Women in Software Engineering

If you ask any experienced Software Engineer to list the 10 most influential people in the field, women will just appear in the list: Barbara Liskov; Adele Goldberg; Grace Hopper. The first programmer was a woman: Ada Lovelace. Programming is a discipline founded by a woman; how many other disciplines can you say that about? Women don't make the list because they're women who happened to become great Software Engineers; they're there because they're great Software Engineers who happened to be women.

So how does this happen? For one thing, Software Engineering is an unbelievably meritocratic discipline

For a discipline to be meritocratic, it (1) can't have barriers to entry, (2) can't have barriers to publication, and (3) needs objective measures to evaluate work.

1. There aren't any barriers to entry. Anyone can write code; you don't even need a computer—obviously it helps. The only prerequisite is that they're interested and curiosity. There aren't political, business, or academic institutions that people need to be let into to work.

2. There aren't any barriers to publication. There aren't publishers that decide which work is worthy of publication. A manager that needs to ship some code isn't going to decide not to ship code that does what it's supposed to because they don't like the person. If a Software Engineer develops an algorithm that solves a problem no one's ever been able to solve, they just put it up on GitHub.

3. As Software Engineers, we use objective measures to evaluate our work. To evaluate code quality, we follow principles like *SOLID* and have equations like *asymptotic space and time complexity*. To evaluate our system designs, we user metrics called *Service Level Indicators* (*SLI*s). No one has used a technology that's slower or less reliable because of a personal aversion they have against a person or a group. Even if that's not true, the idea that someone would relegate themselves to a lifetime of "loading" screens out of spite is too funny to get mad over.