the benefit of completing the steps in order, or which steps would take the most effort.

> [P1/P3 GenderMag Heuristic Evaluation form] "Should all instructions be followed step-by step; is not clear...[Students might wonder] which section of the assignment they should put maximum effort in"
> <Facets: *motivations, attitude toward risk, learning by process vs. tinkering.*>
> <Category: *5: "benefit/pitfall" (Table 3)*>

> [P1/P3 GenderMag Heuristic Evaluation form] "[Abi] does not know the benefit of doing what they are doing - why make sprint plan"
> <Facets: *motivations, attitude toward risk.*>
> <Category: *2: "focus" (Table 3)*>

P1 then validated this result, recalling a student voicing a confusion with the write-up by asking whether the first step of the assignment was for the current development sprint or the next one.

Even submitting an assignment could be problematic, as P2 found.

> [P2 GenderMag Moments notes taken by researcher] "There are two links and one says 'Load assignment' which is the incorrect link...there is no mention about submitting assignments"
> <Facets: *information processing style.*>
> <Category: *1&4: "deadend," "missing information" (Table 3)*>

In this case, part of the problem was that students had to access a separate website to submit their assignment instead of using Canvas's built-in assignment submission functionality.

As these examples help illustrate, inclusivity bugs existed in each of the use-cases the participants evaluated, even in courseware of participants who are themselves knowledgeable about inclusive design. Participants found 29 bugs throughout their own online courseware. Thirteen of these were in SE1, a course that both contained instruction on GenderMag and was taught by two participants experienced in GenderMag (P1 and P2), one of whom was a GenderMag expert. Table 3 summarizes the complete list of the 29 bugs these participants found by category, what the bugs were, where the bugs were, which personas and facets the courseware was failing to support, and which variants of the GenderMag method were used to detect the bugs.

5 PHASE RQ2: CAN AN AUTOMATED GENDERMAG TOOL FIND ONLINE CS COURSEWARE'S BUGS?

It took the participants more than four hours to evaluate the five use-cases (Table 2). Moreover, each GenderMag session was conducted in teams of two, thereby costing eight human hours. To make this process more efficient, we looked for an automated way to detect inclusivity bugs. This brought us to our second research question: *RQ2: Can an automated tool based on GenderMag find the bugs that faculty members found manually?* We answered this by evaluating an automated approach for capturing the inclusivity bugs that participants found in their courseware.

5.1 Phase RQ2's Methodology

The Automated Inclusivity Detector (AID) [16] was the only tool that automated a "vertical slice" of the GenderMag Walkthrough. It takes as input the website URLs that are to be evaluated, use-cases, subgoals, and actions, similar to what is used in a GenderMag Walkthrough and outputs a list of inclusivity bugs. AID uses five decision rules to detect inclusivity bugs—that would affect Abi's information processing style—on OSS platforms. Only the first three of these rules were relevant to online courseware, as the other rules were domain specific (e.g., issue labels on GitHub projects), and would not apply to online courseware.

The first decision rule considers scenarios in which users like Abi might not be able to find all of the information they require to complete their task. This rule makes use of the subgoals and action texts to identify if terms / information, that users will be looking for, are missing on the webpage. Therefore, this rule could capture inclusivity bugs in categories 3, 4 and 6 (refer to Table 3), which included bugs such as newcomer-unfriendly & unconventional navigation, missing terms, and lack of information.

The second and third rules relate to website navigation. The former checks if links are labeled with a keyword or phrase so that Abi-like users can understand where the link leads from its label [16]. This rule could capture inclusivity bugs also from category 6 ("blocked by information", Table 3). The latter determines whether a destination page provides cues to help Abi understand that they have arrived at the correct location after clicking on a link. Users like Abi might become confused if a webpage uses different words than what a link's description implied. This rule could capture inclusivity bugs in category 1, which included bugs such as links going to the wrong places (Table 3).

To make AID work for online courseware platforms in general, we needed to modify how it accepted the inputs. Specifically, we redesigned it to accept HTML files instead of website URLs. This made the tool independent of specific authentication or access-granting modules specific to particular websites. For example, Canvas or Blackboard may require different authentication protocols from the user login URL. We will refer to the modified version as AID/Courseware in the remainder of the paper.

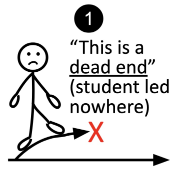
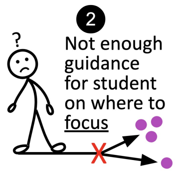
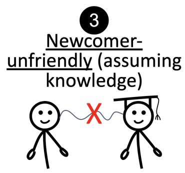
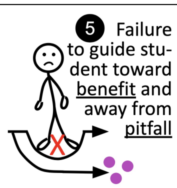
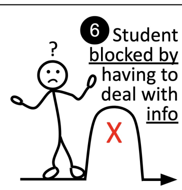
Bug category	Types of bugs found	Where bugs found	Facet (Persona)	GM variant
	Links going to wrong places; No suitable way to create / upload video assignment *Total: 3 bugs of 2 types*	Assign. landing; Assign. integration with 3rd-party app; Assign. resources list	M (Abi) Info (Abi) Risk (Abi) CSE (Abi)	GM HE GM Moments
	Assign. steps / step order not motivated; High-workload steps not emphasized *Total: 4 bugs of 2 types*	Assign. req's	M (Abi) M (Tim) Risk (Abi) Risk (Tim) L-PT (Tim)	GM HE
	Newcomer-unfriendly & unconventional navigation; Lack of reminders / refreshers about terms and concepts *Total: 4 bugs of 2 types*	Homepage; Overview; List of course contents; Assign. req's	M (Abi) Info (Abi) Risk (Abi) L-PT (Abi)	GM HE GM Moments GM Walk.
	Unclear whether course resources are optional or required; Term/label student needs is not on page *Total: 5 bugs of 2 types*	Homepage; "Todo" list; Assign. integration with 3rd-party app	M (Abi) Info (Abi) Risk (Abi) L-PT (Abi)	GM Moments GM Walk.
	Conspicuous path to upload video is the wrong path; Helpful resources not pointed out (Fig 3); Students can complete steps out-of-order but in-order is necessary; Distracting options (Fig 3) *Total: 4 bugs of 4 types*	Assign. resources list (Fig 3); Assign. req's	M (Tim) Risk (Tim) L-PT (Abi)	GM HE
	Vague info.; Too much info. at once; Too many options; Unexplained terminology; Duplicate info. worded inconsistently *Total: 9 bugs of 5 types*	Homepage; Assign. req's; List of course tools; "Todo" list; Assign. resources list (Fig 3)	M (Abi) Info (Abi) Risk (Abi) L-PT (Abi)	GM HE GM Moments GM Walk.
	Total bugs: 29			

Table 3: RQ1 results. Participants used the Abi and Tim personas to locate 29 inclusivity bugs throughout their courseware. They found bugs related to all five GenderMag facets. The distribution of facets used is shown in Fig 2. Purple dots: Educational benefit; Red X's: Source of inclusivity bug. Persona = The persona whose facet found the bug. GM variant = Variant of the GenderMag method used for evaluation. M = Motivations; Info = Information processing style; Risk = Attitude toward risk; CSE = Computer self-efficacy; L-PT = Learning by process vs. tinkering; GM Walk. = GenderMag Walkthrough; GM HE = GenderMag Heuristic Evaluation.

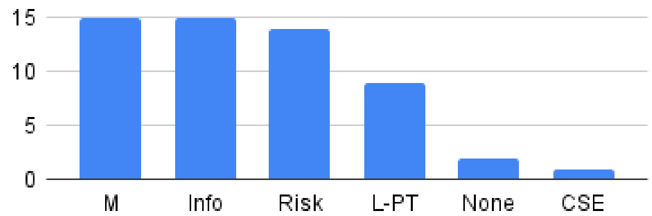


Figure 2: Number of times participants used each GenderMag facet to find inclusivity bugs in their online courseware. M: Motivations; Info: Information processing style; Risk: Attitude toward risk; CSE: Computer self-efficacy; L-PT: Learning by process vs. tinkering; None: No facet found relevant (general usability bug).

Figure 3: Multiple inclusivity bugs on assignment resources list. Students were given links to necessary tools but the online CS faculty did not explain what the tools were for or any benefits/ drawbacks of using them. In their GenderMag evaluation, P3 and P9 decided Abi might not investigate the tools (since they are unexplained and might be a waste of time) and Tim might waste their time tinkering with the tools (since they are unexplained and could be inherently interesting from a technical perspective).

To see how well AID/Courseware (and its underlying decision rules) would perform, we ran the tool on the courseware that the participants evaluated through the GenderMag Walkthrough and GenderMag Moments in Section 4 (refer to Table 2). We did not include the data from the GenderMag Heuristic Evaluation sessions as they do not include subgoals and actions, which are required inputs for the tool. AID/Courseware evaluates a tuple of [use-case, subgoal, action] as its input. There were 24 such tuples when considering the action steps evaluated by the participants in Phase RQ1. These action steps constitute our ground truth to report on the performance of AID/Courseware.

5.2 Phase RQ2's Results: Can a tool automatically find online courseware's inclusivity bugs?

Here we check how well the tool identified inclusivity bugs as compared to the human evaluators. The Phase RQ1 participants identified inclusivity bugs related to Abi's information processing style in 11 out of the 24 action steps. AID/Courseware also identified 11 bugs, of which 8 were correct. Table 4 shows the use-cases evaluated from Section 4, the categories of bugs found in the use-cases and the number of bugs captured by AID/Courseware.

Overall, the tool achieved a 75% accuracy. Despite focusing only on one facet (information processing style), AID/Courseware was able to detect 8 of the 11 bugs that Phase RQ1 participants had found when they considered all five facets through their GenderMag evaluations. There were three false positives (i.e., these bugs were not identified by the participants) and three false negatives (i.e., the tool missed these bugs identified by the participants) in AID/Courseware's results.

False Positives. Two of the false positives occurred because AID/Courseware did not address the underlying semantics of English language usage in courseware [16]. For example, for the subgoal: *"Find the summary of activities and deadlines"* and the action: *"Scroll down to 'Course Summary' and read through it,"* the courseware webpage had a list of assignments such as quizzes, help session surveys, and assignments. However, the tool did not "realize" that quizzes/assignments are analogous to activities in this domain, and when it did not find the keywords from the subgoal/action text (e.g., "activities") it reported an inclusivity bug. Such semantic limitations can be addressed in the future through the use of synonym dictionaries for courseware.

The third false positive instance was in fact not a false positive, but a case where the participants had failed to identify the bug. AID/Courseware reported a bug for the subgoal, *"Go to Gradescope to submit assignment"* and, after performing the action, *"Click on load assignment 7 for submission."* This was reported as a bug since all assignments were referred to as "Projects" on the submission webpage. That is, students would have to click on the link labelled "Project 1" to submit "Assignment 1." This bug falls in Bug category

Use-case	Categories of bugs found	#Bugs found in GenderMag Walkthrough and Moments	#Bugs captured by AID/Courseware
[P1/P3 GenderMag Walkthrough 1 form]: "[Online CS2] student wants to figure out what to do during Week 1 and what the Week 1 deadlines are"	3("newcomer-unfriendly"), 4("missing information"), 6("blocked by information")	✓✓	✓✓
[P1/P3 GenderMag Walkthrough 2 form]: "[Online CS2] student wants to figure out what to do to complete the course to be able to plan their term"	3("newcomer-unfriendly"), 4("missing information"), 6("blocked by information")	✓✓✓	✓✓
[P2 GenderMag Moments notes taken by researcher]: "Abi wants to submit Assignment 7"	1("deadend"), 4("missing information"), 6("blocked by information")	✓✓✓✓✓✓	✓✓✓✓

Table 4: Number of inclusivity bugs AID/Courseware detected that were previously detected manually by Phase RQ1 participants. Each checkmark (✓) in the table indicates a step in the use-case where an inclusivity bug was found for Abi's information processing style facet.

6 (Table 3), where terms are used inconsistently to refer to the same information. The Phase RQ1 Participants had listed a similar problem as a bug in other instances, but failed to do so here. Research on analytic evaluations where humans perform methods in the CW family have shown that humans frequently miss detecting some of the problems [50], which can be up to 70%. GenderMag Walkthrough evaluators, likewise, also miss detecting inclusivity bugs [13, 79].

False Negatives. There were three instances where AID/Courseware failed to identify inclusivity bugs that the participants identified. These misses occurred when the underlying decision rules did not capture these bugs.

The first false negative arose in the use-case: "*[Online CS2] student wants to figure out what to do to complete the course to be able to plan their term*." The course webpage had multiple links with the label "Tools," and would confuse Abi as to which is the correct link to lead them to their goal (Bug category 6). Identifying situations where multiple links with the same label lead to different locations was not a decision rule in AID [16], but one that can be easily implemented in future work.

The other two false negatives were in the third use-case: "*Abi wants to submit Assignment 7*." The Phase RQ1 participants had noted that Abi would require more information before clicking on an assignment link (Bug category 6). Making this determination is subjective and it is difficult to automatically gauge how much information is the right amount of information for the user. The third false negative was for a situation where students were redirected to a third-party platform ("GitHub Classroom") from a link on the Canvas webpage (Bug category 1). The participants felt that this would confuse Abi as to whether they are on the right path. AID/Courseware currently evaluates only pages on a single platform.

Despite these false positives and negatives, the 75% overall accuracy of the tool was promising. AID/Courseware can thus serve as a cognitive aid to help online CS faculty evaluate their own courseware. The tool can pinpoint the inclusivity bugs and why they occur, which online CS faculty can then use to figure out how to fix the bugs. Therefore, we brought the tool to interested online CS faculty to see how they would use AID/Courseware to evaluate their own courseware.

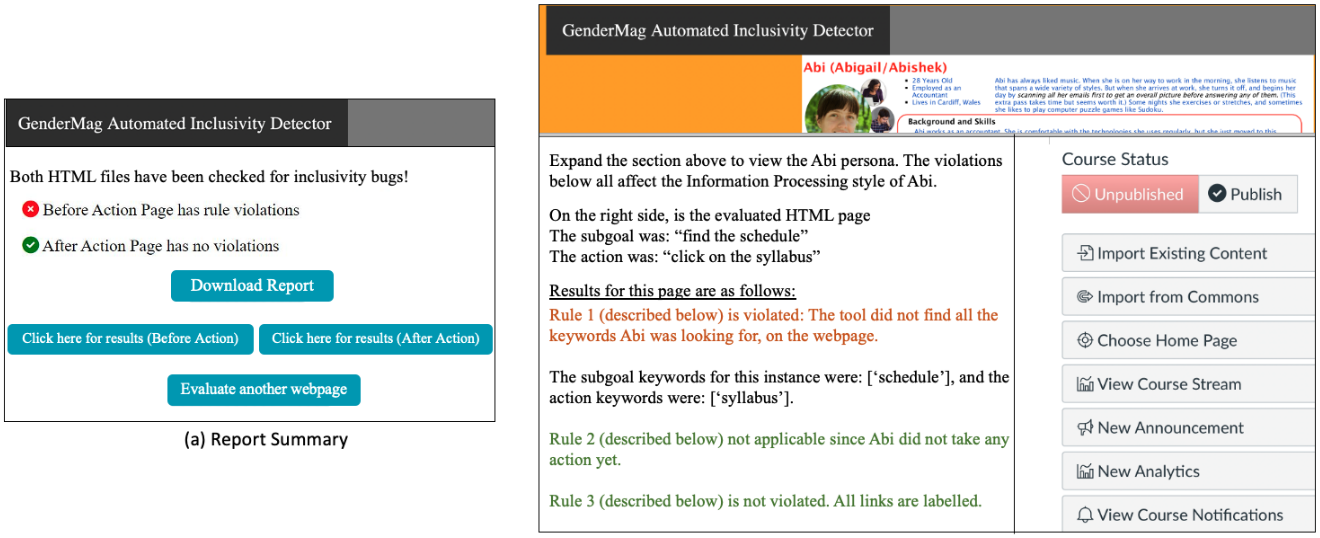
6 PHASE RQ3: THE ONLINE CS FACULTY USING THE TOOL

6.1 Phase RQ3's Methodology

6.1.1 The AID/Courseware GUI. Before asking participants to use AID/Courseware, we developed a Graphical User Interface (GUI). This was done to help participants who were unfamiliar with such inclusivity evaluations and provide a familiar approach for those who already used other GUI auto-checker tools. The GUI guides the user through each step of AID/Courseware. The first page explains the decision rules with examples. The next page, an upload page, then prompts the user for the use-case details. With this information, the GUI generates a results page, as shown in Figure 4a, where it presents a summary of any rule violations with options to download the report as a CSV file or view detailed results for each evaluated page, as shown in Figure 4b. All steps of the tool are explained in detail, along with figures, in the Supplemental Document [2].

Elements of the GUI (such as the button styles) were taken from the open-source Recorder's Assistant [56] (described in Section 2). Since the AID/Courseware decision rules were implemented in Python, we integrated them with the GUI using Flask micro web framework [68], also written in Python. We deployed the application using the Heroku Cloud Platform and tested it on Windows 10 and MacOS X using Google Chrome, Microsoft Edge, and Safari. The GUI design was finalized after multiple pilots and iterations.

6.1.2 Field study with online CS faculty. In Phase RQ3, we investigated how participants used the AID/Courseware GUI to evaluate their own courseware. To capture participants' thought processes while using the GUI, we used a think-aloud protocol followed by a semi-structured interview. Prior to the study, we refined the study



(a) Report Summary

(b) Detailed Report

Figure 4: Portions of the AID/Courseware graphical user interface we created. (a) Results presented by AID/Courseware for two HTML files provided by the user. (b) Details for the first HTML file explaining why Rule 1 was violated.

task and interview questions through pilots with online CS graduate teaching assistants. All study sessions were audio-taped and transcribed.

We contacted 14 online CS faculty members and sent a sign-up survey (included in Supplemental Document [2]) to those who expressed interest. Seven faculty members (P3-P9) signed up, one of whom (P3) also participated in Phase RQ1. Participants provided demographic information and identified courseware from one of their online courses to evaluate with AID/Courseware (shown in Table 1).

Some of these participants were already familiar with GenderMag. On a scale of one to ten, four out of the seven participants rated their familiarity with GenderMag as a five or higher. Two participants, P3 and P4, had incorporated GenderMag into their courses. For example, P4 had incorporated inclusivity lessons about the GenderMag personas into their course:

> [P4 interview] "I do have the [GenderMag] lesson in chapter 1 now."

Each participant completed an hour long study session using video conferencing software (Zoom). This allowed them to share their screen during the study while also keeping their camera off for privacy. Participants used their own equipment and courseware so we could observe them in their normal work environment (they all worked from home).

Each study session began with an introduction to the study and AID/Courseware. The study facilitator led the participant through an example to explain AID/Courseware's purpose and usage. Participants then practiced thinking aloud. To preface their AID/Courseware evaluation, they also walked through their course and explained what they expected students to do during the first week.

Participants then used the AID/Courseware GUI to evaluate one complete use-case for their courseware. Because some participants were not familiar with the GenderMag Walkthrough, we pre-defined the use-case for the task: "*The student wants to get started with the course.*" We chose a newcomer use-case as it is similar to the use-cases evaluated in Phase RQ1. Moreover, this use-case would apply to every course.

We then conducted semi-structured interviews to ask participants about their familiarity with inclusivity methods, existing inclusivity practices, experience with the tool, and takeaways from the evaluation. Interview questions included the following (see Supplemental [2] for the full set of questions):

- Are inclusivity evaluations a part of your course designing process? If yes, can you provide an example?
- If there was an automated way to help you detect inclusivity issues, what would you expect from such a tool?

6.1.3 Data analysis. We transcribed each participant's think-aloud audio recordings and interview sessions and used inductive thematic analysis to identify themes [8]. In the first step, two authors read the transcripts and the study notes several times to become familiar with the data. Relevant phrases and sentences were labeled by both authors on one Google Doc, using a constant comparative approach. Throughout this process, as more themes emerged, previously coded transcripts were revisited. This cycle of descriptive coding captured participant actions while using the tool and their reflections on courseware designs. In the next phase, the authors performed axial coding by comparing and contrasting themes to look for connections among them and ensuring there was no repetition. This step was also repeated multiple times, before merging into the final themes. (See [2] for an example)

6.2 Phase RQ3's Results: The online CS faculty and the tool—with their own courseware

6.2.1 Inclusivity checking online courseware: Not a thing? Beyond using accessibility checkers, participants' interviews suggested that the concept of inclusivity-checking their courseware was not an explicit part of their workflows. Two of this phase's seven participants, P3 and P8, did not report any form of inclusivity checking at all. P4 and P7 reported use of accessibility checkers, but no other form of inclusivity checking.

The remaining three participants, as well as P7, reported using informal processes that they had used to try to be mindful of certain inclusivity bugs. The bugs these four participants reported trying to look out for were vocabulary issues, information visibility, multiple types of content, and pronouns:

> [P6 interview] "...it is an unneeded barrier to have highfalutin vocabulary."

> [P9 interview] "... written in very easy language. Or the links are very visible so that it's not hidden or you don't have to work too much."

> [P5 interview] "There are things I will include that have some videos and audios... "

> [P7 interview] "My course designer will be looking for accessibility... and... things like pronouns..."

However, none of the participants reported having any formal/systematic process for evaluating their online courseware for inclusivity bugs beyond accessibility.

6.2.2 The faculty meet the tool-identified bugs. All seven Phase RQ3 participants found inclusivity bugs in their courseware when they used the tool. P4, P8, and P9 began by thinking through the tool's rules that were summarized on the tool's landing page. For example:

> [P9 think-aloud]: "Rule 1 and rule 2 and rule 3. You walked me through those, how to use them and these are very properly written...what [AID/Courseware] actually wants to do."

Other participants were more output-driven, waiting to see what the tool would say first and then reflecting upon its results:

> [P8, reflecting upon the tool-identified bug]: "...somewhere on here should have...had the words tool or module and it didn't."

> [P7, reflecting upon the tool-identified bug] "The action was...Click on Start Here module. The tool did not find a schedule... that Abi is trying to do..."

Participants were sometimes surprised by a tool-identified bug. In some of these cases, the surprising bug identification led them to realize that a portion of their courseware that they had thought to be unproblematic actually did have inclusivity bugs:

> [P6 interview] "[a tool-identified bug] is an oversight I didn't know existed."

> [P5 interview] "...having a fresh eye or a fresh automated tool to say, Hey, you think you may have said this but you never did. I think is very useful."

> [P6 interview] "...if I hadn't looked ... and you said, do you think it's easy for the students to find a schedule, then yeah absolutely. But now I'm not so sure...that was definitely enlightening"

But participants did not always agree with what the tool's output seemed to be suggesting they do:

> [P8, while going through the evaluation report] "[AID/Courseware] would have expected to see [the word] 'tools' on the homepage, but I'm not sure I necessarily would agree..."

Still, despite initially disagreeing with the particular bug the tool had identified, P8 generalized upon the tool's report to consider whether instances similar to that particular bug might need fixing:

> [P8, while going through the evaluation report] "I'm not sure I would want to put 'tools' here [homepage], but I do think that it would make sense to better include that information on [different page in the courseware] because [installing the tools] is a key part [for the course]."

Some generalized even further, starting to notice other potential inclusivity bugs in their courseware that did not closely resemble any bugs that the tool had even reported yet. For example:

> [P6, while walking through their course] "I think on most screens this next button's obscured...So that's probably not a good indication."

> [P4, discussing the subgoal for the evaluation] "I put that schedule on the homepage so that students wouldn't have to hunt for it. But some of them still try to find it somewhere else, even though it's sitting right there."

6.2.3 Getting from courseware bugs to courseware fixes. The way AID/Courseware finds and reports inclusivity bugs is analogous to a concatenation of testing and fault localization. Like testing, AID/Courseware identifies that bugs are present; and like fault localization, AID/Courseware points out where those bugs are. In the case of courseware, the location of inclusivity bugs is not in source code, but rather in the information arrangements, labeling, objects and structures the students see.

The AID/Courseware tool can thus be thought of as an automation of testing plus the first two parts of Guizani et al.'s Why/Where/Fix inclusivity debugging process [30]. The "Why" part is the facet that explains why the bug exists, which in the current AID/Courseware prototype is always information processing style. The "Where" part (fault localization) is the location (information element, label, structure, etc.) that the tool points out.

Guizani et al. pointed out that the "Fix" part of inclusivity debugging is a matter of fixing the "Where"s in a way that satisfies the "Why"s. The participants in our study did exactly that when thinking about how to fix the tool-identified bugs in their courseware. For example:

> [P7 interview] "...maybe I should just send them to modules...instead of the start here page..."

> [P5 interview] "Definitely will be implementing a change into that course...So really just a guide [saying] this is where you'll find x and y."

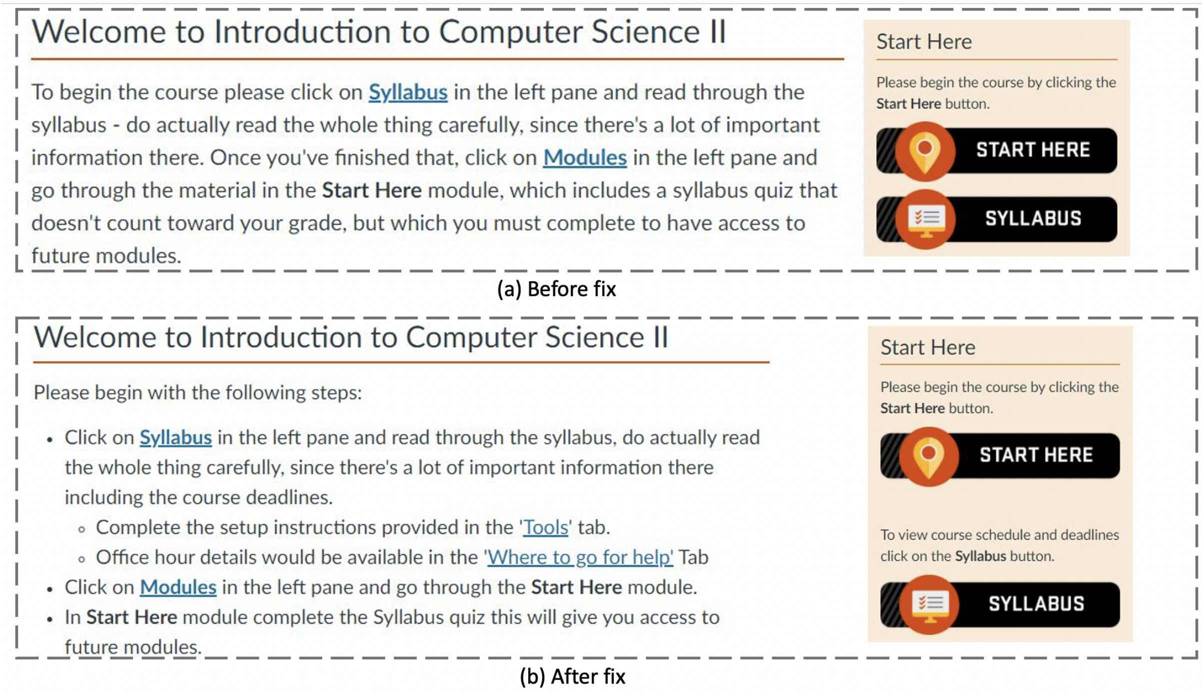


Figure 5: The inclusivity bug P3 found and fixed: (a) Before, there was little structure to the information provided and little guidance on how to get to the course schedule. (b) The fix structured the information, and added labeling to guide the student to the Syllabus for the schedule and deadlines.

> [P3 interview] "...since finding schedule is very important for students to start the course, I can afford to add a sentence for that.."

> [P8 interview] "[I will be] at least adding to the module outcomes, that one of the outcomes is going to be installing the software required for the course."

Although the field study ended only 10 days ago, P3 has already implemented their courseware fix, which is shown in Figure 5.

In summary, the results showed that every Phase RQ3 participant found inclusivity bugs in their courseware. Even participants who had previous inclusivity experience found inclusivity bugs they had not recognized before in their courseware.

Finally, P3, the only participant who had evaluated their courseware manually in Phase RQ1 and used the tool in Phase RQ3, remarked on the time and cognitive load savings that using the tool brought to them:

> [P3 interview] "...because the time that GenderMag evaluation walkthrough basically took...mentally draining to think like somebody else and then fill in the notes... So [AID/Courseware] just removes all that load from me. And it just gives the results, which is what I'm interested in."

7 DISCUSSION

The inclusivity bugs highlighted in this work, in addition to being general instructional design considerations for online courses, put the focus specifically on gender-inclusion. This work has shown that, even when online courseware is built to follow established quality standards such as Quality Matters [55], it can still fail to be gender-inclusive and cognitively inclusive.

7.1 Potential effects of inclusivity bugs on students

Inclusivity bugs in individual courseware might not *prevent* a student from proceeding in a particular course, but running into inclusivity barriers again and again that end in "microfailures" could be discouraging. For example, if faculty assume students know as much as they do about the online courseware (category 3 in Table 3), newcomer students might feel lost as soon as they arrive, and risk-averse, process-oriented learners might not be willing to "click around" on parts of the courseware that seem irrelevant or risky—but this might be the only way provided for the student to learn how the course is structured. Such momentary frustrating experiences in CS courses are associated with poor student outcomes like lower grades, especially among students with low self-efficacy [47]. Rahman [67] also reported that online students tend to have difficulty becoming part of their peer community within their classes.

Previous work has shown that the Phase RQ1 six bug categories are connected to supporting marginalized students in their learning environments. Table 5 summarizes evidence connecting each of the bug categories to marginalized student communities and corresponding GenderMag facets.

Bug Category	Facet values for "Abi"	Marginalized students & educational context	Connection of bugs to other research
1 ("deadend"): No suitable way to proceed in the course	Lower computer self-efficacy compared to their peers	Women in CS1 in a university	Women responded to performance feedback early in the course, revising their self-efficacy beliefs earlier, suggesting that responses to early failures (unable to proceed) in CS could be causing women students to disengage from CS [48].
2 ("focus"): Not enough guidance for student on where to focus (process-oriented)	Learning Style: Learning by process	Women in post-secondary introductory science courses	Women's scientific beliefs, scientific motivation, and scientific thinking and learning improved when a rubric (outcome-based assessment criteria) was added to their course (guiding them on where to focus) [59].
3 ("newcomer unfriendly"): Unconventional course navigation and lacking review of previous content	Information Processing Style: Comprehensive	Older students and women in four edX MOOC courses	Women, older students, and those from countries with lower student-teacher ratio were more comprehensive (first reason about context before focusing on details) and non-linear when navigating through the course (with more backjumps to review previous content) [31].
4 ("missing information"): Student has not formed a complete understanding of course content or comprehensive information is not available	Information Processing Style: Comprehensive	First-generation, low-income, and multi-racial students in 61 post-secondary courses	When course content was supplemented with more comprehensive explanations of why teachers structured assignments in certain ways, first-generation, low-income, and multi-racial students' academic confidence, sense of belonging, and performance increased (comprehensive information processing supported) [82].
5 ("benefit/pitfall"): Failure to guide students to benefits (e.g., what would help them complete task) and away from pitfalls (e.g., options unrelated to task)	Motivations: Task-motivated	Girls in early childhood education	During free play, girls constructed with a goal in mind (concentrating on completing task) compared to boys' activities, which centered on technical process (not necessarily related to task) [32].
6 ("blocked by information"): Student blocked by having to deal with information or vague information	Motivations: Task-motivated	Muslim women in public high schools in Canada	Students attempting to do an assignment (complete their task) lost motivation when assignment requirements were unclear and ambiguous (vague information) [72].

Table 5: Gender-inclusivity bugs vs. Learning environment: Examples from literature tying inclusivity bug categories to "Abi" GenderMag facet values shown by marginalized students and issues they have faced in their learning environments.

7.2 Using an inclusivity checker like AID/Courseware: From course preparation to advocacy

By the end of Phase RQ3, the participants began reflecting upon how they envisioned using the tool in their own courseware design workflows. For example, P3, P5, and P9 envisioned using the tool early while (re-)designing a courseware artifact. As P5 put it, AID/Courseware is "something that can be included in a prep process" [P5 interview]—perhaps at the same point in time as one would use an automated accessibility checker.

But some participants wanted more frequent inclusivity checking, perhaps whenever they make an edit to any aspect of the courseware. For some online courses, this can happen as often as everyday, which may explain P6's comment: "at the end of every day more periodically" [P6 interview].

P7 was one of the participants who found inclusivity bugs that seemed beyond their control, because the bug was inherent in the learning platform (here, Canvas) or in university-mandated guidelines. For situations like this, P7 saw AID/Courseware's automated reports as a communication tool, which they could use to argue for particular inclusivity changes in the platform or guidelines: "at least could show the difficulty that a student faces" [P7 interview]. P7 felt quite strongly about the need for change, and by the end of the interview had decided that their next step would be to go "talk to the various people involved" in creating the standards and platforms they used [P7 interview] to advocate for change.

P8 shared P7's interest in inclusivity changes to the platform and guidelines, but envisioned a different path forward to achieve that goal. Because the instructional designers at Oregon State University are the ones who create many of the online course guidelines, P8 saw "getting [instructional designers] access to the tool and having them do the analysis" [P8 interview] as the way to enact changes in online course guidelines.

These participants' reflections suggest a variety of possible uses for a courseware inclusivity checker like AID/Courseware, ranging from an inclusivity design aid to a regularly run "inclusivity regression tester" to a tool for advocating change.

7.3 Limitations

Like most other field studies, our study eschewed controls in order to gather data about real online CS faculty members' real courseware instances in use-cases that mattered to them. Thus, we did not set the scope of our investigation: the online CS faculty did. Specifically, the three Phase RQ1 participants chose which two courses they were interested in evaluating, which use-cases they wanted to evaluate (five), and which personas to evaluate with (Abi and Tim). A broader set of evaluations from a broader set of individuals could produce a different set of RQ1 results.

Participants' RQ1 findings were used as the "ground truth" for RQ2, since we relied on participants' expertise as educators to be able to identify what might be problematic for online students. Future work with students could further validate participants' findings.

AID/Courseware checks for inclusivity bugs affecting only the Abi persona's information processing style. Since the rules are not exhaustive, further research is required to investigate decision rules for more facets and more personas. Further research is also needed to understand how we could capture bugs from category 2 ("not enough guidance on where to focus") and category 5 ("failure to guide student toward benefit"), which were identified through GenderMag Heuristic Evaluation.

All of the participants worked at the same university within the same CS department and online CS program; this could limit the generalizability of the RQ1 and RQ3 results (e.g., because other universities use different course management systems, use different standards for building online courseware, etc.). This limitation can only be addressed by additional empirical studies 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 such approaches for evaluating online CS courseware.

8 CONCLUSION

In this paper, we have presented a three-phase field study in which online CS faculty evaluated their own online courseware to find its inclusivity bugs. As the first study to investigate the presence and types of cognitive- and gender-inclusivity bugs in online CS courseware and whether an automated tool can find them, our investigation produced some surprising results:

(1) Experienced online CS faculty participants found inclusivity bugs throughout their own courseware, *despite the courses having been designed around established standards for online courses.* Further, they found a lot of these bugs—29 of them, in all five use-cases and all nine courses they considered. (From RQ1 & RQ3)
(2) The AID/Courseware tool found 8 of the 11 bugs the human evaluators found, covering all the same bug categories. Thus, the tool did about 3/4 as well as human evaluators—*despite using only one facet (information processing style)*, compared to the humans using all five facets. (From RQ2)
(3) Every participant's courseware had at least one inclusivity bug, *even courseware created by faculty with inclusivity experience.* These inclusivity bugs had previously gone unnoticed, even though two thirds of the participants were already familiar with GenderMag, four were teaching it, and one was an expert. (From RQ1 & RQ3)

Perhaps most important, our results suggest that online courseware–the course web pages and other asynchronous, online materials that, for online students, are the "student-course interface"—is erecting barriers to inclusivity in online computer science courses. This adds yet another set of inclusivity barriers to the many other negative experiences that research shows await underrepresented genders—experiences their gender-majority peers are less likely to encounter. Fortunately, our results also suggest that the AID/Courseware tool can help to pinpoint the inclusivity barriers that lurk within online CS courseware, so that faculty know what to fix. We invite everyone interested to try it out on their own courseware, and to help expand this emerging tool on the Open Source site.[2]

ACKNOWLEDGMENTS

We thank the participating instructors for their interest and time in evaluating their own online courseware to address the inclusivity bugs that lurked within. This work was supported in part by the OSU Ecampus Research Fellows Program, the OSU Center for Research in Engineering Education Online, the Center for Inclusive Computing at Northeastern University, NSF 1901031 and 2042324, and joint support by NSF and USDA-NIFA under 2021-67021-35344. Any opinions, findings and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] I Elaine Allen and Jeff Seaman. 2010. *Learning on demand: Online education in the United States, 2009.* Babson Survey Research Group, United States of America.
[2] Rosalinda Garcia Doshna Reddy Rudrajit Chaudhuri Sabyatha Satish Kumar Patricia Morreale Anita Sarma Amreeta Chatterjee, Lara Letaw and Margaret Burnett. 2022. *ICER_2022_Supplemental_Documents_03242022_1016.* Oregon State University. https://doi.org/10.6084/m9.figshare.19412204 (accessed Mar. 2022).
[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] Fiora T. W. Au, Simon Baker, Ian Warren, and Gillian Dobbie. 2008. Automated Usability Testing Framework. In *Proceedings of the Ninth Conference on Australasian User Interface - Volume 76* (Wollongong, Australia) *(AUIC '08)*. Australian Computer Society, Inc., AUS, 55–64.
[5] 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

[2]https://github.com/Amreeta-18/AID-GUI

[6] William H. Bares, Bill Manaris, Renée McCauley, and Christine Moore. 2019. Achieving Gender Balance through Creative Expression. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 293–299. https://doi.org/10.1145/3287324.3287435

[7] Lecia J Barker, Charlie McDowell, and Kimberly Kalahar. 2009. Exploring factors that influence computer science introductory course students to persist in the major. *ACM Sigcse Bulletin* 41, 1 (2009), 153–157.

[8] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[9] 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.

[10] Margaret Burnett. 2021. *GenderMag personas*. The GenderMag Project. https://gendermag.org (accessed Mar. 2022).

[11] Margaret Burnett, Anicia Peters, Charles Hill, and Noha Elarief. 2016. Finding Gender-Inclusiveness Software Issues with GenderMag: A Field Investigation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2586–2598. https://doi.org/10.1145/2858036.2858274

[12] Margaret Burnett, Anita Sarma, Claudia Hilderbrand, Zoe Steine-Hanson, Christopher Mendez, Christopher Perdriau, Rosalinda Garcia, Catherine Hu, Lara Letaw, Aishwarya Vellanki, and Heather Garcia. 2021. *Cognitive Style Heuristics*. The GenderMag Project. https://gendermag.org/Docs/Cognitive-Style-Heuristics-from-the-GenderMag-Project-2021-03-07-1537.pdf (accessed Mar. 2022).

[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 (2016), 760–787.

[14] 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.

[15] Gary Charness and Uri Gneezy. 2012. Strong Evidence for Gender Differences in Risk Taking. *Journal of Economic Behavior & Organization - J ECON BEHAV ORGAN* 83, 1 (06 2012), 50–58. https://doi.org/10.1016/j.jebo.2011.06.007

[16] Amreeta Chatterjee, Mariam Guizani, Catherine Stevens, Jillian Emard, Mary Evelyn May, Margaret Burnett, Iftekhar Ahmed, and Anita Sarma. 2021. AID: An automated detector for gender-inclusivity bugs in OSS project pages. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1423–1435.

[17] Ingrid Maria Christensen, Melissa Høegh Marcher, Paweł Grabarczyk, Therese Graversen, and Claus Brabrand. 2021. Computing Educational Activities Involving People Rather Than Things Appeal More to Women (Recruitment Perspective). In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 127–144. https://doi.org/10.1145/3446871.3469758

[18] Online Learning Consortium. 2022. *Quality Framework*. OLC. https://onlinelearningconsortium.org/about/quality-framework-five-pillars/ (accessed Mar. 2022).

[19] 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, Springer International Publishing, Cham, 45–50.

[20] Alexiei Dingli and Justin Mifsud. 2011. Useful: A framework to mainstream web site usability through automated evaluation. (2011).

[21] Bob Dougherty and Alex Wade. 2008. *Vischeck*. Vischeck.com. http://www.vischeck.com/ (accessed Mar, 2022).

[22] Rodrigo Duran, Lassi Haaranen, and Arto Hellas. 2020. *Gender Differences in Introductory Programming: Comparing MOOCs and Local Courses*. Association for Computing Machinery, New York, NY, USA, 692–698. https://doi.org/10.1145/3328778.3366852

[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] 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.

[25] 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.

[26] Tom Forth. 2013. *Gender bias calculator*. https://www.tomforth.co.uk/genderbias/ (accessed Mar. 2022).

[27] Jordan Friedman. 2017. *U.S. News Data: The Average Online Bachelor's Student*. U.S. News & World Report L.P. https://www.usnews.com/higher-education/online-education/articles/2017-04-04/us-news-data-the-average-online-bachelors-student

[28] 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.

[29] Danielle Gaucher, Justin Friesen, and Aaron C Kay. 2011. Evidence that gendered wording in job advertisements exists and sustains gender inequality. *Journal of Personality and Social Psychology* 101, 1 (2011), 109.

[30] Mariam Guizani, Igor Steinmacher, Jillian Emard, Abrar Fallatah, Margaret Burnett, and Anita Sarma. 2022. How to Debug Inclusivity Bugs? A Debugging Process with Information Architecture. *ACM/IEEE ICSE Software Engineering in Society (ICSE-SEIS'22)* (2022), 12 pages.

[31] Philip J Guo and Katharina Reinecke. 2014. Demographic differences in how students navigate through MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference*. 21–30.

[32] Jonas Hallström, Helene Elvstrand, and Kristina Hellberg. 2015. Gender and technology in free play in Swedish early childhood education. *International journal of technology and design education* 25, 2 (2015), 137–149.

[33] Karen Hamrick. 2022. *Women, Minorities, and Persons with Disabilities in Science and Engineering*. NSF National Center for Science and Engineering Statistics (NCSES). https://ncses.nsf.gov/pubs/nsf19304/digest/field-of-degree-women#computer-sciences (accessed Mar. 2022).

[34] John D Hansen and Justin Reich. 2015. Democratizing education? Examining access and usage patterns in massive open online courses. *Science* 350, 6265 (2015), 1245–1248.

[35] Claudia Hilderbrand, Christopher Perdriau, Lara Letaw, Jillian Emard, Zoe Steine-Hanson, Margaret Burnett, and Anita Sarma. 2020. Engineering Gender-Inclusivity into Software: Ten Teams' Tales from the Trenches. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 433–444. https://doi.org/10.1145/3377811.3380371

[36] 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.

[37] Melody Y Ivory. 2000. Web TANGO: towards automated comparison of information-centric web site designs. In *CHI'00 extended abstracts on Human factors in computing systems*. 329–330.

[38] Leonard R Kasday. 2000. A tool to evaluate universal Web accessibility. In *Proceedings on the 2000 conference on Universal Usability*. 161–162.

[39] Prem Nawaz Khan, Cathy O'Connor, and Srinivasu Chakravarthula. 2021. *Automated Accessibility Testing Tool (AATT)*. AATT. https://github.com/paypal/AATT (accessed Mar, 2022).

[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] René F. Kizilcec and Sherif Halawa. 2015. Attrition and Achievement Gaps in Online Learning. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (Vancouver, BC, Canada) *(L@S '15)*. Association for Computing Machinery, New York, NY, USA, 57–66. https://doi.org/10.1145/2724660.2724680

[42] René F Kizilcec and Andrew J 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*. 1–10.

[43] René F Kizilcec, Andrew J Saltarelli, Justin Reich, and Geoffrey L Cohen. 2017. Closing global achievement gaps in MOOCs. *Science* 355, 6322 (2017), 251–252.

[44] Maria Klawe. 2013. Increasing female participation in computing: The Harvey Mudd College story. *Computer* 46, 3 (2013), 56–58.

[45] Sophia Krause-Levy, William G. Griswold, Leo Porter, and Christine Alvarado. 2021. The Relationship Between Sense of Belonging and Student Outcomes in CS1 and Beyond *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 29–41. https://doi.org/10.1145/3446871.3469748

[46] Lara Letaw, Rosalinda Garcia, Heather Garcia, Christopher Perdriau, and Margaret Burnett. 2021. Changing the Online Climate via the Online Students: Effects of Three Curricular Interventions on Online CS Students' Inclusivity. In *Proceedings of the 17th ACM Conference on International Computing Education Research*. ACM, Online, 42–59.

[47] Alex Lishinski and Joshua Rosenberg. 2021. All the Pieces Matter: The Relationship of Momentary Self-efficacy and Affective Experiences with CS1 Achievement and Interest in Computing. In *Proceedings of the 17th ACM Conference on International Computing Education Research*. 252–265.

[48] Alex Lishinski, Aman Yadav, Jon Good, and Richard Enbody. 2016. Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals, and Self-Efficacy on Performance. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Melbourne, VIC, Australia) *(ICER '16)*. Association for Computing Machinery, New York, NY, USA, 211–220. https://doi.org/10.1145/2960310.2960329

[49] Patrick R Lowenthal and Charles B Hodges. 2015. In search of quality: Using quality matters to analyze the quality of massive, open, online courses (MOOCs). *International Review of Research in Open and Distributed Learning* 16, 5 (2015), 83–101.

[50] Thomas Mahatody, Mouldi Sagar, and Christophe Kolski. 2010. State of the art on the cognitive walkthrough method, its variants and evolutions. *Intl. Journal of Human–Computer Interaction* 26, 8 (2010), 741–785.

[51] Jennifer Mankoff. 2006. Practical service learning issues in HCI. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 201–206.

[52] Melissa Høegh Marcher, Ingrid Maria Christensen, Paweł Grabarczyk, Therese Graversen, and Claus Brabrand. 2021. Computing Educational Activities Involving People Rather Than Things Appeal More to Women (CS1 Appeal Perspective). In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 145–156. https://doi.org/10.1145/3446871.3469761

[53] Neeraj Mathur, Sai Anirudh Karre, and Y Raghu Reddy. 2018. Usability Evaluation Framework for Mobile Apps using Code Analysis. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. 187–192.

[54] Quality Matters. 2022. *Enhancing Inclusiveness within the Quality Matters Framework*. Quality Matters. https://www.qualitymatters.org/qa-resources/resource-center/conference-presentations/enhancing-inclusiveness-within-quality (accessed Mar. 2022).

[55] Quality Matters. 2022. *Quality Matters*. https://www.qualitymatters.org/ (accessed Mar. 2022).

[56] Christopher Mendez, Zoe Steine Hanson, Alannah Oleson, Amber Horvath, Charles Hill, Claudia Hilderbrand, Anita Sarma, and Margaret Burnett. 2018. Semi-Automating (or not) a Socio-Technical Method for Socio-Technical Systems. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 23–32.

[57] Christopher Mendez, Hema Susmita Padala, Zoe Steine-Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, and Margaret Burnett. 2018. Open source barriers to entry, revisited: A sociotechnical perspective. In *Proceedings of the 40th International Conference on Software Engineering*. 1004–1015.

[58] 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.

[59] Carrie B Myers and Scott M Myers. 2008. Addressing the gender gap: A teaching and learning strategy in undergraduate science courses. *Journal of Women and Minorities in Science and Engineering* 14, 4 (2008).

[60] NCES. 2021. *Digest of Education Statistics, Table 311.15*. National Center for Education Statistics. https://nces.ed.gov/programs/digest/d20/tables/dt20_311.15.asp (accessed Mar. 2022).

[61] NCES. 2021. *Digest of Education Statistics, Table 311.33*. National Center for Education Statistics. https://nces.ed.gov/programs/digest/d20/tables/dt20_311.33.asp (accessed Mar. 2022).

[62] Jakob Nielsen. 2020. *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed Mar. 2022).

[63] Alannah 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

[64] Susmita Hema Padala, Christopher John Mendez, Luiz Felipe Dias, Igor Steinmacher, Zoe Steine Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Margaret Burnett, Marco Gerosa, and Anita Sarma. 2020. How Gender-biased Tools Shape Newcomer Experiences in OSS Projects. *IEEE Transactions on Software Engineering* (2020).

[65] Shailendra Palvia, Prageet Aeron, Parul Gupta, Diptiranjan Mahapatra, Ratri Parida, Rebecca Rosner, and Sumita Sindhi. 2018. Online education: Worldwide status, challenges, trends, and implications. , 233–241 pages.

[66] 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.

[67] Farzana Rahman. 2020. Through the Looking-Glass: Barriers, Motivations, and Desires of Non-Traditional Students Learning Programming in an Online CS1 Course. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Vol. 1. IEEE, 1–2.

[68] Armin Ronacher. 2022. *Flask 2.0.3*. https://pypi.org/project/Flask/.

[69] 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)*. ACM, Association for Computing Machinery, New York, NY, USA, 191–198. https://doi.org/10.1145/3196839.3196869

[70] Dilruba Showkat and Cindy Grimm. 2018. Identifying gender differences in information processing style, self-efficacy, and tinkering for robot tele-operation. In *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, 443–448.

[71] Christine K Sorensen and Danilo M Baylen. 2009. Learning online. *DISTANCE LEARNING EDITORS AND EDITORIAL ADVISORY BOARD* 7 (2009).

[72] Shannon T Stafiej. 2012. Exploring the Educational Experiences of Gifted Muslim Women at High School in Canada. (2012).

[73] 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 preprint arXiv:1905.02813* (2019).

[74] 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.

[75] 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.

[76] 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.

[77] Totaljobs. 2022. *The Totaljobs Gender Bias Decoder*. Totaljobs Group Ltd. https://www.totaljobs.com/insidejob/gender-bias-decoder/ (accessed Mar. 2022).

[78] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511.

[79] 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)*. ACM, Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300283

[80] 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.

[81] Matthew Wickline. 2001. *Coblis — Color Blindness Simulator*. Colblindor. https://www.color-blindness.com/coblis-color-blindness-simulator/ (accessed Mar, 2022).

[82] Mary-Ann Winkelmes, Matthew Bernacki, Jeffrey Butler, Michelle Zochowski, Jennifer Golanics, and Kathryn Harriss Weavil. 2016. A teaching intervention that increases underserved college students' success. *Peer Review* 18, 1/2 (2016), 31–36.

[83] 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.