

WHY DESIGN & USABILITY?

Design helps empower 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.

A Late Addition

Designers in Scientific & Research OSS often need to justify design and educate others about the need for it to get buy-in.

This is particularly expected because design is usually a late addition to projects. Other practices, like collaboration on code, are often established well before a design process. Without educational work, design would remain an activity that is not seen to be relevant in practice. Designers need to balance both making design its own recognized activity while not being perceived as threatening or opposing existing practices.

Despite designers' approach to be non threatening to existing conventions, they also try to advocate to allocate resources for design work. Designers can’t cast their design into code themselves and thus they need to secure support and buy-in from developers and maintainers who want to collaborate with them and respect and value their work.

Tool Makers = End Users

Scientific & Research OSS software 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 OSSRS.

We found that getting stuck when working with software is fairly common even for experienced people. Scientific & Research OSS is often built by a single person or small team. Even if they document their work well, there will be quirks in that OSS that others will not understand. Often, the creators will use their own experiences as a benchmark though they will represent the upper range of skill.

This is where Usability can make a difference: Bringing in methods that allow creator to check how well their software is understood and if it can be used by people with different skills than their own.

A LACK OF UNDERSTANDING

Science and Research OSS misses out on the ways that dedicated Usability efforts could improve their tools. They know about the problems, yet they don't realize that these problems are connected to Usability.

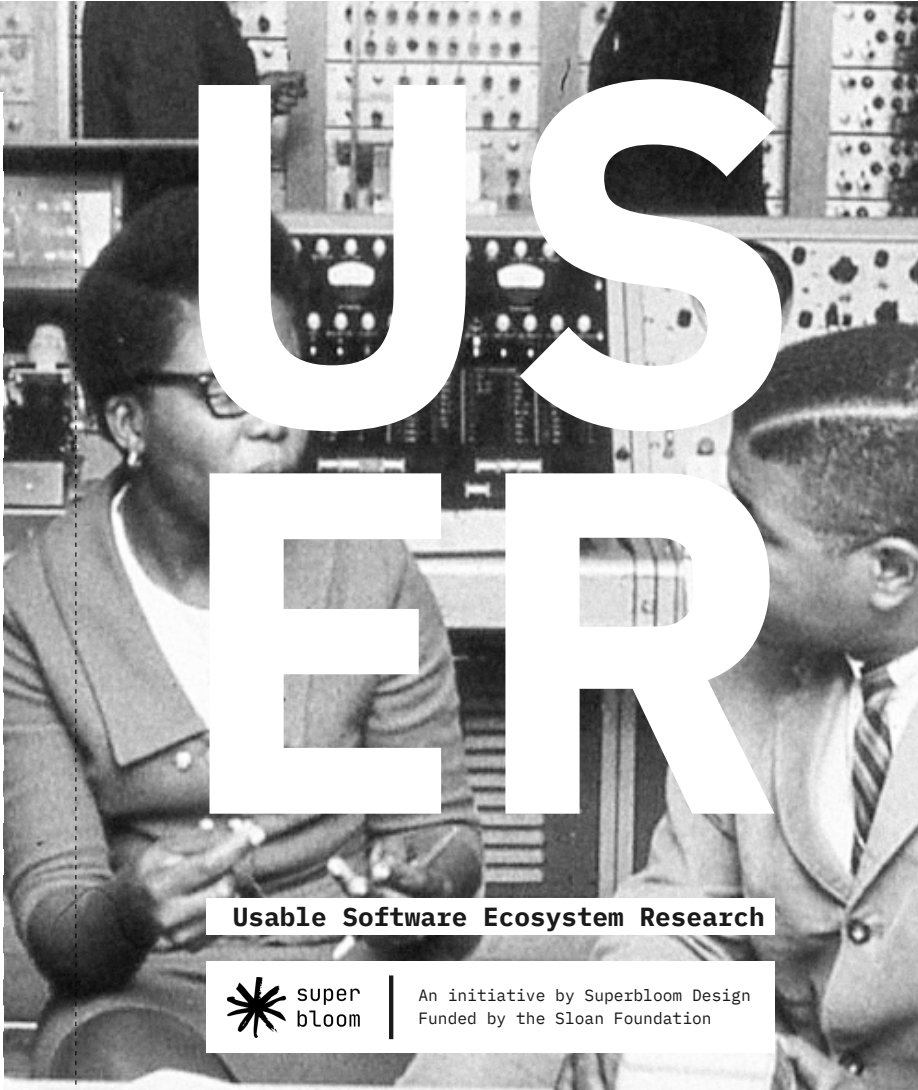
Developers and maintainers of early-stage projects with low user numbers often enjoy answering user support emails. This also helps them to learn about problems and to make usability improvements — they are doing customer support-informed usability improvements.

On the other hand, larger project teams often find support requests time-consuming: “If software would be easier to use for people, that would kind of free up more of our time to actually work on developing [...]” They often try to solve these problems by adding documentation or improving it — Here, they are performing usability improvements in a way that makes sense to them and their training, yet not fully knowing that these users' struggles could be better solved by improving the usability of the software.


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 & Research open-source software 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



super bloom

An initiative by Superbloom Design  
Funded by the Sloan Foundation

The Usable Software Ecosystem Research (USER) project was initiated by Superbloom Design and funded by the Sloan Foundation. It explores how Scientific & Research open-source software teams understand, consider, and undertake usability and design opportunities in their projects.

Through a variety of research methods such as literature reviews, semi-structured interviews, surveys, 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 overview of how design and usability is perceived in the Scientific & Research open-source software ecosystem.

Design & Usability

>> as defined in Scientific & Research open-source software <<

From December 2022 to February 2023, we spoke to 24 research participants who are maintainers, developers, designers, scientists, researchers, funders, Open Source Program Offices and stakeholders – all involved with creating and maintaining Scientific & Research OSS.

One of the questions we asked was “What does design mean to you in the context of software projects?” — Here is 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 software architecture, 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 and interacting with the software and to make sure it works for everyone.