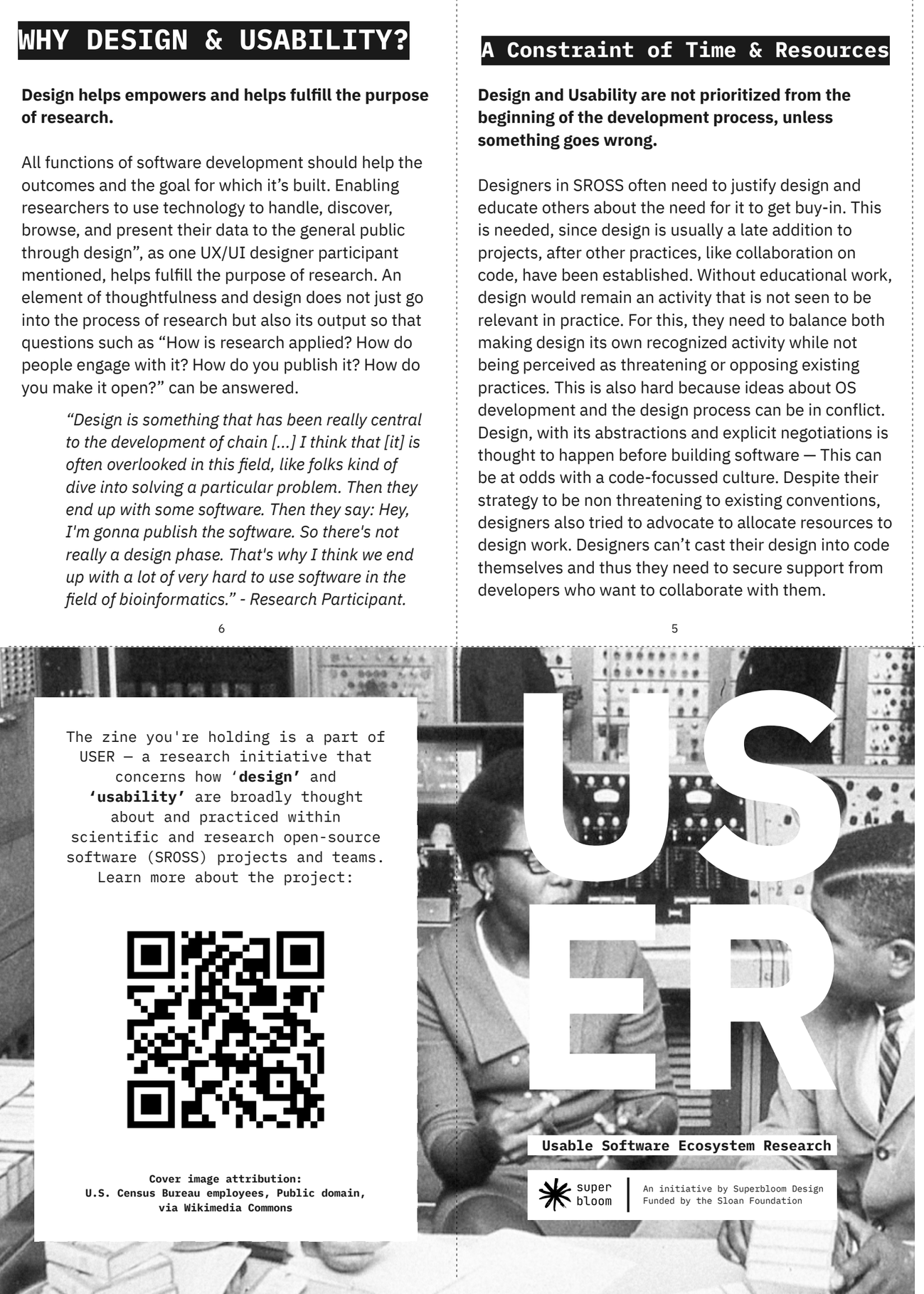
The zine you're holding is a part of USER — a research initiative that concerns how **'design'** and **'usability'** are broadly thought about and practiced within scientific and research open-source software (SROSS) projects and teams. Learn more about the project:

Cover image attribution:
U.S. Census Bureau employees, Public domain, via Wikimedia Commons



Usable Software Ecosystem Research (USER) project is a research initiated by Superbloom Design and funded by the Sloan Foundation. It explores how open-source scientific and research software (SROSS) teams understand, consider, and undertake usability and design opportunities in their project.

Through a variety of research methods such as literature review, semi-structured interview, survey, and ecosystem mapping, the research aims to obtain a better understanding of:

1. How norms in academic, science, and/or open-source working environments affect the choices teams make around their users and different kinds of design interventions.
2. How team dynamics and trust affects those choices.
3. What teams would need to be interested in or able to prioritize usability and design in their work.

In this zine, we'll share with you an overall of how design and usability is perceived in the SROSS ecosystem.

Design & Usability

>> as defined in SROSS <<

During the period of 3 months, from December 2022 to Febuary 2023, we spoke to 24 research participants who are maintainers, developers, designers, scientists, researchers, funders, Open Source Program Offices (OSPOs) and stakeholders – all involved with creating and maintaining open source scientific & research software.

One of the questions we asked was **"What does "design" mean to you in the context of software projects?"** — Here are what we've learned:

According to our research participants, the terms design and usability mean a variety of different things: from designing code, form, and functionality, to planning and documentation. There is a common association between the need for design and usability, and the need to obtain an understanding of how the project team, clients, and end-users are using, interacting with the software and to make sure it works for everyone.

A LACK OF UNDERSTANDING

SROSS teams often express an understanding of how dedicated efforts on improving the user experience or addressing the ease of use could improve their tools and services, but they don't realize that these efforts are connected to Usability.

Developers and maintainers of early-stage projects with low user numbers often enjoy answering user support emails — Here, they are doing customer support-informed usability improvements.

On the contrary, larger SROSS project teams often find support requests time-consuming and not efficient uses of their time. A research participant described the desire to make the software easier to use to free up time for other development work and priorities. They described how the efforts to "encapsulate or abstract the complexity" for users has been aided by documentation improvements — Here, they are performing usability improvements in a way that makes sense to them and their training: writing and updating documentation and responding to requests for help.

Tool Makers = End Users

SROSS tool teams, like most OSS tools, have an 'ideal' user type and level of proficiency. Most OSS build and maintain for these users or people to become these kinds of users — The user best described as 'myself' by developers of SROSS.

A community manager addressed the reliance on norms and knowledge in a community and how this really only suffices until a point where someone gets stuck. In our casual conversations in OSS we found that getting lost is fairly common despite your knowledge of norms and processes and skills. SROSS is most often built by a single person or small collection of people that cycle through intensity of activity on an OSS project. These are the people that are likely to have the most comprehensive knowledge of operating that OSS and even if they document incredibly well, there will be quirks in that OSS that others will not understand until they ask. When the configurers of the OSS are making judgements and assumptions about general use using their own experiences as a benchmark which are often highly knowledgeable benchmarks — This is where Usability can make a difference.

A Constraint of Time & Resources

Design and Usability are not prioritized from the beginning of the development process, unless something goes wrong.

Designers in SROSS often need to justify design and educate others about the need for it to get buy-in. This is needed, since design is usually a late addition to projects, after other practices, like collaboration on code, have been established. Without educational work, design would remain an activity that is not seen to be relevant in practice. For this, they need to balance both making design its own recognized activity while not being perceived as threatening or opposing existing practices. This is also hard because ideas about OS development and the design process can be in conflict. Design, with its abstractions and explicit negotiations is thought to happen before building software — This can be at odds with a code-focussed culture. Despite their strategy to be non threatening to existing conventions, designers also tried to advocate to allocate resources to design work. Designers can't cast their design into code themselves and thus they need to secure support from developers who want to collaborate with them.

WHY DESIGN & USABILITY?

Design helps empowers and helps fulfill the purpose of research.

All functions of software development should help the outcomes and the goal for which it's built. Enabling researchers to use technology to handle, discover, browse, and present their data to the general public through design", as one UX/UI designer participant mentioned, helps fulfill the purpose of research. An element of thoughtfulness and design does not just go into the process of research but also its output so that questions such as "How is research applied? How do people engage with it? How do you publish it? How do you make it open?" can be answered.

> *"Design is something that has been really central to the development of chain [...] I think that [it] is often overlooked in this field, like folks kind of dive into solving a particular problem. Then they end up with some software. Then they say: Hey, I'm gonna publish the software. So there's not really a design phase. That's why I think we end up with a lot of very hard to use software in the field of bioinformatics." - Research Participant.*