Design und User Research

Use Cases und User Research, Design

- Design für die vielen – wie Human-Centered Design Nutzer*innen in den Fokus rückt
- Link collection design, UI and user research
- Software intuitiv gestalten - Podlove Publisher Onboarding & Import Assistant
- Usability und Zielgruppenfokus in FOSS-Projekten — Evaluation der Prototype Fund Runden 15 & 16

□□ Design für die vielen – wie Human-Centered Design Nutzer*innen in den Fokus rückt

Eileen Wagner spricht im Interview über Human-Centered Design – und darüber, was das Designprinzip für Public Interest Tech bereithalten kann.

Dein Steckenpferd ist Human Centered Design – was ist das?

Human-Centered Design (HCD), manchmal auch unter “Design Thinking” bekannt, ist ein Sammelbegriff für Designmethoden, die sich durch eine empirische Herangehensweise auszeichnen. Nutzende werden als Expert*innen befragt und das Produkt wird auf Basis der Ergebnisse entwickelt. Sind die Bedürfnisse von Nutzenden erfüllt, ist das Design gut.

Dabei beinhaltet HCD einen ganzen Wertekanon – von Empathie und Optimismus bis dahin, aus Fehlern zu lernen, auszuprobieren statt zu diskutieren und sich auch in Ambivalenzen wohlzufühlen. Im HCD muss niemand Expert*in sein, um etwas gut zu machen. Ganz im Gegenteil: Unwissen und Neugier können sogar eine Stärke sein!

Was hat Human-Centered Design mit Public Interest Tech zu tun?

Wie auch der Begriff “Public Interest Tech” kommt HCD aus den USA. Ursprünglich ging es bei Public Interest Tech darum, zum Beispiel die Interaktion zwischen Bürger*innen und der Regierung zu vereinfachen und darum, amtliche Prozesse und Tools tatsächlich nutzbar zu machen. Das sind die gleichen Werte, die auch HCD vertritt.

Ein gutes Beispiel ist die Herangehensweise von Code for America, eine der ersten Public-Interest-Tech-Organisationen. Dort geht es nicht nur darum, dass Code und Daten stimmen, sondern ob sie verständlich und zugänglich präsentiert werden, denn nur so haben Nutzer*innen wirklich etwas davon.

In Deutschland ist HCD weniger verbreitet, denn hier gibt es eher traditionelle Software-Unternehmen, die Prozesse weniger iterativ angehen. Gerade bei Public Interest Tech und den vielen ehrenamtlichen Open-Source-Entwickler*innen herrscht eine ganz andere Kultur vor, sodass

bisher wenig mit Prinzipien des HCD gearbeitet wird. Der Ruf von HCD ist eben etwas Silicon Valley – zurecht!

Technologie wird uns nicht retten. Aber vielleicht Design? Denn Design kann dabei helfen, das große Ganze zu sehen und auf diese Weise neue Lösungswege aufzeigen. Wie funktioniert das?

Man tappt leicht in die Falle des Techno-Solutionism – also das Denken, dass Technologie immer die Lösung für alles ist – und ich finde, Design kann diese Perspektive etwas eindämmen. Design fragt: Wem nützt das Tool? Wie soll es genau funktionieren? Was passiert davor? Und was passiert danach? Wie passt das Produkt ins Gesamtsystem?

Auf diese Weise erreicht man, dass ein **Produkt nicht einfach auf den Markt gebracht wird, sondern dass es genau dort landet, wo es benötigt wird** – bei Nutzer*innen, die schon darauf warten. Design ist also ein mächtiges Werkzeug. Aber wie heißt es so schön: „Wer als Werkzeug nur einen Hammer hat, sieht in jedem Problem einen Nagel.“ Natürlich löst Design genauso wenig wie Technologie realpolitische oder soziale Probleme. Darauf müssen wir als Public Interest Technologists immer achten.

Kannst du Beispiele dafür nennen, wie Design innovative Lösungswege aufzeigt?

Ein super Beispiel kommt von Code for America. In Kalifornien gibt es ein Onlineportal für Lebensmittelmarken – “GetCalFresh”. Die Entwickler*innen haben sich in diesem Fall ganz genau angesehen, in welchem Kontext diese Marken benötigt werden. Also versuchen sie zu erfahren, wie die Lebenswirklichkeit der Menschen, die sie beziehen, aussieht und schauen dann, wie sie in diesen bestimmten Kontext hineinarbeiten können, damit das Portal am Ende für sie funktioniert.

Dafür haben sie verschiedene User Research Studien gemacht, zum Beispiel zum Dokumentenupload. Das Ergebnis sieht so aus, dass bei “GetCalFresh” nicht gleich bei der Anmeldung alle Dokumente hochgeladen werden müssen, die den Bedarf der Person nachweisen. Das macht die Anmeldung um einiges einfacher, denn viele Menschen haben einfach keinen Scanner zuhause. Sie können sich trotzdem anmelden und die Dokumente einfach nachreichen. Das klingt erst einmal nicht nach einem besonders ausgefeilten Feature, aber viele kleine Dinge können den Prozess maßgeblich beeinflussen und am Ende dafür sorgen, dass ein gutes und nützliches Ergebnis entsteht, das auf die bestehenden Bedürfnisse abgestimmt ist. Ich empfehle die ganze Reihe von Case Studies von diesem Team, sie sind lesenswert!

Public Interest Tech hinterfragt die Art und Weise, in der Software entwickelt wird und stellt eine Alternative dafür dar. Kann mit HCD eine neue, stärker an Gemeinwohl orientierte Wertorientierung in der Software-Entwicklung erreicht werden?

Der Designdiskurs des 20. Jahrhunderts ist sehr kontrovers. Insgesamt werden viele verschiedene Ansprüche an Design formuliert, zum Beispiel der, dass Design nicht nur menschenzentriert sondern auch umweltzentriert sein soll. Und einige Stimmen sagen sogar, Design brauche überhaupt keine Methodologie.

Aus meiner Sicht ist HCD nicht an sich gut oder schlecht, es ist ein Werkzeug, das auf unterschiedliche Weise eingesetzt werden kann. Es berücksichtigt klare Werte, aber die konkreten Ergebnisse kommen immer aus den übergeordneten Zielen und der konkreten Herangehensweise.

Dark Patterns beispielsweise sind aus ethischer Sicht natürlich nicht gut, aus Designsicht erfüllen sie aber ihren Zweck. Und warum sind denn Google und Facebook so erfolgreich? Zum großen Teil wegen des UX Designs! Da kann man sich fragen: Wer ist daran Schuld? Sind es die Designer*innen, die im Sinne des Unternehmenszwecks designen, oder ist es der Unternehmenszweck?

Deswegen denke ich, dass nicht die Methodologie von HCD überarbeitet sondern die Ziele neu definiert werden müssen. Denn mit den richtigen Zielen ist HCD ein mächtiges Werkzeug.

Die Geschichte von HCD ist spannend, aber nicht unproblematisch. Sie ist sehr US-amerikanisch, sehr corporate – und das gefällt nicht allen. Die Frage, die wir stellen müssen, lautet deshalb: **Wie können wir Teile der Methodologie übernehmen, ohne die Werte des Silicon Valleys eins zu eins mit zu übernehmen?**

Was sind neue Entwicklungen in HCD, die auch für Public Interest Tech interessant sein könnten?

Da geht es aktuell in verschiedene Richtungen. Ein interessanter Ansatz ist beispielsweise “Consequence Scanning”. Hierbei werden Folgen des Designs so früh wie möglich betrachtet und abgeschätzt.

Bei der Betrachtung der Zielgruppen lautet eine neue Idee: “Personas non gratas”. Im Normalfall werden im Designprozess ja sogenannte Personas als idealtypische Zielgruppe entworfen. Bei der *Persona non grata* geht es hingegen um die Menschen, die nicht direkt als Zielgruppe geplant sind, das Produkt aber trotzdem nutzen werden.

Auch spannend ist die “Design Justice” Bewegung. Der Gedanke dahinter ist, dass Designer*innen nicht den Alleinanspruch auf gute Ideen haben. Aus diesem Grund sollten alle Betroffenen an einem Design arbeiten und Lösungen im Co-Design entwerfen. Das ist besonders wichtig, wenn es einen großen Unterschied in den Machtverhältnissen gibt. Aus der Entwicklungszusammenarbeit kennen wir genügend Beispiele, in denen Institutionen aus dem globalen Norden sich Lösungen für Menschen im globalen Süden überlegen, welche diesen dann einfach vorgesetzt werden, ohne dass ihre Bedürfnisse erfragt und einbezogen wurden. Oder aber die Bedürfnisse wurden erfragt, aber die Betroffenen haben trotzdem kein Entscheidungsrecht für Lösungen, die ihr Leben beeinflussen. Mit Co-Design wird versucht, das von Anfang an zu vermeiden.

Wie kann man es schaffen, einen Human Centered Ansatz flächendeckend in der Technologieentwicklung durchzusetzen? Wo sind die Hürden?

Ich denke, HCD wird sich in der IT früher oder später durchsetzen, denn es ist schlicht und einfach profitabel. Die Frage ist deshalb eher: Wie schaffen wir es, dass es flächendeckend auch in Public Interest Technologien und im öffentlichen Sektor Anwendung findet?

Fördermittelgeber müssen verstehen, dass HCD nicht das Sahnehäubchen sondern grundlegend für gute Technologie ist. Deswegen sollte es sich in viel mehr Förderprogrammen wiederfinden. Und HCD benötigt dringend auch mehr Fürsprache: **Viele Menschen reden enthusiastisch über Design und viele Menschen sprechen begeistert von Public Interest Tech. Das Problem ist, dass das in der Regel nicht dieselben Menschen sind.** Public Interest Tech und HCD können und sollen voneinander lernen und gemeinsam wachsen. Dafür braucht es gemeinsame Arbeit und mehr persönliche Verbindungen.

Eine weitere Herausforderung sehe ich in Fragen der Kultur. Wir müssen uns fragen, was HCD im deutschen Kontext eigentlich ist. Das Konzept sollte definiert und übersetzt, nicht als reine Imitation der US-amerikanischen Version verstanden werden. Wenn wir das schaffen, kann HCD einen großen Beitrag zur Entwicklung von Public Interest Technologien in Deutschland spielen.

Eileen Wagner war von 2015 bis 2017 als Content-Managerin im Team des Prototype Fund. Aktuell arbeitet sie aus der Perspektive von Human-Centered Design daran, offene, sichere und menschenwürdige Technologien zu gestalten: Als UX Coach von Simply Secure, auch für die Projekte des Prototype Fund.

□□ Link collection design, UI and user research

Accessibility

See our link collection on accessibility

Visual design

- Open Source Design, *Resources*: https://opensourcedesign.net/resources/
- Senongo Akpem, *Cross-Cultural Design*: https://abookapart.com/products/cross-cultural-design

Color

- Read:
  - Superbloom, *Don't Let Color Drown Out Your Message*: https://superbloom.design/learning/resources/color-message/
  - Danny Sapio, *10 principles for Color Usage in UI Design*: https://uxdesign.cc/10-principles-for-color-usage-in-ui-design-65174b213004
- Color palette tools:
  - *Popular color palettes*: https://coolors.co/palettes/popular
  - Color combinations: https://www.colourlovers.com/
  - Color picker with HTML Hex codes: htmlcolorcodes.com
- Color System: https://m3.material.io/styles/color/system/overview
- Use *Contrast Grid* for checking color contrast for accessibility: https://contrast-grid.eightshapes.com/

Fonts

- Sustainable Open Source Design (Podcast), *Jérémy Landes on Type Designing*: https://sosdesign.sustainoss.org/16
- Finding fonts
  - https://www.velvetyne.fr/about/
    - "Researching and disseminating typography and typeface creation"

    - https://www.fontsquirrel.com/
        - Font squirrel has a useful Open Font License filter option

Icons & Mascots

- Finding icons: thenounproject.com
- Graphéine, *A history of mascots*: https://www.grapheine.com/en/graphic-design-en/history-of-mascots

Branding

- Read:
    - Superbloom, *Open source & branding – a contradiction?* https://superbloom.design/learning/blog/branding-tails/
    - Superbloom, *How to name your app*: https://superbloom.design/learning/resources/how-to-name-your-app/
    - Superbloom, *Communications Toolkit for Civic & Nonprofit Tech*: https://superbloom.design/learning/resources/comms-toolkit/
    - New York Times, *The Weird Science of Naming New Products*: https://www.nytimes.com/2015/01/18/magazine/the-weird-science-of-naming-new-products.html
- Working with the community:
    - Open Food Network as an example of an OSS community doing a community led project regarding vision and mission: https://community.openfoodnetwork.org/t/ofn-organizational-vision-request-for-feedback/2090
    - An article on how Simple organised their open source brand design: https://medium.com/simple-dot-org/open-source-brand-design-for-simple-474f03a7fc15
- A list of helpful (OSS) brand standards/guidelines:
    - Red Hat brand standards: https://www.redhat.com/en/about/brand/standards
    - OpenSSF brand guidelines: https://openssf.org/about/brand-guidelines/
    - Beyond Gravity brand portal: https://brand.beyondgravity.com/d/jbxsqV491EeP
    - Sonos brand identity: https://www.sonos.com/en-us/blog/sonos-brand-design-refresh
    - OpenAI brand guidelines: https://openai.com/brand/

Logo design

- AntennaPod, *Getting to a new icon*: https://antennapod.org/it/blog/2020/06/getting-to-a-new-icon

    - A good example about how to write down your requirements
- Fordes, *Designing a logo for your software project*: https://www.fordes.de/posts/logodesign_softwareproject.html
    - A worked example for a logo creation process

Working with designers

Finding a designer

- Open Source Design Jobboard: https://opensourcedesign.net/jobs/
- User Experience & Discovery Lab: https://www.opentech.fund/labs/uxd-lab/
    - The lab provide free user-experience research and design support to internet freedom projects.
- Hylke Bons is designing app icons & icon sets - for free, financed by the community: https://planetpeanut.studio/services

Redesign & preparations for working with designers

- *Design Project Scoping Guide*: https://dschool.stanford.edu/resources/design-project-scoping-guide
- DesignKit, *Align on Your Impact Goals*: https://www.designkit.org/methods/align-on-your-impact-goals.html
- DesignKit, *Define Your Audience*: http://www.designkit.org/methods/define-your-audience.html

UX/UI design

Design systems and consolidation

- Figma, *Name and organize components*: https://help.figma.com/hc/en-us/articles/360038663994-Name-and-organize-component
- Example of a big team of OSS designers working on a single OSS tool: https://www.figma.com/design/GLg0zPNnwMEGIb9TytCrxQ/Resilience-App?node-id=1257-553&node-type=canvas
- *Design for Code* on Figma: https://www.figma.com/community/file/967069265207476318/design-for-code-config-2021
- FOSDEM talk on *Documenting a Design System. Lessons learned from open sourcing the Orbit docs*: https://archive.fosdem.org/2022/schedule/event/osd_lessons_learned_from_open_sourcing

_the_orbit_docs/

- Joplin, *Design patterns, systems, guidelines*: https://notes.saneux.ei8fdb.org/shares/oSoPeKIFSz3SNyZ9c44gf1
- Women-Centric Design: https://www.womencentricdesign.org/

Usability principles and rules-of-thumb

- Jan Dittrich, Advice *I give when helping teams with UX*: https://www.fordes.de/posts/TipsUXSmallProjects.html
- J. Raskin, *Intuitive equals familiar*: https://www.asktog.com/papers/raskinintuit.html
  - “how to make something can be used as something that is familiar to users” can usually be answered more productively than "how to make this intuitive"
  - "*Don’t make me think*" (Steve Krug) is an excellent, short introduction: https://sensible.com/dont-make-me-think/
- AskTog, *First Principles of Interaction Design*: https://asktog.com/atc/principles-of-interaction-design/
- Icons and graphical control design:
  - Icons are often harder to understand than labels, see *Icon Usability*: https://www.nngroup.com/articles/icon-usability/
  - *Write on the button: the grammar of interactivity*: https://www.theguardian.com/info/developer-blog/2013/jan/02/interactive-button-text-grammar
  - *Progressive Disclosure* as a way to manage complexity: https://en.wikipedia.org/wiki/Disclosure_widget
    - The typical example is a print dialog: Show essentials (“Print”-button, Printer selection if multiple are connected) and offer detailed settings on demand (page margins, dpi selector , duplex etc.). The settings have “good” defaults, but they do not *need* to be changed to get the text printed on a page.
- Web form design:
  - *Website Forms Usability: Top 10 Recommendations*: https://www.nngroup.com/articles/web-form-design/
  - Luke Wroblewski, *Web Form Design. Filling in the blanks* (Thema): https://rosenfeldmedia.com/Themen/web-form-design/
  - Caroline Jarret/Gerry Gaffney, *Forms that Work. Designing Web Forms for Usability* (Thema): https://www.sciencedirect.com/Thema/9781558607101/forms-that-work
- *Macintosh Human Interface Guidelines*: http://interface.free.fr/Archives/Apple_HIGuidelines.pdf
  - The 1995 *Machintosh Interface Guidelines* do not help you to up-to-date interfaces, but contain a lot of useful principles.
- Divio, *The Documentation System*: https://docs.divio.com/documentation-system/

Tools

- *Penpot* is a FOSS-alternative for Figma, a UI design tool: https://penpot.app/
- Tools for wireframes: Excalidraw seems to be a favorite: https://excalidraw.com/

User research & journeys

- Visualize user activities over time with *user journey mapping*:
  https://www.usability.de/leistungen/methoden/user-journey-mapping.html
- Superbloom, *Qualitative Interviewing*:
  https://simplysecure.org/resources/qualitative_interviewing.pdf
- Jan Dittrich, *A Beginner's Guide to Finding User Needs*: https://urbook.fordes.de/
- *Empathy map*: Start with an assumption for user group/s and see if the empathy map indicates that your idea is going to support their goals or resolve their problems:
  https://commons.wikimedia.org/wiki/File:Empathy_map.jpg
- Superbloom, *Using Personas in Open-Source Projects*:
  https://superbloom.design/learning/blog/personas/

Evaluation methods

On finding out if your ideas work. Very often this is done too late – with paper prototyping you can run a simple evaluation within a few hours, so do test early and often!

Prototyping

You do not need an almost finished product to test (the earlier the better, actually): You can evaluate a single feature (see *tracer bullet development*), a mockup (see *penpot* or *figma*) or a *paper prototype*.

- *Tracer Bullet Development*: https://growsmethod.com/practices/TracerBullets.html
- Paper prototyping, a way to try ideas with users without needing to write any code.
  - *Paper prototyping* on Wikipedia: https://en.wikipedia.org/wiki/Paper_prototyping
  - Alistapart, *Paper Prototyping*: https://alistapart.com/article/paperprototyping/
- Concept prototype: Testing your idea when it's still at the conceptual stage: Keep it basic, keep it at the big picture level (so users won't focus on the finer details, like the color, design etc), use an exisiting product that is similar to get the users talking.

Analytics

- Privacy-preserving tracking of user actions: https://cleaninsights.org/
- Google analytics alternative: https://matomo.org/

Heuristic evaluation

Using rules to spot usability problems in a usage scenario; ideally done by several people.

- *10 Usability Heuristics for User Interface Design* the most common set of rules: https://www.nngroup.com/articles/ten-usability-heuristics/
- *How to conduct a heuristic evaluation*: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Usability tests & user testing

In a usability test, potential users try to use an early version of your product to find potential problems that you need to fix. User testing is not meant to sell the idea to the users, it's to gather data/feedback that supports you to continue product development, so use any resources available as early as possible to communicate the idea.

To read, listen and watch

- Superbloom, *How To Test Software For Usability And Usefulness When You Haven't Finished Writing The Code*: https://superbloom.design/learning/blog/how-to-test-software-for-usability-and-usefulness-when-you-havent-finished-writing-the-code/
- When is the right time to do user testing? As early as possible is often not ideal, since projects first need to figure out where they want to go. See Christoph Lattemann, Stefan Stieglitz, F*ramework for Governance in Open Source Communities*: https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/1824/file/stieglitz_lattemann_2005.pdf
- Before testing or having a software, maybe you want to document user context and processes: https://jdittrich.github.io/userNeedResearchBook/#coDocumenting
- *A Beginner's Guide to Finding User's needs* by Jan Dittrich: https://jdittrich.github.io/userNeedResearchBook/#dataGathering
- In general, it is better to do several smaller usability tests than a one or two big ones. Even small numbers of users can help to find problems. A classic article on this is *Why you only need to test with 5 users*: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/
- Know what power/significance are and how they are related to sample size, what you know from academia may be way too conservative for your needs. See also *Improving your statistical inferences* by Daniël Lakens: https://lakens.github.io/statistical_inferences/
- The Thema *Rocket surgery made easy* (Steve Krug) describes usability testing (and recruiting for- and communication of it): https://sensible.com/rocket-surgery-made-easy/
- Superbloom, *A Guide to Gathering Feedback*: https://superbloom.design/learning/blog/feedback-gathering-guide/
- Superbloom, *User Testing Cheat Sheet*: https://superbloom.design/learning/blog/user-testing-cheatsheet/

- *A Dev's guide to User testing*, made by Superbloom, has resources that can help OSS tool teams start to do user testing and usability testing in their OSS tool projects either with or without a designer helping out: https://sprblm.github.io/devs-guide-to/
- *Usability Testing can be fun! An FAQ for small open source teams*: https://usable.itch.io/user-testing-can-be-fun-a-guide-for-oss-developers-and-tool-teams-on-how-to-user
- User testing examples:
  - for the process: https://docs.ushahidi.com/platform-developer-documentation/dev-legacy-v3/design/user-testing-process
  - for design documentation: https://github.com/ushahidi/platform/issues/3681
  - Here is also an example video for a usability test: https://www.youtube.com/watch?v=QckIzHC99Xc

Methods & tools

- *Cardsort* helps UX researchers quickly analyse data from open card sorting exercises using hierarchical cluster analysis. This task helps to understand how people organize information, and is frequently used to develop information architectures for websites: https://pypi.org/project/cardsort/
- *CRC Cards* are a collaboration tool to find out which concepts the code needs and how these concepts should be defined: https://de.wikipedia.org/wiki/Class-Responsibility-Collaboration-Karten
- Role play workshop content for learning how to do user testing, for example through wireframing, *Illustrated Quick-Start Intro To Wireframing*: https://superbloom.design/learning/blog/wireframing-intro/
- The *System Usability Scale (SUS)* provides a “quick and dirty”, reliable tool for *measuring* the usability (measuring sounds cool, but it only makes sense if you have something to compare it to, like a previous measurement!): https://digital.gov/topics/usability/
- *The Rainbow Spreadsheet: A Collaborative Lean UX Research Tool*: https://www.smashingmagazine.com/2013/04/rainbow-spreadsheet-collaborative-ux-research-tool/
- Get in touch with why you're doing this via *Empathy map:* https://web.archive.org/web/20160711191804/http://thetoolkitproject.com/tool/empathy-map
- *EventStorming*: Post it workshops for software development: https://www.eventstorming.com
- A method for beta testing documentation: *the highlighter test:* https://userresearch.blog.gov.uk/2014/09/02/a-simple-technique-for-evaluating-content/
- Writing *unit tests* can also help to divide your code in meaningful parts (Since big thing where everything ins somehow connected is hard to test:

https://en.wikipedia.org/wiki/Unit_testing
    - Caveat: Some parts are easier to test, usually the ones that are not related to the user interface. Keep using tracer bullet dev!
- For making it possible to change the UI while keeping the rest constant or vice versa: Use the *command pattern*: https://refactoring.guru/design-patterns/command

Sourcing testing users

- You can get free access to an otherwise expensive suite of user testing tools through UserTesting.com's social impact program, *OneWorld*:

  https://www.usertesting.com/company/programs-initiatives/usertesting-oneworld
    - (Note for usertesting.com (and similar products): You won't get a lot of people from marginalized communities e.g. because payment needs a bank account or credit card etc.)
- "Citizen User Testing" groups concept, a formal model to identify more diverse feedback from communities you want to serve w/ your product or service- Examples and free e-Thema on the method: https://directory.civictech.guide/listing-category/user-research-groups
- The Mom Test - frequently cited Thema on this, essentially people will lie to you to be nice, how to talk to customers to get honest feedback (note: as a mom, I really hate this name. - Molly): https://www.momtestbook.com/

□□ Software intuitiv gestalten Podlove Publisher Onboarding & Import Assistant

Das Team hinter Podlove Publisher Onboarding & Import Assistant (Prototype Fund Runde 15) spricht darüber, wie man Software entwickeln kann, die User*innen gut bedienen und gern nutzen - und das (natürlich) im Podcast:

Design und User Research mit Podlove

□□ Usability und Zielgruppenfokus in FOSS-Projekten — Evaluation der Prototype Fund Runden 15 & 16

„Seamlessness“, also die möglichst reibungslose Nutzungserfahrung, ist im UX Design ein Kernziel für Apps und Websites, aber auch für Softwarebausteine. Der Erfolg vieler dominanter, profitorientierter Anwendungen liegt oft auch in bequemen Voreinstellungen und dem Fokus darauf, möglichst viele Nutzer*innen zu erreichen, nicht zuletzt weil die Profitorientierung dazu einen Anreiz setzt. Freier- und Open-Source-Software (FOSS) hingegen haftet die verbreitete Annahme an, dass Design, Usability und Zielgruppenorientierung nur eine untergeordnete Rolle spielen — kein Schnickschnack, aber eben auch kein Fokus auf die Nutzungserfahrung. Die vermuteten Gründe dafür sind vielseitig. In unserem Evaluationsbericht sind wir deshalb der Frage nachgegangen, welche Rolle die Nutzendenfreundlichkeit und das Erreichen der eigenen Zielgruppen in den vom Prototype Fund geförderten Projekten spielen.

In diesem Blogpost bekommt Ihr einen Einblick, wie der Status Quo zu Usability und Zielgruppenerreichung in den Förderrunden 15 und 16 aussieht (1), welche möglichen Gründe dahinter stecken (2) und wie der Prototype Fund versuchen kann, die Projekte noch besser zu unterstützen (3).

Eine genauere Beschreibung der Ergebnisse findet ihr im vollständigen Evaluationsbericht.

1 Wie sind Geförderte in den Bereichen Usability und

Zielgruppenerreichung aktiv?

Um mehr zu den Aktivitäten zu Usability und Zielgruppenerreichung in den Förderrunden 15 und 16 zu erfahren, haben wir eine Online-Umfrage gestartet und dreizehn Interviews mit Geförderten geführt.

Es zeigt sich ein durchmischtes Bild, wie die Projekte in den beiden Bereichen aktiv sind: Während manche im regen und direkten Austausch mit ihrer Nutzendengruppe sind, holen andere während der Förderphase kein Feedback ein. Auch die Methoden um die Bedürfnisse von Nutzenden in Erfahrung zu bringen unterscheiden sich stark und reichen von informellen Gesprächen bis hin zum klassischen User Testing. Der Grad der Struktur variiert ebenfalls deutlich; viele Projekte agieren in der Förderphase jedoch eher intuitiv als methodisch strukturiert.

Ähnlich vielseitig sind die Kanäle, über die Feedback eingeholt wird und über die Nutzende sich melden können: Zur Anwendung kommen E-Mails, Kontaktformulare sowie die für FOSS-Projekte üblichen Issue-Tracker in Code-Repositorien, aber auch Soziale Medien, Foren, Chat-Channels und Community-Treffen.
In den Interviews zeigt sich weiter, dass einerseits ein Drittel der Befragten (noch) keine Zielgruppe erreicht. Andererseits werden zwei Drittel der Prototypen von Testanwender*innen der Organisation, in der das Projekt entwickelt wurde, oder von Dritten bereits genutzt.

2 Mögliche Gründe für die Herangehensweise an Nutzendenfreundlichkeit und Zielgruppenerreichung

Dafür, dass in der Förderzeit eher intuitiv und mit weniger Nachdruck mit potenziellen Nutzer*innen gearbeitet wird, lassen sich unterschiedliche mögliche Gründe heranziehen:

Nicht-wirtschaftliche Motivation

Wirtschaftliche Interessen scheinen unter den Befragten eine untergeordnete Rolle zu spielen. Eine zahlende und möglichst große Nutzendengruppe für das eigene Projekt zu gewinnen, verliert dadurch an direkter Notwendigkeit im Vergleich zu proprietären Anwendungen.

Erwartungen an die Zielgruppe

Einige der Projekte der beiden Förderrunden sind besonders für Entwickler*innen von Interesse. Unter Geförderten besteht der Wunsch, dass sich die Zielgruppe bzw. die Community mit einbringt. So würden diese mit Konfigurationen nach ihren Bedürfnissen die Nutzendenfreundlichkeit in den Projekten mit vorantreiben.

Weniger Vorerfahrung mit Design und Öffentlichkeitsarbeit

Ihre eigenen praktischen Erkenntnisse in den Bereichen Design, Usability sowie Kommunikation und Öffentlichkeitsarbeit schätzen die Geförderten als eher gering ein. Diese Verteilung kann, aufgrund von fehlender Vorerfahrung, zu weniger Aktivität in den beiden Fokusbereichen geführt haben.

Begrenzte Ressourcen

Maßnahmen wie extensive A/B-Tests sind teuer; FOSS-Projekte leiden notorisch unter Ressourcenknappheit. Der Prototype Fund setzt durch die vorrangige Ansprache von Softwareentwickler*innen und kurze Förderzeiträume ebenfalls eine enge Grenze für die Projektentwicklung. Das kann dazu führen, dass für bestimmte Aufgaben wenig Kapazitäten vorhanden sind. Zudem ist es möglich, dass in Projektphasen, die nicht (mehr) in den Förderzeitraum fallen, mehr Aktivitäten in den beiden Bereichen stattfinden.

3 Wie unterstützt der Prototype Fund in Punkto Nutzendenenfokus und was kann in Zukunft noch verbessert werden?

Der Prototype Fund unterstützt Geförderte in beiden Bereichen bereits durch individuelle Coachings, verschiedene Online-Sprechstunden zu den Themen User Testing, Community Building und Branding sowie einer Knowledge Base. Auf Basis der Umfrageergebnisse sehen wir sowohl in der Bewerbendenauswahl als auch in der Förderphase Stellschrauben:

Auswahl der Bewerbenden

In der Bewerbendenauswahl kann ein größeres Augenmerk darauf gelegt werden, dass Kompetenzen, die benötigt werden um nutzendenfreundliche Software zu entwickeln und die richtigen Zielgruppen zu erreichen, in den Projekten stärker vertreten sind. Zudem kann in der Bewertung von Bewerbungen mitberücksichtigt werden, dass Projekte beabsichtigen, breitere Nutzendengruppen zu erreichen.

Unterstützung während der Förderung

Inhaltlich kann ein noch größeres Gewicht auf die fachliche Unterstützung gelegt werden. Geförderte können beispielsweise dabei unterstützt werden, ihre Erwartungen an ihre Zielgruppen zu schärfen und gegebenenfalls unrealistische Wünsche, z. B. in Punkto Community-Beiträge, anzupassen. Zusätzlich können Ressourcen, die explizit für weniger technische Aufgaben bereitgestellt werden, die Priorisierung von Nutzenden und Zielgruppenerreichung vorantreiben.

4 Fazit

Generell spielen Nutzendenfreundlichkeit und Zielgruppenerreichung in den geförderten Projekten eine Rolle. Das Gesamtbild ist jedoch sehr heterogen — wie es auch die Projekte der beiden Runden und deren Zielgruppen sind.

Wenn die Rolle der beiden Felder auch weniger zentral ist, als beispielsweise in streng profitorientierten Kontexten, sind Usability und Zielgruppenerreichung dennoch nicht zu vernachlässigen. Damit FOSS-Projekte einen möglichst großen Effekt für möglichst viele Nutzende entfalten können, sind Nutzendenfreundlichkeit und Zielgruppenerreichung zentral. Deshalb sollte beiden Feldern sowohl in der Projektplanung als auch in der Förderung entsprechende Aufmerksamkeit zukommen.