

The SocioeconomicMag Method

To Find and Fix Socioeconomic-Inclusiveness “Bugs” in Software

Recent updates by: Margaret Burnett, Anita Sarma, Catherine Hu, Abrar Fallatah, Puja Agarwal, and many others

Individual differences in ways people use software often cluster by socioeconomic status – especially when problem solving, such as working with visualizations, figuring out budgets, or debugging. However, many software features are inadvertently designed in a way that fails to support users with lower socioeconomic status.

The SocioeconomicMag method helps software developers & usability professionals find and fix software features with such socioeconomic status-inclusiveness “bugs”.

How the SocioeconomicMag Method Works

- At the SocioeconomicMag method’s core are six problem-solving facets that matter to software’s socioeconomic status-inclusiveness:
 - a user’s access to reliable technology,
 - their communication literacy, education, and culture,
 - their attitudes toward technology risk,
 - their concern for technology privacy and security,
 - their perceived control and attitude toward authority,
 - and their technology self-efficacy.
- The SocioeconomicMag method brings these facets to life with 3 personas (Dav, Ash, and Fee), and sets the personas and facets into a simple, systematic process (the SocioeconomicMag Cognitive Walkthrough).
- Software developers can use the SocioeconomicMag method to identify and remove barriers that may disproportionately affect certain socioeconomic groups.



Ash



Dav



Fee

The SocioeconomicMag Method

To Find and Fix Socioeconomic-Inclusiveness “Bugs” in Software

Recent updates by: Margaret Burnett, Anita Sarma, Catherine Hu, Abrar Fallatah, Puja Agarwal, and many others

A SocioeconomicMag Evaluation Looks Like This:

