

Small Educational Steps Towards Improving the Status of Women in Software Engineering

Pankaj Kamthan
Computer Science and Software Engineering
Concordia University
Montreal, Canada
pankaj.kamthan@concordia.ca

Nazlie Shahmir
Information Services
Canadian Pacific Railway Company
Calgary, Canada
nazlie_shahmir@cpr.ca

Abstract—The women in software engineering continue to face a culture of discord that manifests itself in the form of underrepresentation, unpleasantness, and/or inequitableness. This somewhat dire situation was only exacerbated during the COVID-19 pandemic when the women in software engineering education and profession had to deal with multiple ‘crisis’. The status quo is clearly unacceptable, not least because of pervasiveness of software in society. In that regard, relying on a multipronged approach grounded in a body of knowledge, ethicality, and history, this paper proposes certain basic steps in software engineering courses and projects that could be put into practice for improving “gender literacy” among students. These educational steps are illustrated by anecdotes and examples.

Keywords—accessibility; community smell; DEI; history; persuasion; SECEPP; social activism; SWEBOK

I. INTRODUCTION

It is commonly understood that, to a large extent, the engineering of software is inherently a social endeavor. Then, creating an amicable and amenable social environment that maximizes the potential of all stakeholders involved, regardless of their gender, is therefore natural. However, when it comes to inconsideration, the computing community, in general, and the software engineering community, in particular, has a long disreputation of singling-out one gender, namely women [1–3]. For the rest of the paper, SoWSE is used as an all-encompassing term to denote the *current status of women in relation to software engineering*.

This paper employs a multipronged *preventative* approach to addressing SoWSE *early*, that is, during the phase in the lives of men and women when they are students and therefore still learning and growing, in the hope that attitudes and/or behaviors developed and shaped early during their software engineering education (SEE) will carry over later during their software engineering profession. The steps involved are based on authors’ own educational and professional experiences of more than 25 years, derived from “first principles”, aligned with previous work, and require minor (re)orientation in the current direction of SEE. These relatively smaller steps can serve as a basis as well as a prerequisite for a commitment to relatively larger steps. The issues pertaining to non-binary genders, albeit important, are beyond the scope of this paper.

The rest of the paper is organized as follows. In Section II, necessary background is provided and related work is discussed in order to set the stage of the underlying problem.

The collection of steps that could be pursued by teachers and students are explored at some depth in Section III. In Section IV, potential directions for future research are outlined. Finally, in Section V, concluding remarks are given.

II. BACKGROUND AND RELATED WORK

A. A Glimpse into the Nature of a Perennial Social Problem and Proposed Solutions

1) An Epigrammatic Synthesis of the Problem

The women in software engineering (WSE) have been under siege for the past several decades, spanning 20th and 21st centuries. The social issues continue to be faced by WSE in educational and/or professional settings include the following: *dismissiveness* (for example, comments or questions not taken seriously) [4], *empowerment imbalance* (for example, low recognition or low reward) [4], *harassment* (for example, name-calling on social media) [5], *inadequate support* (for example, lack of growth opportunities or inflexibility during the COVID-19 pandemic) [6–8], *lack of autonomous decision-making* (for example, mandated supervision on significant matters) [7], *lack of peer parity* (for example, absence of peers, mentors, or role models) [4], *marginalization* (for example, not invited or discouraged to participate in software project team meetings) [7], *maternal discrimination* (for example, overtly or covertly discouraged to continue after maternity leave) [7], *microaggressions* (for example, exoticization of women of ethnic minority during interactions), *ostracization and isolation* (for example, by being the only women in a software project team, sometimes repeatedly) [4, 9], *salary imbalance* (for example, compared to men in the same or similar positions) [7], *scapegoating* (for example, ridiculed and blamed, not always with due diligence, for shortcomings) [7, 9], *stereotyping* (for example, told implicitly or explicitly that men are better suited) [4], and *workload imbalance* (for example, obligated to overwork) [6–8].

These issues can have a long-term impact on mental and/or physical health of those women who are *compelled* to adapt and need to resort to *coping and mitigation strategies* in order to continue to be involved in some capacity in software engineering education or profession [9], and even of those who do not. They also hardly engender a natural environment for essentially any type of work for women, let alone a productive environment for engineering of software, not to mention that resilience and compromise have their inevitable limits.

The recent large-scale movements initiated, mobilized, and sustained by women, such as #metoo and #mahsaamini, only reinforce that the breadth and depth of the problem is not specific to software engineering, or even computing.

2) An Epigrammatic Synthesis of the Solutions

In the past few decades, multiple solutions have been proposed, at different times, examining SoWSE in different geographical regions, at different breadths and depths, from different perspectives, and published in different avenues. For example, motivational (but, sometimes, opportunistic and self-serving) speeches to girls and women have been given [4]; prevalence of a “brogrammer culture” has been raised [4, 10]; labels such as “(conscious and subconscious) gender bias”, “gender disparity”, “gender divide”, “gender inequality”, and “gender” inequity” have been created [11]; recurring pseudo-scientific claims about the “predispositions” and “differences” between men and women have been discredited [12]; myths and misconceptions of “gender-neutrality” of software have been debunked and clarified [13]; surveys to determine root causes of the plight of WSE have been conducted [11]; calls for action have been made [11]; proposals have been put forth [11]; slogans have been proselytized [11]; and books have been written [14]. Yet, not much has changed over the generations and the struggle of WSE continues unabated today.

It is possible that the problem may be extremely difficult, even “wicked”, and a lasting solution elusive and insusceptible to eventalism (say, in form of “big bang”). However, it at the very least does show that there is a lack of sustained interest, perhaps even a lack of empathy, at *scale*, by those in positions of influence or privilege in changing the status quo for the better. This is unsurprising, as a change would require admitting of their own volition that there indeed is a problem and they are part of the problem [11], overcoming the *bystander effect*, and need to accept responsibility and bear a cost to self to be a part of its solution, a tall order for many.

B. Implications for Interplay between Software and Society

1) Unacceptable Status Quo of Women in Software Engineering

The consequences are solemnly evident: women are part of a culture in which they are not naturally welcome in software engineering, which *implies* and *is implied by*, there are a disproportionately low number of WSE (meaning, there are women who either do not enter software engineering or do not stay in software engineering), and there is a significant number of WSE who, *ceteris paribus*, are not treated the same as men in software engineering and treated as “second-class”. Fig. 1 iconizes this continuing *self-fulfilling prophecy*.

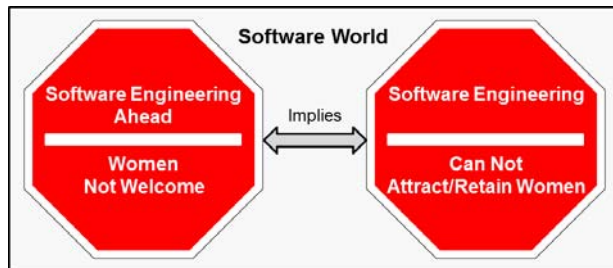


Figure 1. An icon symbolizing SoWSE.

This is extremely disconcerting also to women in other disciplines who rely intimately and regularly on the use of software, and are therefore affected, directly or indirectly, by how software is engineered. It patently situates software engineering in a uniquely precarious and potentially disreputable position among all the engineering disciplines.

2) Unacceptable Status Quo of Software Engineering Products and Services for Women

The aforementioned consequences inevitably also have a lasting ripple effect on both open-source and closed-source software engineering, and manifests itself in different ways: *rejection* (a large segment of society is disregarded in educating prospective software engineers or employing qualified software engineers), *suppression* (women with sustained interested in software engineering do not have the opportunities to reach their full potential), *elimination* (the retention rate of women is threatened unnecessarily), and *exclusion* (the needs and preferences of a large segment of potential users are not taken into account in developing and/or maintaining socio-technical interactive systems). In other words, women are essentially not welcome either as producers or as consumers of software. The message, implicitly or explicitly, seems to be that women and software should exist and persist in two completely separate worlds, as implied by Fig. 1, where one never comes across the other, an evident implausibility in today’s society, to say the least.

The systems resulting from such ‘engineering’ (as a euphemism for ‘persecution’) cannot be considered as truly ‘representative’. This, in turn, means that such systems are, *by design*, not “gender-neutral”, and any assertions of such systems being utilizable and usable by women are speculative. In addition, such non-representativeness does not get any better if the designers of the aforementioned systems are “asshole designers” [5, 15] creating “asshole design” (defined as “design that possesses clearly coercive, deceptive, and/or malicious intent on the part of its creator, rather than be inconvenient to a user due to a poor design decision or implementation” [15]).

The indelible loss of those quality attributes deemed necessary for socio-technical interactive systems is not only a matter of grave concern in and of itself, but even more so given the increasingly indispensable role being played by software in society-at-large. Indeed, such reliance-by-necessity on software was only accentuated and reaffirmed repeatedly during the mandated lockdowns at the peak of the COVID-19 pandemic.

C. SEE and SoWSE

SEE has a long and illustrious history [16, 17], and undergraduate and/or graduate software engineering programs continue to proliferate globally. However, the issues pertaining to gender have to date been discussed rarely and peripherally [18], which is one of the motivations for this paper.

III. A COMPENDIUM OF SMALL STEPS IN SOFTWARE ENGINEERING EDUCATION AS PART OF THE SOLUTION

A. Improving SoWSE and Software Engineering Courses

There are a number of concrete steps that can be taken, both inside and outside the classroom, towards improving SoWSE.

From an *epistemological perspective*, these steps advocate ethically-sensitive and historically-guided SEE, apply to both undergraduate and graduate students, represent iterative and incremental rather than radical changes in attitudes and/or behaviors on part of teachers and students, ‘nudging’ the students slowly but certainly purposively by means of positive as well as negative reinforcements towards desired outcomes [19]. (These steps are part of a marathon, not a sprint.)

From a *logistical perspective*, these steps are inexpensive in terms of time and effort, and do not require acquisition or management of extra resources by the institutions. Indeed, as seen later, they could be integrated naturally in a variety of software engineering-related courses.

1) *Improving SoWSE and Software Engineering Outreach*

The Web Sites of courses, such as those using *Moodle* or another learning management system, or even otherwise, could be used as venues for creating awareness among students of events, movements, and organizations that (do not subscribe to #genderwashing but) are aimed towards accentuating and celebrating the accomplishments of women in computing, in general, and software engineering, in particular. Fig. 2 shows such an example.



Figure 2. A montage of initiatives committed to congregating, recognizing, and supporting WSE.

Furthermore, the teachers could, for example, invite guest women speakers on the *International Women in Engineering Day* or on other auspicious occasions, and the students, both men and women, could be encouraged (and, if possible, even credited) to participate, volunteer, or work, as suitable, for these outlets. (The authors, incidentally, have been involved with *Chic Geek* and are signatories of the *Diversity Charter*.) Indeed, the broad availability both inside and outside the classroom of videoconferencing tools, such as *Microsoft Teams* and *Zoom*, has to a large extent overcome the logistical limitations due to time and place, and made such speaker arrangement and participation ever so easier.

2) *Improving SoWSE and Software Engineering Ethics*

The *Guide to the Software Engineering Body of Knowledge* (SWEBOK) represents the current state of generally-accepted consensus-based knowledge emanating from interplay between theory and practice of software engineering [20]. The SWEBOK is decomposed into a set of topics through *Knowledge Areas* (KAs). The *Software Engineering*

Professional Practice KA of SWEBOK is especially relevant to this paper as it mentions codes of ethics.

The *ACM/IEEE Software Engineering Code of Ethics and Professional Practice* (SECEPP) lists generic principles, each of which is refined into specific clauses, related to the behavior of and decisions made by professional software engineers as well as students of the profession.

In particular, the following clauses from Principle 5: MANAGEMENT, Principle 7: COLLEAGUES, and Principle 8: SELF, respectively, are applicable:

5.07. Offer fair and just remuneration.

5.08. Not unjustly prevent someone from taking a position for which that person is suitably qualified.

7.04. Review the work of others in an objective, candid, and properly-documented way.

7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.

8.07 Not give unfair treatment to anyone because of any irrelevant prejudices.

In teaching a course involving any of the aforementioned SECEPP clauses, or even otherwise, the students could be introduced to at least those *fallacies* and *biases* that can impede rational thinking and are team dynamic-specific or gender-specific. For example, these could include, but are not limited to, the following: *Argumentum Ad Hominem*, *Appeal to Authority*, *Appeal to Flattery*, and *Cherry Picking* (logical fallacies), the *Bias Blind Spot* and the *Dunning-Kruger Effect* (meta-cognitive biases), and the *Gender Stereotyping Bias*, the *Confirmation Bias*, the *Affinity Bias*, and the *Inter-Group Bias* (cognitive biases). The students could, time permitted, also be introduced to refutation and debiasing techniques.

3) *Improving SoWSE and Software Engineering History*

There have been several notable contributions by women to computing since the dawn of computing [21, 22, 23]. Indeed, the indispensable role of women in computing, in general, and cost estimation, in particular, was evident even more than 75 years ago [1]:

Sometime in 1944, computers became “girls.” The University of Pennsylvania hired “girl computers”; Warren Weaver started calling Applied Mathematics Panel computers “girls”; Oswald Veblen, who had once led a team of computing men, used the term “girls”; George Stibitz began ranking calculating projects in “girl-years” of effort.

Furthermore, the term ‘software engineering’ was coined in the early 1960s by Margaret Hamilton. However, the ‘standard’ software engineering textbooks seldom address the human or social aspects of the history of software engineering at any depth, and rarely highlight the role of women per se in advancing software engineering to its current state.

In teaching a course based on any of the SWEBOK KAs, or even otherwise, the students could be introduced to the *genealogy* of that KA, moving both backward and forward in time, where contributions of women alongside men (and the challenges that they overcame) could be highlighted. For example, Table 1 shows areas of contributions of certain women in one KA from SWEBOK, Version 3 and another KA

from SWEBOK, Version 4. (The presence of three entries in each case is intentional and follows the ‘Rule of Three’ criterion of a (software) pattern.)

TABLE I. EXAMPLES OF WOMEN CONTRIBUTORS TO SWEBOK KAS

SWEBOK KA	Notable Women Contributors
Software Architecture	Patricia Lago, Barbara Paech, Mary Shaw
Software Design	Nancy Leveson, Perdita Stevens, Rebecca Wirfs-Brock

Indeed, such efforts need not be restricted to SWEBOK KAs, and could extend to cognate disciplines. For example, Table 2 shows areas of contributions of certain men and women in human-computer interaction (HCI).

TABLE II. EXAMPLES OF MEN AND WOMEN CONTRIBUTORS TO HCI

HCI Area	Notable Men and Women Contributors
Interaction Design	Helen Sharp, Ben Shneiderman, Jennifer Tidwell
Mental Models	Felienne Hermans, Donald Norman, Indi Young

The publications, recordings of speeches, or other work by women software engineers could serve, for example, as topics for classroom discussions and/or presentations, as reading material for term papers, and as a basis for essay-type, open-ended questions on, say, assignments and/or examinations.

B. Improving SoWSE and Software Engineering Projects

It is common for software engineering-related courses to have semester-long team-based projects as a major component of the overall assessment. These projects tend to have several goals, including developing interpersonal skills for working in a team environment in the software industry and beyond [24].

1) Improving SoWSE and Software Project Topics

The teachers could give software project topics that are appealing to *both* men and women students. For example, this could be ascertained by conducting an informal poll at the beginning of the course. Indeed, there are several systems today, including *Google Forms* and *Moodle*, which provide effective and efficient means to conduct such polls.

2) Improving SoWSE and Software Project Teams

There are at least two common ways in which teams for projects are formed, each with their own advantages and disadvantages [24]: (a) either, based on some criteria, the teachers place the students in teams, or (b) the students could be asked to form their own teams. In either case, every member in each team needs to play a *role*, such as a team leader, modeler, designer, programmer, tester, documenter, and so on.

The teachers could ensure that each role is considered equally important and receives equal credit, and that the roles assigned within each team rotate during the semester (to avoid compartmentalization and for each member to gain experience in each role, and to avoid stereotyping of any particular gender to any particular role). Furthermore, to reduce the effect of the so-called *Imposter Syndrome* [25], women could be encouraged to “experiment”, meaning to try and make mistakes *early* and, in some cases, *deliberately* in the process [26], so that they not only learn to make less of the same or similar types of mistakes later, but also develop their confidence and competence in their adopted role over time. Indeed, doing so is

among the characteristics of agile methodologies, design thinking, and prototyping. Finally, each member could be asked to create and maintain throughout the duration of the project a private *learning journal* chronicling, for example, intrapersonal and interpersonal positive and/or negative experiences, challenges encountered along the way and means used to overcome them, and signs of progress made. The contents of such a journal could be shared selectively with the others later if deemed useful for reflection and retrospective.

3) Improving SoWSE and Software Process

The students could be asked to adopt a software development methodology that has a low probability of the presence of community smells (such as the *Organizational Silo Effect* or *Radio Silence*) [27] and, therefore, accrual of the so-called *Social Debt* [27]. This is because, according to the *Conway’s Law*, the type of the software development methodology can determine the communication structure of a software project team, irrespective of gender. In this regard, human-centered agile methodologies have been relatively more successful than machine-centered rigid methodologies, due to their inherent support for communication and collaboration in carrying out stated activities and creating corresponding artifacts. The selected agile methodology could be customized to be ethically-sensitive and gender-inclusive [28], using the SECEPP, the ISO/IEC/IEEE 24748-7000:2022 Standard, or even otherwise.

The software development process adopted could include practices, such as collective brainstorming and mind mapping [29, 30], which necessitate interaction among all team members, including women, to learn about and learn from each other. Furthermore, the software project teams could be asked to use a *Social Wiki* for communication and collaboration [31]. The postings on the Wiki could be subject to an enforceable *Communications Protocol* that includes policies for reportable conduct related to civility and etiquette, among other things, and could be monitored by teaching assistants (TAs). (The argument generalizes to other social media technologies and tools, and more generally to the use of the Social Web.)

4) Improving SoWSE and Software Product

In recent years, there has been an increasing awareness of having *diversity, equity, and inclusivity* (DEI) [32]. By embracing DEI and being more supportive of women, the teachers can show that they are “moving with the times”, they “do agile” but also “be agile”, and that “women software engineers matter”, thereby making courses more appealing to women and contribute to better controlling attrition rate.

For example, as part of software requirements elicitation for socio-technical interactive systems, the interviews could include women interviewees and user models could include *empathy map*-based women personas [33]. (The argument extends to other dimensions of *intersectionality*, such as women of ethnic minority and women with mobile disabilities.) For the development of such systems, DEI could be among non-functional (specifically, quality) requirements [13], to ensure its consideration during subsequent stages of software development. Finally, women could be encouraged to get involved in deciding the acceptance criteria, envisaging interaction design, and/or conducting user acceptance testing.

C. Other Steps for Improving SoWSE

1) Improving SoWSE and Maternity

It is not uncommon these days to come across returning students, especially graduate students but even undergraduate students pursuing, say, another degree, having their own family already or planning to start their own family.

The teachers could be sensitive towards students who are pregnant or have very young children that require constant care, and proactively reach out and find ways to accommodate them, as necessary, while still being fair to the other students, even when such accommodation may not be mandated by their institution. (This is not all that different from accommodating those students who have identified themselves as having one or more disabilities. The argument extends to paternity.)

2) Improving SoWSE and Teaching Assistants

For courses with large class sizes, it is common to have multiple TAs for support. The TAs are expected to be in touch with the students in unique ways, serve as candidates for role models, even mentors, and can play an important role in creating a congenial atmosphere in which the software project teams can operate and the software projects can evolve.

The teachers could support women students by encouraging them to apply for the positions of TAs, and by ensuring that they acknowledge the contributions of these TAs, giving them due credit, as appropriate, while avoiding *benevolent sexism*, say, by appearing unnecessarily charitable. For example, this could be done by at least having a transparent criteria for recommending students for the positions of TAs before the beginning of the course, and sending a personalized ‘Thank You’ note for their help after the end of the course. The teachers could also guide, as necessary, men TAs to work harmoniously with women TAs. For example, this could be done by making expectations regarding gender inclusivity explicit and developing *A Guide for Teaching Assistants* that the TAs need to abide by for continued employment.

D. Limitations

It is implicit throughout the aforementioned steps that the teachers themselves are prepared to deal with the issues pertaining to gender, in general, and women, in particular. In certain institutions, there are seminars and workshops related to DEI, but discussion of these is beyond the scope of this paper.

IV. DIRECTIONS FOR FUTURE RESEARCH

There are a number of directions that emanate naturally from this paper.

A. Looking Forward to Change: COVID-19 and SoWSE

The COVID-19 pandemic has been an unprecedented, life-altering event, especially for those who became seriously ill or suffer from its long-term effects, or lost loved ones. It also has led people to self-assess and question their ingrained beliefs and values, and consequences of their actions. In that regard, it would be useful to conduct a survey of students aimed to compare whether there has been any notable change in their attitudes and/or behaviors as per SoWSE before and after the COVID-19 pandemic, and is therefore of research interest.

B. Looking Forward to Change: Men and SoWSE

It is unsurprising that men have an important role to play in changing the status quo of SoWSE to an extent that could be considered acceptable by women. In that regard, it would be useful to investigate whether there has been any notable change among men in their attitudes and/or behaviors as per SoWSE before and after having been introduced to the steps proposed in this paper, their *rationale* for a change, the things they had to *do* as well as *not do* in making a change, and personal cost for making a change, and is therefore also of research interest.

V. CONCLUSION

I am no longer accepting the things I cannot change.

I am changing the things I cannot accept.

— Angela Davis

In the last 50 years or so, software engineering as a discipline has progressed far less socially than it has industrially and technically. This is not only unsustainable, it is antithetical to the essence of software engineering, and, by reference, to the spirit of software as a means for societal good.

The status quo of WSE is not only unnecessarily constraining and profoundly inegalitarian, the society has no compelling reason or the right to deprive women of the delights (and disappointments) and triumphs (and tribulations) of a software engineering experience. The WSE (or women in any other discipline of interest or matter of relevance) need to be treated as “first-class” in a society that vies to be progressive and prosperous. To do that at scale, as with dealing with the COVID-19 pandemic, “it will take all of us” as opposed to “everyone for themselves”. If not, then for many women software engineering has regressed to their detriment.

The engineering of software evidently involves being “polyliterate” not only in a combination of multiple technical dimensions, but also multiple non-technical dimensions. As the history and evolution of software engineering has shown, both the *number* of dimensions and the *nature* of each dimension are a function of time, and can therefore change over time. In particular, there is ever increasing attention on non-technical dimensions, especially *soft skills* [34]. SEE needs to evolve accordingly [35]. In that regard, it is the thesis of this paper that a software engineering-specific “gender literacy” (defined as “an awareness and understanding of those issues related to gender that affect the context and possibilities of women’s lives”) [36] should be considered as a soft skill and become an integral, and, over time, even natural, part of SEE.

It is possible to make modest but practical changes in SEE for improving SoWSE, as this paper has attempted to show. The proposed steps should come across as commonsensical to those teachers who understand SEE and are serious about contributing to making positive changes to SEE, and perhaps insightful to those students who are committed to improving the situation in which some of their peers continually find themselves in, and, in being so, are willing to make necessary changes to their own attitudes and/or behaviors, even if it comes at a personal cost. These steps are by no means exhaustive, and are only a beginning not the end, but then “a journey of a thousand miles begins with a single step”.

ACKNOWLEDGMENT

This paper is dedicated to all those women who, against all odds, continue to strive and thrive in software engineering education and profession today, hoping for a better tomorrow, and to all those men who, all the while enduring cost to self, unequivocally and unconditionally support these women in their pursuit. The authors are grateful to their women colleagues and students for sharing their experiences and to CUPFA for a Professional Development Grant.

REFERENCES

- [1] D. A. Grier, *When Computers Were Human*. Princeton University Press, 2005.
- [2] N. Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. The MIT Press, 2010.
- [3] M. Hicks, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing*. The MIT Press, 2017.
- [4] K. Kohl and R. Prikladnicki, Benefits and Difficulties of Gender Diversity on Software Development Teams: A Qualitative Study. The Thirty Sixth Brazilian Symposium on Software Engineering (SBES 2022), Virtual Event, Brazil, October 5-7, 2022.
- [5] S. Brooke, “Condescending, Rude, Assholes”: Framing Gender and Hostility on Stack Overflow. The Third Workshop on Abusive Language Online (ALW 2019), Florence, Italy, August 1-2, 2019.
- [6] L. S. Machado, C. M. Caldeira, M. G. Perin, and C. R. B. de Souza, Gendered Experiences of Software Engineers during the COVID-19 Crisis. *IEEE Software*, 38(2): 38-44, 2021.
- [7] B. Trinkenreich, R. Britto, M. A. Gerosa, and I. Steinmacher, An Empirical Investigation on the Challenges Faced by Women in the Software Industry: A Case Study. The Forty Forth International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS 2022), Pittsburgh, USA, May 22-24, 2022.
- [8] E. D. Canedo, A. T. S. Calazans, G. R. S. Silva, and E. T. S. Masson, ICT Practitioners’ Perception of Working from Home During the Covid-19 Pandemic: Exploring Gender Differences. The Thirty Sixth Brazilian Symposium on Software Engineering (SBES 2022), Virtual Event, Brazil, October 5-7, 2022.
- [9] B. Trinkenreich, I. Wiese, A. Sarma, M. Gerosa, and I. Steinmacher, Women’s Participation in Open Source Software: A Survey of the Literature. *ACM Transactions on Software Engineering and Methodology*, 31(4): 81:1-81:37, 2022.
- [10] E. Winter, L. Thomas, and L. Blair, ‘It’s a Bit Weird, but it’s OK’? How Female Computer Science Students Navigate Being a Minority. The 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2021), Virtual Event, Germany, June 26-July 1, 2021.
- [11] M. M. Moro, Lack of Diversity: Are You Part of the Problem or its Solution? The 2022 Simpósio Brasileiro de Educação em Computação (EduComp 2022), Feira de Santana, Brazil, April 24-29, 2022.
- [12] C. Fine, *Delusions of Gender: How Our Minds, Society, and Neurosexism Create Difference*. W. W. Norton and Company, 2010.
- [13] I. Nunes, A. Moreira, and J. Araujo, GIRE: Gender-Inclusive Requirements Engineering. *Data and Knowledge Engineering*, 143:1-19, 2023.
- [14] K. Pot’Vin-Gorman, *Crushing the IT Gender Bias: Thriving as a Woman in Technology*. Apress, 2019.
- [15] C. M. Gray, S. S. Chivukula, and A. Lee, What Kind of Work Do “Asshole Designers” Create? Describing Properties of Ethical Concern on Reddit. The 2020 ACM Conference on Designing Interactive Systems (DIS 2020), Eindhoven, Netherlands, July 6-10, 2020.
- [16] N. R. Mead, Software Engineering Education: How Far We’ve Come and How Far We Have To Go. *The Journal of Systems and Software*, 82(4): 571-575, 2009.
- [17] J. Maguire, S. Draper, and Q. Cutts, What Do We Do When We Teach Software Engineering? The First UK and Ireland Computing Education Research Conference (UKICER 2019), Canterbury, UK, September 5-6, 2019.
- [18] O. Hazzan, T. Lapidot, and N. Ragonis, *Guide to Teaching Computer Science: An Activity-Based Approach*, Third Edition. Springer-Verlag, 2020.
- [19] R. H. Thaler and C. R. Sunstein, *Nudge: Improving Decisions About Health, Wealth, and Happiness*. Yale University Press, 2008.
- [20] P. Kamthan and H. Washizaki, SWEBOK Matters: Report and Reflection of a SEKE Panel on the Educational and Professional Implications of SWEBOK. *International Journal of Software Engineering and Knowledge Engineering*, 32(11/12):1771-1781, 2022.
- [21] C. L. Evans, *Broad Band: The Untold Story of the Women Who Made the Internet*. Portfolio/Penguin, 2018.
- [22] C. Frieze and J. L. Quesenberry, *Cracking the Digital Ceiling: Women in Computing around the World*. Cambridge University Press, 2020.
- [23] K. W. Tracy, *Software: A Technical History*. Association for Computing Machinery, 2021.
- [24] P. Kamthan and N. Shahmir, On Software Projects in Academia and Industry from a Perspective of Software Engineering Education. The Sixth Annual Conference on Computational Science and Computational Intelligence (CSCI 2019), Las Vegas, USA, December 5-7, 2019.
- [25] L. Heels and M. Devlin, Investigating the Role Choice of Female Students in a Software Engineering Team Project. The Third Conference on Computing Education Practice (CEP 2019), Durham, UK, January 9, 2019.
- [26] H. Petroski, *To Engineer Is Human: The Role of Failure in Successful Design*. Vintage Books, 1992.
- [27] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, and F. Ferrucci, Gender Diversity and Community Smells: Insights from the Trenches. *IEEE Software*, 37(1): 10-16, 2020.
- [28] P. Kamthan and N. Shahmir, On Ethically-Sensitive User Story Engineering. The Fourth International Conference on Computer Science and Software Engineering (CSSE 2021), Singapore, Singapore, October 22-24, 2021.
- [29] P. Kamthan, Towards Understanding the Use of Wiki in Agile Software Development. The Seventh International Workshop on Information Technology for Innovative Services (ITIS 2014), Victoria, Canada, May 13-16, 2014.
- [30] P. Kamthan and N. Shahmir, On the Educational and Professional Implications of Integrating Mind Mapping in Software Testing. The Ninth Annual Conference on Computational Science and Computational Intelligence (CSCI 2022), Las Vegas, USA, December 14-16, 2022.
- [31] P. Kamthan, On the Implications of Human-Paper Interaction for Software Requirements Engineering Education. The Thirty Fourth International Conference on Software Engineering and Knowledge Engineering (SEKE 2022), Pittsburgh, USA, July 1-10, 2022.
- [32] P. Bjørn, M. Menendez-Blanco, and V. Borsotti, *Diversity in Computer Science: Design Artefacts for Equity and Inclusion*. Springer Nature, 2023.
- [33] N. Marsden, J. Hermann, and M. Pröbster, Developing Personas, Considering Gender: A Case Study. The Twenty Ninth Australian Conference on Human-Computer Interaction (OzCHI 2017), Brisbane, Australia, November 28-December 1, 2017.
- [34] W. Groeneveld, J. Vennekens, and K. Aerts, Identifying Non-Technical Skill Gaps in Software Engineering Education: What Experts Expect But Students Don’t Learn. *ACM Transactions on Computing Education*, 22(1):1-21, 2021.
- [35] *Gender Literacy: Enhancing Female Self-Concept and Contributing to the Prevention of Body Dissatisfaction and Eating Disorders*. By B. A. Sigall, M. S. Pabst. *Social Science Information*. Volume 44. Issue 1. 2005. Pages 85-111.
- [36] B. Boehm, A View of 20th and 21st Century Software Engineering. The Twenty Eighth International Conference on Software Engineering (ICSE 2006), Shanghai, China, May 20-28, 2006.