Doing Inclusive Design: From GenderMag in the Trenches to InclusiveMag in the Research Lab

Margaret Burnett
School of EECS
Oregon State University
Corvallis, Oregon 97331 USA
burnett@oregonstate.edu

ABSTRACT

How can user interface and user experience (UI/UX) professionals assess whether their software supports diverse users? And if they find problems, how can they fix them? We begin this keynote address with a summary of GenderMag, a systematic inspection method for finding and fixing "gender inclusivity bugs"—biases against different genders in software interfaces and workflows. We then show what UI/UX professionals are doing with it in the real world, from their bias finds & fixes to their practices & pitfalls in using it. Finally, we present InclusiveMag, a meta-method that can be used by HCI researchers to generate systematic inclusiveness methods for other dimensions of diversity.

CCS CONCEPTS

• Human-centered computing → Human-Computer Interaction (HCI) → HCI design and evaluation methods

KEYWORDS

Inclusive software; gender biases; GenderMag

ACM Reference format:

Margaret Burnett. 2020. Doing Inclusive Design: From GenderMag in the Trenches to InclusiveMag in the Research Lab. In *Proceedings of International Conference on Advanced Visual Interfaces (AVI'20). ACM, New York, NY, USA, https://doi.org/10.1145/3399715.3400871*

1 Introduction

Software has repeatedly failed diverse populations, falling short of aiding their productivity or even being usable by some populations [6, 14, 16, 19, 21, 24]. Such failures are serious: they marginalize people who "don't fit"—where "don't fit" can simply mean being different from the people who wrote the software. Of the many forms of diversity for which this problem arises, its connection with gender diversity is particularly well documented [3, 4, 5, 6, 7, 8, 10, 14, 16, 19, 20, 21, 23, 24, 25, 26].

Making software products usable to people regardless of their gender has practical importance. If software teams fail to achieve inclusiveness, their market size shrinks. If a project's development tools or products fail to achieve inclusiveness, not only is product adoption reduced, but also the involvement of women and other underrepresented populations in the teams themselves [17, 19].

2 GenderMag

To enable software professionals—such as user experience researchers, software developers, designers, software project managers, content designers, and/or other professionals on software creation teams—we devised GenderMag. GenderMag (Gender-Inclusiveness Magnifier) [7] is a method for finding, and also fixing [24], gender-inclusivity "bugs" in software. Empirical research reports that GenderMag is effective at helping software practitioners find and fix such inclusivity bugs in their teams [7, 24].

2.1 GenderMag's Foundations

GenderMag helps a software team find and fix user-facing inclusivity bugs in their own products, using five "facets" of individuals' cognitive styles for going about problem solving—an individual's motivations, computer self-efficacy, attitude(s) toward risk, information processing style(s), and learning style(s). These facets form the core of the GenderMag method.

GenderMag literature defines inclusivity bugs as issues tied to one or more of these cognitive facets. Such "bugs" are cognitive inclusivity bugs, but also gender-inclusivity bugs because the facets capture well-established (statistical) gender differences in how people problem-solve [2, 3, 4, 8, 9, 11, 13, 14, 16, 18, 21, 23]. For example, using these facets, a software team might discover an inclusivity bug if a feature is easily discoverable by people with a tinkering learning style, but not easily discoverable by people with a process-oriented learning style.

In essence, the diverse problem-solving styles represented by the facets capture cognitively diverse behaviors that occur both within a given gender as well as those with statistical differences between one gender and another. Thus, supporting multiple facet values in software tends to make software better for people of all genders.

GenderMag makes the five facets concrete with a set of three faceted personas—"Abi", "Pat", and "Tim". Personas [1] are a widespread technique in industry. Each persona represents a subset of a system's target users—here, their purpose is to represent differences in the facet values. Abi's facet values represent the opposite end of the problem-solving style spectrum from Tim's, and Pat's facet values are a mixture of Abi's and Tim's. Tim's facet values are most often the ones software developers tend to design for, and Abi's facet values are often overlooked. Portions of the personas that are not about the facets (e.g., appearance, demographics, experience, job title, etc.) are customizable (Figure 1).

2.2 How GenderMag Works

GenderMag sets these faceted personas into a systematic process via a specialized Cognitive Walkthrough (CW), as follows. Evaluators "walk through" each step of carrying out a scenario, as shown in Figure 2, and answer questions about subgoals and actions a user would need to accomplish those subgoals, as follows:

SubgoalQ: 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).

ActionQ1: Will <Abi/Pat/Tim> know what to do at this step? (Yes/no/maybe, why, what facets ...).

ActionQ2: If <Abi/Pat/Tim> does the right thing, will they know they did the right thing and are making progress toward their goal? (Yes/no/maybe, why, what facets ...).

As these questions show, identifying issues using this process includes identifying the facets that are tied with each, because the facets pinpoint the kinds of cognitive styles the inclusivity bugs are excluding. One field study of four software teams' GenderMag evaluations on their own products revealed inclusivity bugs in 25% of the actions and subgoals evaluated [6]. A more recent collection of data from 17 software teams averaged 32% (Figure 3).

These facets are also key to the fixes—an inclusivity bug's fix can be designed around the facet that raised the issue. For example, to fix an issue that was raised for a particular problem-solving style, a team would revise that part of the UI to support multiple problem-solving styles: the already supported one(s) and the unsupported one(s).

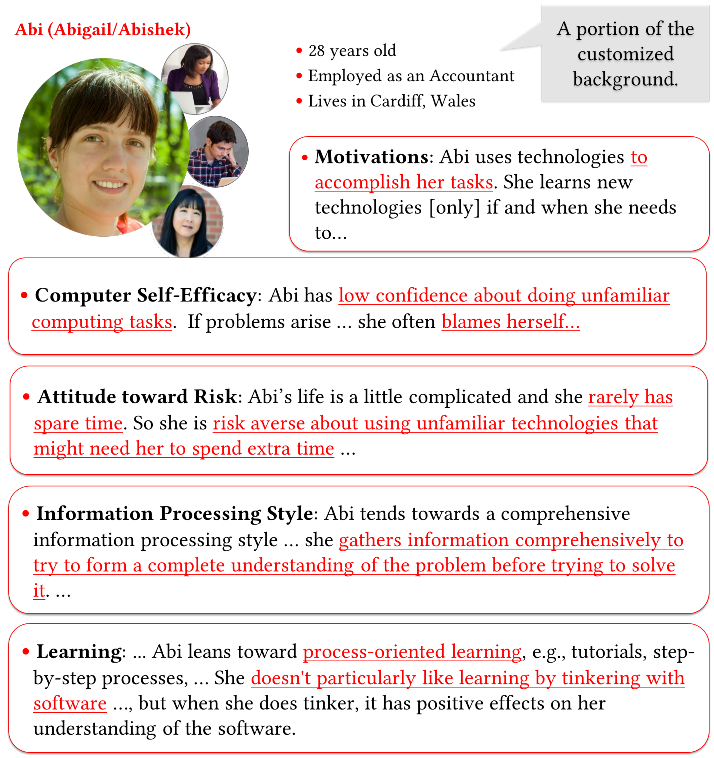

Figure 1: Key portions of the Abi persona.

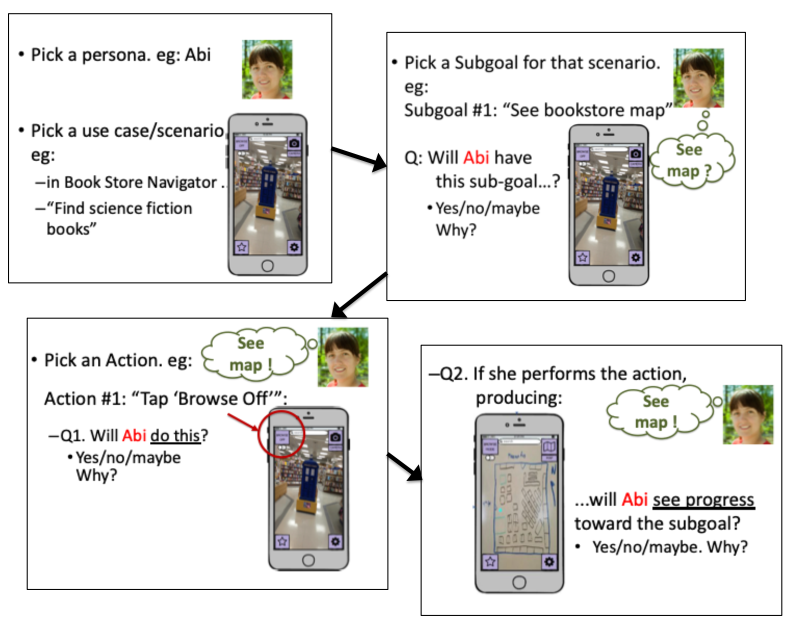

Figure 2: In the GenderMag process, evaluators walk through a scenario (e.g., "Find science fiction books") in the prototype (e.g., a navigation app for inside a bookstore) from the perspective of one of the GenderMag personas (e.g., Abi), answering the CW questions (see text) along the way.

In one lab study, when user experience researchers used GenderMag to identify usability issues, over 90% of the issues were validated by other empirical results or field observations, and 81% aligned with gender distributions of those data [7]. More generally, previous empirical studies have found GenderMag to be effective at identifying issues, and at pointing toward effective fixes [5, 7, 10, 20, 24].

2.3 "Doing" GenderMag in the Trenches

Real-world teams are ever-mindful of the practicalities of adding new methods on top of their existing processes. For example, if they add GenderMag to their processes, how can they keep the time costs viable? How can they maximize impacts of using it? To find out how software teams "in the trenches" handle these and similar questions, we collected the GenderMag-based processes of 10 real-world software teams for periods ranging from 5 months to 3.5 years.

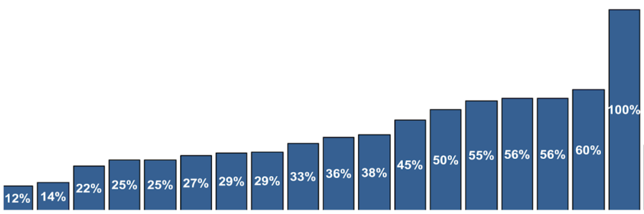

Figure 3: 17 real world teams using GenderMag on their own software. The teams found gender biases in 12% to 100% of the features, 32% on average.

	First GM session	Multi GM sessions	Follow-up mtgs	Interviews	Emails	Evidence in prior lit.
			Minimizing Costs			
1 Learning/doing	✓			✓		[5]
2 Multi-path evals	✓		✓	✓		
3 Abstracting			✓✓			
Beyond control	✓	✓				[6]
Eval'ing proxy	✓✓					
			Maximizing Benefits			
4 Abi first	✓✓✓✓✓ ✓✓✓✓✓					[6,19]
5 Speak thru Abi			✓	✓✓		
6 Calculating bias	✓✓	✓✓				
			Beyond the Session			
7 GenderMag'ing beyond products			✓✓	✓	✓✓	
8 Facet survey			✓	✓	✓✓✓	[24]
9 GenderMag Moments			✓	✓✓✓ ✓✓	✓	

Table 1: Some practices and pitfalls from our field study[15]. The checkmarks are evidence instances of each practice or pitfall.

Our results revealed a number of practices and pitfalls, some of which are documented in [15] and summarized in Table 1. We summarize a few of them next.

2.3.1 Abstracting Beyond—Carefully. A GenderMag evaluation session's outputs are concrete, in a way analogous to testing. It produces concrete outputs given the concrete inputs of a particular customized persona in a particular scenario using a particular prototype.

Despite this concreteness, two teams worked out a way to abstract beyond a session's concrete outputs. They did so by choosing for their evaluation a single instance of a UI pattern used in multiple places in their system. They then treated the single instance's evaluation as being applicable across all instances of that UI pattern.

This optimization saved the teams time and effort, by eliminating the need to evaluate each instance in its own context. However, some teams ran into pitfalls if they were not precise about exactly what they were abstracting beyond what. For example, one team tried to evaluate an upcoming system using an older prototype, because they did not have the new system on any of the machines they had brought to the evaluation session. Unfortunately, this backfired—they became so confused about what "Abi" would and would not see in the new system, that their evaluation session turned out to be a waste of time. Thus, the practice of "abstracting beyond" paid off when it was used with care, i.e., only for multiple instantiations of a single pattern, but not when systems were merely "similar."

2.3.2 Facets Drive Fixes. Five of the teams took Abi's facets one step further: they used the *facets* to engineer the *fixes* to the inclusivity bugs they found. For example, in one recent study, a team fixed the UI widget in Figure 4 to better support Abi's motivations and risk facets (recall Figure 1). They removed the counts (the right side of each bar in Figure 4) to make the filters look more like filters, so that if a task-motivated user like Abi was trying to filter, they would see that widget as the way to accomplish their task.

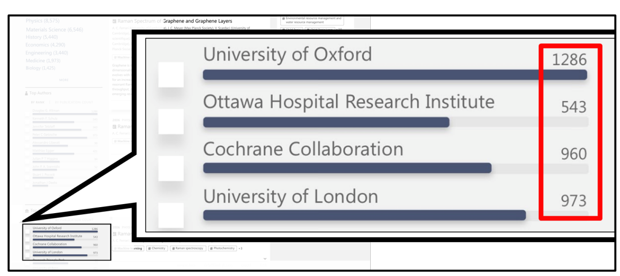

Figure 4. The filtering widget originally included a counts column (right). The team decided that task-motivated users like Abi might not see it as a filtering device, since it looks more like statistics, so they removed it [24].

2.4 Effectiveness by Facet and by Gender

GenderMag-based fixes like the above have produced promising results. For example, Figure 4 was part of an existing product that was redesigned using the "Facets Drive Fixes" practice above [24]. Their results are summarized in Table 2. (Figure 4's fix was in response to "Issues 1&2" in the table.)

	Facets	Redesigned For		Effects
Issues 1&2	Motivations	✓		+ +
	Self-Efficacy	✓		+ +
	Risk	✓		* +
	Info-Process	✓	✓	+ *
	Learning	✓	✓	+ +
Issue 3	Motivations	✓		+ −
	Risk	✓		− +
	Learning	✓		= =
Issue 4	Risk	✓	✓	+ +
	Info-Process		✓	+ +
	Learning		✓	+ +
Issues 5&6	Self-Efficacy	✓	✓	+ *
	Risk	✓		* +
	Info-Process	✓	✓	+ *
	Learning	✓		+ =

Table 2. Each row shows effects of the Facets-Drive-Fixes redesigns, for each facet value for Abi (left symbols, in orange) and Tim (right symbols, in blue) [24].
✓: redesigned for that facet value.
+: redesign better: (fewer failures than Original).

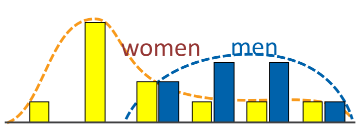

Figure 5. Y-axis: Counts of the 20 participants by their facet values [24]. Yellow: women, blue: men. X-axis: Number of Abi vs. Tim facets. Example: the left bar says that the only participant with 5 Abi facets (0 Tim facets) was a woman; the right pair of bars says that one man and one woman had 5 Tim facets (0 Abi facets).

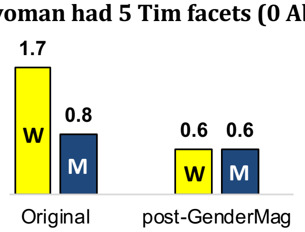

Figure 6. Average number of action failures per person by gender identification. In the Original version, women's action failure rates were over twice as high as men's; with the post-GenderMag redesign, all failure rates went down, and the gender gap disappeared [24].

Totaling up Table 2 shows that, the net result of the fixes designed according to the facets was positive for all five of the facets. Specifically, the facet tallies improved for both Abi- and Tim-style Motivations, improved for both Abi- and Tim-style Self-Efficacy, improved for Tim-style Risk (no net change for Abi-style Risk), for both Abi- and Tim-style Information Processing, and for both Abi- and Tim-style learning styles.

Turning to gender, as Figure 5 shows, the facets aligned with the participants' genders such that the Abi facet values were most prevalent in women and the Tim facet values were most prevalent in men. (Women and men were the only gender identifications present in this participant pool.) As the results of Table 2 and Figure 5 would predict, the facet improvements also resulted in gender inclusivity and equity improvements (Figure 6).

3 InclusiveMag

We recently introduced InclusiveMag [17], a meta-method that can produce inclusive design methods like GenderMag. We built InclusiveMag inductively, by generalizing upon the principles and processes used in creating GenderMag. Our inductive process is similar to one defined by Sjøberg et al. [22] on how theories (and methods) can be inductively defined from concrete practice to more generalized forms.

The InclusiveMag method allows *inclusivity researchers* to set up a systematic inclusiveness inspection method, for *software practitioners* to then apply to their own software to systematically evaluate how it supports (or doesn't) diverse populations. As Figure 7 shows, InclusiveMag has three steps—(1) Scope, (2) Derive, and (3) Apply. Inclusivity researchers perform Steps 1 and 2, and software practitioners perform Step 3.

In Step 1, inclusivity researchers scope the inclusiveness method. They select a software type, select a diversity dimension, and perform research on what might affect how populations along the diversity dimension use the software type. Step 1 results in a set of facet categories, which are relevant to both the under-served and mainstream populations, and facet values, which differ between the under-served and mainstream populations. The facets form the core of the InclusiveMag-generated method.

For example, the results of Step 1 for generating GenderMag were:

- *Software type*: problem-solving software
- *Diversity dimension:* gender

- *Facets*: Motivations, Self-Efficacy, Risk, Information Processing Style, and Learning Style.

In Step 2, inclusivity researchers use the facets produced in Step 1 to derive customizable personas and an analytic process specialized to their selected diversity dimension. Step 2 begins with projecting (flattening) the values of each facet (category) onto a linear scale for that facet: one value at each "endpoint" of each facet, and one somewhere within, to make clear that the facet values are on a continuum, not binary (yes/no) values.

For each facet, the inclusivity researchers assign to the under-served persona facet values that represent the endpoint of the under-served population, and to the mainstream persona the opposite end-point, selecting endpoints that are reasonably common among those populations, not extreme outliers.

For example, the results of Step 2 for generating GenderMag were:

- *Personas:* Abi (underserved), Tim ("mainstream," i.e., best served), Pat (different mix of facet values).

Populations considered	Diversity dimension	Software type	Facets from research
ADHD, ≠ADHD	Cognitive	Managing finances	Focus, Organization, Impulsivity, Memory, Financial responsibility
Autism kids, ≠Autism kids	Cognitive	Math learning	Comprehension ability, Ability to follow instruction, Concentration level
Dementia, ≠Dementia	Cognitive	Self-driving car	Motivations, Memory, Problem-solv. & learning ability, Self-sufficiency/independence, Attention
Diabetic retinopathy, Good vision	Vision	Chore robot	Physical/visual ability, Technology preferences, Emotional state & well-being, Financial stability & status, Social interactions
Low literacy, Med/High literacy	Education	Language learning	Confidence in using tech, Reading skills, Learning style, Motivations/frustrations with tech. Susceptibility/sensitivity to tech requiring reading
Low socio-economic status (SES), Med/high SES	Socio-economic status	University's website	Home life, School experience, Psychological health, Career aspirations
Older Adults, ≠Older Adults	Age	Email	Tech. comfortable with, Attitude toward tech, Physical difficulties
Pre-schoolers, Adults	Age	Media player	Motivations, Approach to learning, Attitude to recovery, Interaction style, Approach to tech.

Table 3. A multi-case study of using InclusiveMag in an Inclusive Design class [17], along with some information on

- *Analytic Process Basis:* Cognitive Walkthrough. (There has also been some work using Heuristic Evaluation.)

The outcome of Step 2 is a generated method built upon the facets selected in Step 1. In Step 3, a team of one or more software practitioners applies it to their software. Section 2.3 and Section 2.4 of this paper describe instances of several software teams doing Step 3—i.e., applying GenderMag to their own software.

The InclusiveMag method is new, and efforts to evaluate it are still emerging. However, early results from applying it to a variety of inclusive design activities (Table 3) in HCI education have been encouraging [17].

4. Conclusion

Businesses and research agree that making software inclusive pays off. From the practical perspective, it expands market-share and reduces customer support costs by producing fewer usability errors (e.g., recall Figure 6). From a software quality perspective, inclusivity methods like GenderMag provide feasible and measurable forms of quality assurance (recall Section 2.3). But most important, methods like these matter to society, by removing inclusivity bugs from technology that inequitably erect barriers from diverse populations who deserve fair and equitable treatment by technology itself.

ACKNOWLEDGMENTS

We are grateful to the many collaborators who have contributed to this work. Especially notable were the leadership by Claudia Hilderbrand of the work reported in Section 2.3, by Mihaela Vorvoreanu and Claudia Hilderbrand for the work reported in Section 2.4, and by Christopher Mendez for the work reported in Section 3. This work has been supported in part by the National Science Foundation under Grants 1528061 and 1901031.

REFERENCES

[1] Tamara Adlin and John Pruitt. 2010. *The Essential Persona Lifecycle: Your Guide to Building and Using Personas.* Morgan Kaufmann/Elsevier, San Francisco, CA.

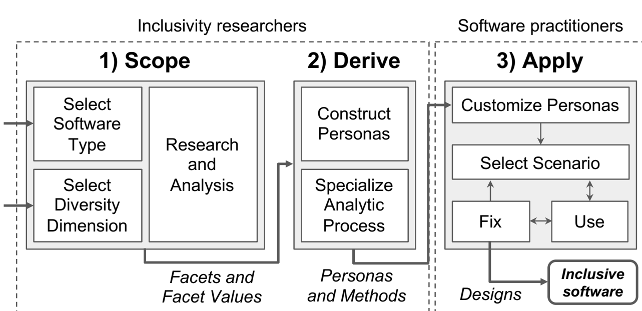

Figure 7. The InclusiveMag process [17] has three steps, each of which has multiple components. Inclusivity researchers perform Steps 1 and 2, and software practitioners perform Step 3.

[2] 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* 7, 2 (August 2012), 106-120.

[3] Laura Beckwith, Cory Kissinger, Margaret Burnett, Susan Wiedenbeck, Joseph Lawrance, Alan Blackwell, and Curtis Cook. 2006. Tinkering and gender in end user programmers' debugging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, New York, NY, 231-240.

[4] Margaret M. Burnett, Laura Beckwith, Susan Wiedenbeck, Scott D. Fleming, Jill Cao, Thomas H. Park, Valentina Grigoreanu and Kyle Rector. 2011. Gender pluralism in problem-solving software. *Interacting with Computers* 23, 5 (Sept. 2011), 450–460.

[5] Margaret Burnett, Robin Counts, Ronette Lawrence, and Hannah Hanson. Gender HCI and Microsoft: Highlights from a longitudinal study. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).* IEEE, 139-143.

[6] 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 (CHI '16)*. ACM Press, New York, NY, 2586-2598.

[7] 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, 6-19 (Nov. 2016), 760-787.

[8] 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*. ACM Press, New York, NY, 674-686.

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

[10] Sally Jo Cunningham, Annika Hinze, and David M. Nichols. 2016. Supporting gender-neutral digital library design: A case study using the GenderMag toolkit. In *International Conference on Asian Digital Libraries*. Springer, 45-50.

[11] Thomas Dohmen, Armin Falk, David Huffman, Uwe Sunde, Jürgen Schupp, and Gert G. Wagner. 2011. Individual risk attitudes: Measurement, determinants, and behavioral consequences. *Journal of the European Economic Association* 9, 3 (June 2011), 522–550.

[12] Denae Ford, Justin Smith, Philip J. Guo, and Chris Parnin. 2016. Paradise unplugged: Identifying barriers for female participation on stack overflow. In *Proceedings of theACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016). ACM, New York, NY, USA, 846-857.*

[13] Catarina Gralha, Miguel Goulao, and Joao Araujo. 2019. Analysing gender differences in building social goal models: a quasi-experiment. *IEEE Intl. Requirements Engineering Conference* (12 pages).

[14] 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 (July 2014), 137-149.

[15] Claudia Hilderbrand, Christopher Perdriau, Lara Letaw, Jill Emard, Zoe Steine-Hanson, Margaret Burnett, Anita Sarma. 2020. Engineering gender-inclusivity into software: Ten teams' tales from the trenches. *ACM/IEEE International Conference on Software Engineering.* 11 pages.

[16] Caitlin Kelleher. 2009. Barriers to programming engagement. *Advances in Gender and Education* 1, 1, 5-10.

[17] Christopher Mendez, Lara Letaw, Margaret Burnett, Simone Stumpf, Anita Sarma, Claudia Hilderbrand. 2019. From GenderMag to InclusiveMag: An inclusive design meta-method. *IEEE Symposium on Visual Languages and Human-Centric Computing*, 97-106.

[18] Joan Meyers-Levy and Barbara Loken. 2015. Revisiting gender differences: What we know and what lies ahead. *Journal of Consumer Psychology* 25, 1 (Jan. 2015), 129-149.

[19] Hema Susmita Padala, Christopher Mendez, Luiz Felipe Dias, Igor Steinmacher, Zoe Steine Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Margaret Burnett, Marco Gerosa, Anita Sarma, 2020. How gender-biased tools shape newcomer experiences in OSS projects. *IEEE Transactions on Software Engineering.* (19 pages). DOI: https://doi.org/10.1109/TSE.2020.2984173

[20] Arun Shekhar and Nicola Marsden. 2018. Cognitive Walkthrough of a learning management system with gendered personas. *4th Gender & IT Conference (GenderIT'18)*, 191-198. DOI: https://doi.org/10.1145/3196839.3196869

[21] Anil Singh, Vikram Bhadauria, Anurag Jain, and Anil Gurung. 2013. Role of gender, self-efficacy, anxiety and testing formats in learning spreadsheets *Computers in Human Behavior* 29, 3 (May 2013), 739–726.

[22] Dag Sjøberg, Tore Dybå, Bente Anda, and Jo E. Hannay. 2008. Building Theories in Software Engineering. In: F. Shull, J. Singer, D. Sjøberg (eds) *Guide to Advanced Empirical Software Engineering*. Springer, 312-336.

[23] 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.

[24] 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 ACM Conference on Human Factors in Computing Systems (CHI '19)*. ACM Press, New York, NY.

[25] Zhendong Wang, Yi Wang, and David Redmiles. 2018. Competence-confidence gap: a threat to female developers' contribution on Github. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. ACM Press, New York, NY, 81-90. DOI: https://doi.org/10.1145/3183428.3183437

[26] Gayna Williams. 2014. Are you sure your software is gender neutral? *Interactions* 21, 1 (January 2014), 36–39.