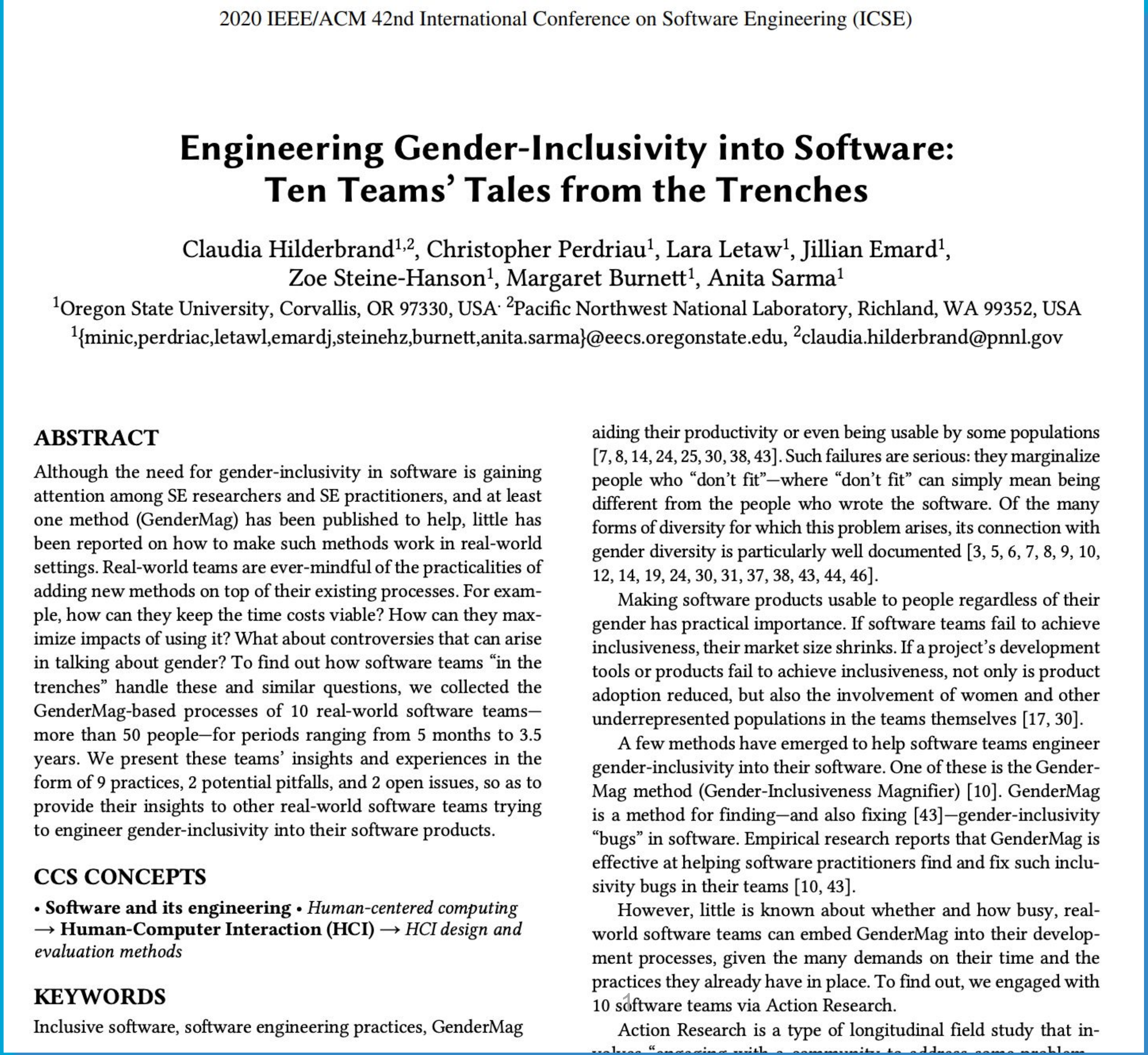
Engineering Gender-Inclusivity into Software: Ten Teams’ Tales from the Trenches

Claudia Hilderbrand, Christopher Perdriau, Lara Letaw, Jillian Emard, Zoe Steine-Hanson, Margaret Burnet, Anita Sarma

Presented by:

Louise Leibbrandt
Nienke Nijkamp
Gaspar Rocha
George Vegelien

TU Delft



2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)

Engineering Gender-Inclusivity into Software: Ten Teams’ Tales from the Trenches

Claudia Hilderbrand[1,2], Christopher Perdriau[1], Lara Letaw[1], Jillian Emard[1], Zoe Steine-Hanson[1], Margaret Burnett[1], Anita Sarma[1]
[1]Oregon State University, Corvallis, OR 97330, USA· [2]Pacific Northwest National Laboratory, Richland, WA 99352, USA
[1]{minic,perdriac,letawl,emardj,steinehz,burnett,anita.sarma}@eecs.oregonstate.edu, [2]claudia.hilderbrand@pnnl.gov

ABSTRACT

Although the need for gender-inclusivity in software is gaining attention among SE researchers and SE practitioners, and at least one method (GenderMag) has been published to help, little has been reported on how to make such methods work in real-world settings. Real-world teams are ever-mindful of the practicalities of adding new methods on top of their existing processes. For example, how can they keep the time costs viable? How can they maximize impacts of using it? What about controversies that can arise in talking about gender? 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—more than 50 people—for periods ranging from 5 months to 3.5 years. We present these teams’ insights and experiences in the form of 9 practices, 2 potential pitfalls, and 2 open issues, so as to provide their insights to other real-world software teams trying to engineer gender-inclusivity into their software products.

CCS CONCEPTS

• **Software and its engineering** • *Human-centered computing* → **Human-Computer Interaction (HCI)** → *HCI design and evaluation methods*

KEYWORDS

Inclusive software, software engineering practices, GenderMag

aiding their productivity or even being usable by some populations [7, 8, 14, 24, 25, 30, 38, 43]. 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, 5, 6, 7, 8, 9, 10, 12, 14, 19, 24, 30, 31, 37, 38, 43, 44, 46].

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, 30].

A few methods have emerged to help software teams engineer gender-inclusivity into their software. One of these is the GenderMag method (Gender-Inclusiveness Magnifier) [10]. GenderMag is a method for finding—and also fixing [43]—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 [10, 43].

However, little is known about whether and how busy, real-world software teams can embed GenderMag into their development processes, given the many demands on their time and the practices they already have in place. To find out, we engaged with 10 software teams via Action Research.

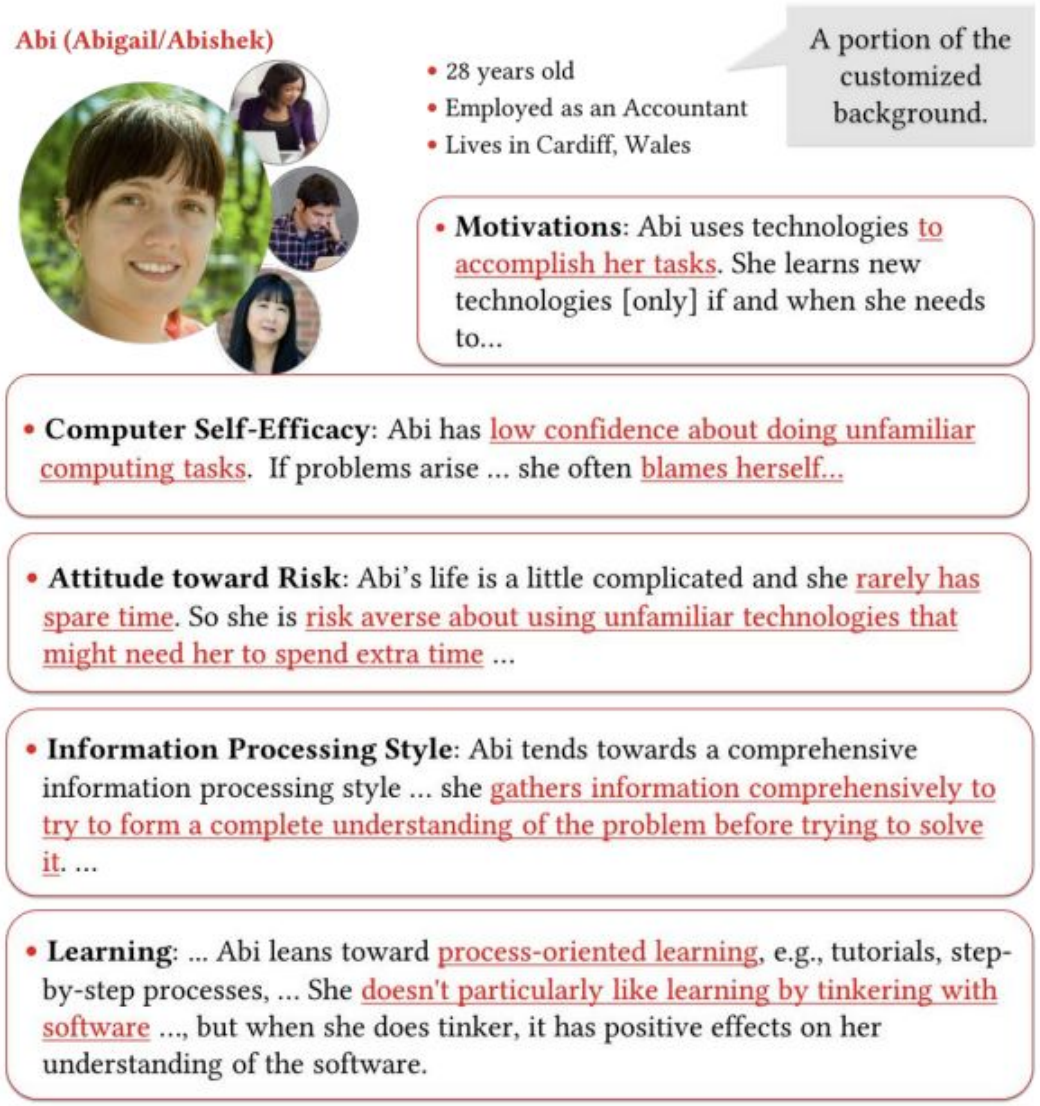
Action Research is a type of longitudinal field study that involves “engaging with a community to address some problem

Software has diversity problems

- Related to **gender inclusivity**
- There are methods to engineer gender inclusivity into software
- No studies on real-world application of these methods
- This is a study on **GenderMag**

What is GenderMag?

- A **software inspection** method
- Evaluation of user stories using 3 personas:
  - Tim: *traditional persona that software developers tend to design for*
  - **Abi: *represents the persona that is often overlooked***
  - Pat: *combination of Tim and Abi*



Abi (Abigail/Abishek)

- 28 years old
- Employed as an Accountant
- Lives in Cardiff, Wales

A portion of the customized background.

- **Motivations**: Abi uses technologies to accomplish her tasks. She learns new technologies [only] if and when she needs to...

- **Computer Self-Efficacy**: Abi has low confidence about doing unfamiliar computing tasks. If problems arise ... she often blames herself...

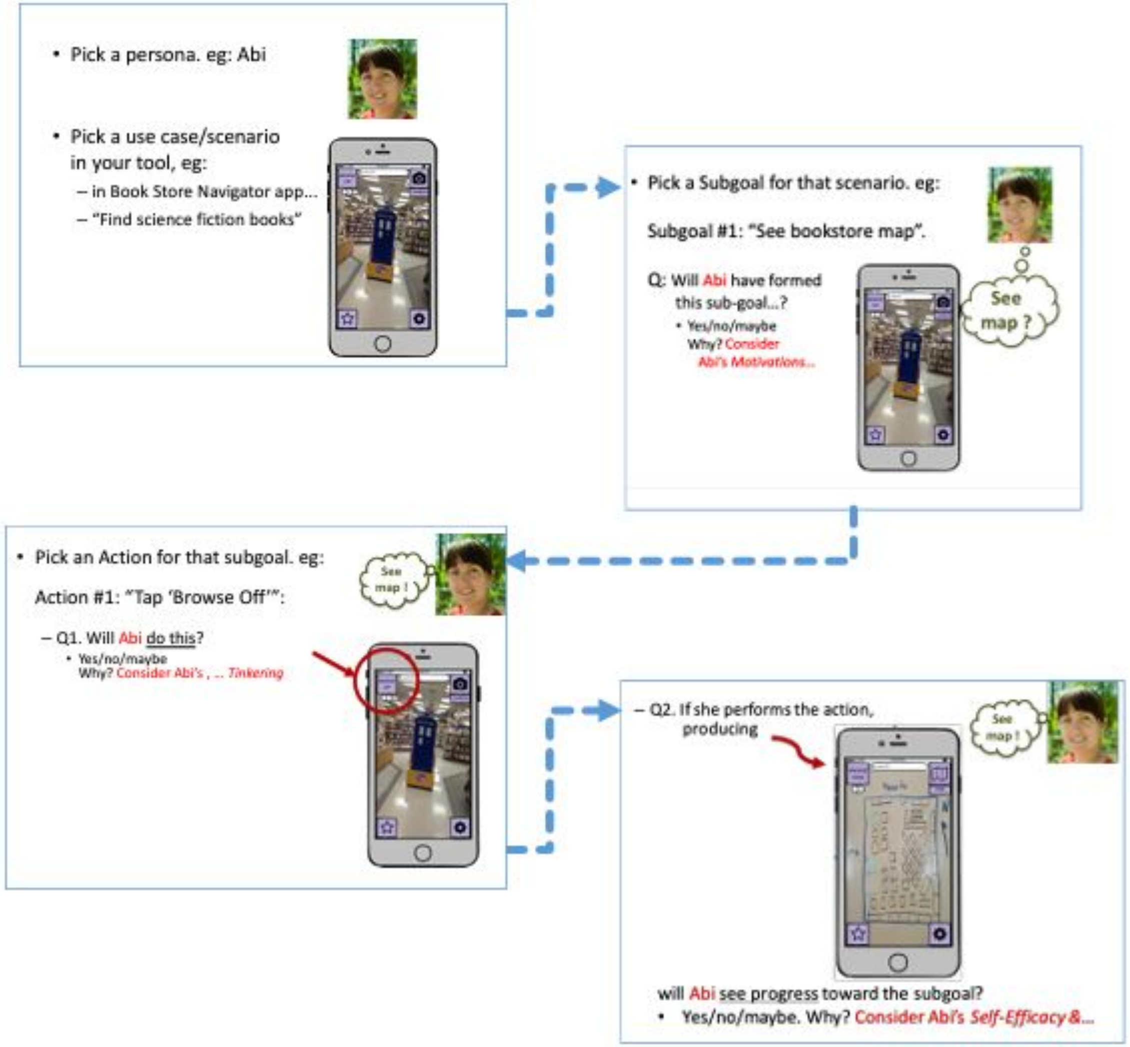
- **Attitude toward Risk**: Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time ...

- **Information Processing Style**: Abi tends towards a comprehensive information processing style ... she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. ...

- **Learning**: ... Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, ... She doesn't particularly like learning by tinkering with software ..., but when she does tinker, it has positive effects on her understanding of the software.

Example of GenderMag

- Pick a persona
- Pick a Subgoal for that scenario
- Pick an Action for that subgoal
- Evaluate 2 questions:
    - *Q1:* Will they know what to do at this step?
    - *Q2:* If they do the right thing, will they know they did the right thing and are making progress toward their goal?



- Pick a persona. eg: Abi
- Pick a use case/scenario in your tool, eg:
    - in Book Store Navigator app…
    - "Find science fiction books"

- Pick a Subgoal for that scenario. eg:

Subgoal #1: "See bookstore map".

Q: Will Abi have formed this sub-goal…?

- Yes/no/maybe
Why? Consider Abi's Motivations…

- Pick an Action for that subgoal. eg:

Action #1: "Tap 'Browse Off'":

- Q1. Will Abi do this?
    - Yes/no/maybe
    Why? Consider Abi's , … *Tinkering*

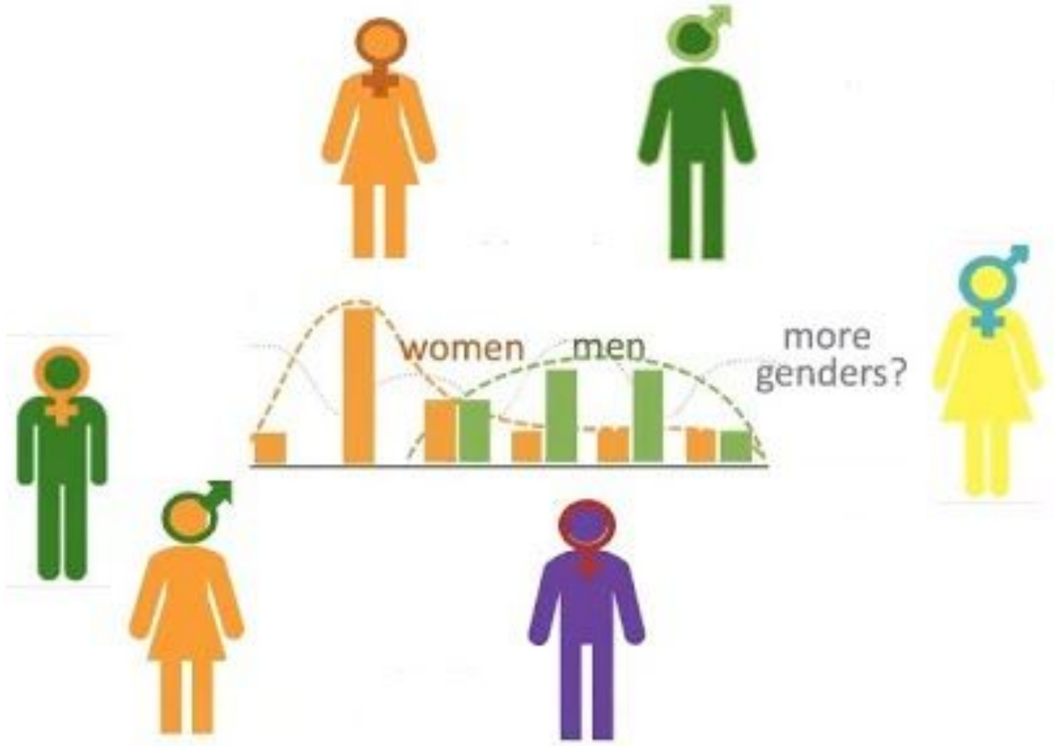
- Q2. If she performs the action, producing

will Abi see progress toward the subgoal?

- Yes/no/maybe. Why? Consider Abi's *Self-Efficacy* &…

Key Takeaways

- Abi as a **communication tool**
  - **comfortability** addressing gender issues
- No personal ego's
- Abi is the most **inclusive**
  - Focus on Abi first to create more accessible software
- Large groups = more perspectives and completeness of evaluation
- Small groups = quick evaluation
- Requirement: Decision making power



Thank you for your attention!