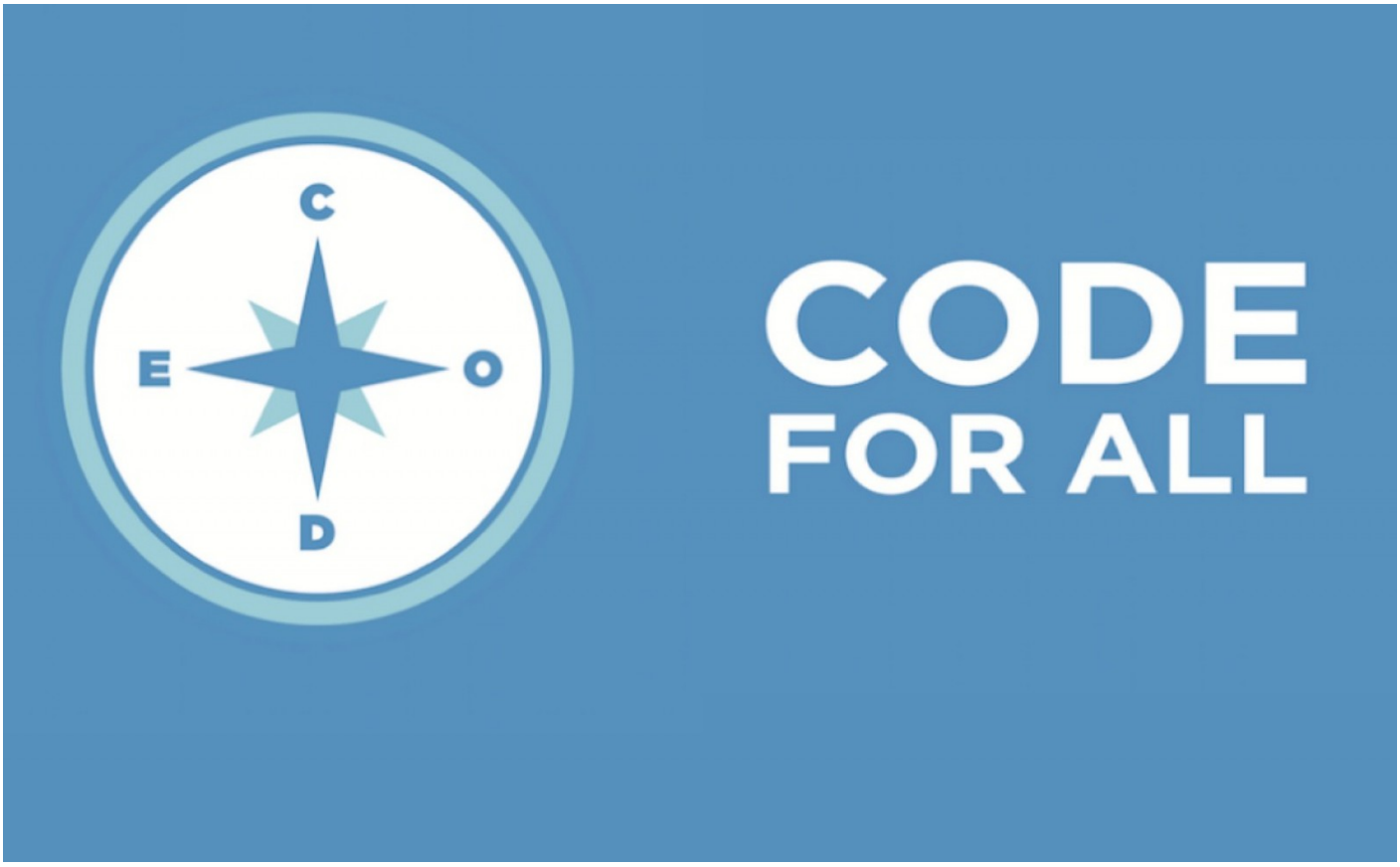
Communityaufbau und -zusammenhalt

Communityaufbau, Organisation, Entscheidungen, Communities und OSS, Umgangston und Soziales

- How to manage and engage with large communities - A Checklist by Lorin Camargo and Mar Marín of Code for All
- Software for and by communities - interview with Dark Crystal Social Key Recovery System / CoBox
- Interview mit Tobias Zwick (OpenStreetMap)
- Link collection community building and management
- Maxi Richt und Benjamin Seibel zu Förderung, Communities und digitalem Ehrenamt
- Kommunikationsstrategien für Community-Building und -Management
- Gründung eines (gemeinnützigen) Vereins als FOSS-Projekt? Eine Übersicht von Tweasel for Web
- Communities für Prototypen – Evaluation der 13. und 14. Förderrunde

How to manage and engag with large communities - A Checklist by Lorin Camargo and Mar Marín of Code for All



Code for All is an international network of civic tech organizations. Our mission is to connect these organizations so they can learn from each other, share resources, and scale projects together, leading to greater impact in their local communities.

Lorin: I started off my civic tech journey volunteering for Code for America's Brigade, Code for San José, and am currently one of Code for All's Co-Directors. As a Co-Director, I help shape communication, team, organizational, and fundraising strategy for the Network.

Mar: I began my journey doing professional practices with Codeando México. Following my interest to understand the intersecting relationship between political sciences and technologies, while implementing storytelling techniques, I got the opportunity to join Code for All as their Communications Assistant.

How do you manage to bring all participants together in such a large community?

1. **Have a central place for your community to meet asynchronously** to share their work, discuss challenges, promote opportunities, and to connect about non-work stuff (it's always good to have some fun channels Slack is our central meeting hub; it allows space for Code for All community members to connect with each other on their own time, and it's also a place to find ample civic tech news, events and opportunities.
2. **Bring people together in at least one consistent, recurring way**. We host an annual Summit, which aims to bring the global community together and allow space for showcasing projects, sharing knowledge and workshopping ideas. Along with all the other events we host, I find the Summit to be very important because it's consistent. It's something the community can rely on each year as a way to connect and share work.
3. **Be aware of and accommodating to diverse needs**. We spend a lot of time thinking about inclusivity and accessibility and always strive to create welcoming, safe spaces to bring people together. We often ask for feedback after events, or when we design just about anything, to make sure we are taking our community's needs into consideration. It's important to us that accessibility is something that's built into the things we do, versus being an afterthought. We're nowhere near perfect in this realm, but it's a priority for us to always be improving.
4. **Be aware of and accommodating to different time zones**. Being a global network, we spend a good deal of time working around time zones and trying to accommodate everyone. We aren't always able to make events work for everyone, but we often host multiple sessions of the same event in order to account for different time zones. We also record some events for post-watching, and like to create documents with concise, key take-away points so folks who miss sessions can quickly catch up without having to pour through notes that haven't been edited down.
5. **Use what you have**. Our member organizations all have extensive communities and networks of their own. When we have something we want to spread throughout the global civic tech community, we always share the information with our members and ask them to pass it along to their communities. For each of our member organizations, we have at least one representative who acts as a point of contact between us and the organization. Having this point of contact makes it easy for us to share information and promote opportunities with entire communities in just one quick step.

What are useful methods when engaging with a large community?

The Code for All community is both large and international, so one big thing we keep in mind is the language we use. We haven't yet had the resources to offer all of our materials in several

languages, and we primarily use English for communication, however we try to avoid using complicated jargon, so that our articles, opportunities, and resources are as easy-to-understand as possible for a varied audience. Similarly, we've heard from our member organization g0v, that in order to engage their large community, they often create multiple versions of communication material for any given event or opportunity, each version targeted to a different type of community member (for example, different age groups or professions, or reaching out to a student, or parents of students). You can find some brief key take-aways from a discussion around this here.

How do you deal with conflicts within the community?

We have a Code of Conduct which explains what is expected from network activities, events, and digital forums. Luckily, we haven't had many conflicts within the community, but depending on the severity of the conflict it is something we would either deal with as the Core Team, or would consult Governing Members of the network. In many cases, it would be important for us to try to understand all perspectives involved before solving the conflict.

How do you do outreach? What are your methods when acquiring new community members?

1. **Annual Summit:** Our annual summit is an amazing opportunity to showcase what our member organizations, volunteers and civic tech organizations around the world are doing and draw attention towards Civic Tech and the amazing projects around the world. The themes are always voted in, and are based on the interests of the community. The event has proven to be an occasion for organizations to gain interest in joining the Network.
2. **Communities of Practice (CoPs)**: These groups are about bringing network members, and organizations beyond the network, together to form groups and discuss common topics relevant to the civic tech ecosystem. This has proven to be an amazing opportunity for experts and people interested in connecting with other organizations working on similar topics to discuss best practices around civic tech projects.
3. **Events**: This year we have been working on developing strong relations with key stakeholders that can draw attention to the intersection between Civic Tech and other relevant fields. We began the **Civic Tech & SDG Webinar Series**, an initiative that was born as our members have been advocating for data-driven development for years. To voice their work and collaborations that put the power of data behind delivering the Sustainable Development Goals, we have designed and organized the webinar series with key stakeholders in international and regional organizations.
4. **Social Media Strategy:** We have noticed people like to know what's going on with our member organizations and also want to know more about civic tech in general. We try to share updates from the global civic tech community in engaging, fun and easy-to-understand formats. Understanding our audience is an important step in forming our social media strategies. For example, as a generalization, many civic techies are fans of certain branches of popular culture, and have fun interacting in conversations that use engaging themes such as Star Wars (here's a quick example).

Do you recommend specific tools that help the community in collaborating (online and offline)?

- **Slack**: A mighty collaboration tool that Code for All has been using since its inception.
- **Trello**: To organize projects, share information or conduct group discussions. For example, we sometimes use Trello mixed with a Zoom call to conduct our bi-monthly retrospectives.
- **Miro/Jamboard**: Both are great online whiteboards to brainstorm collaboratively.
- **Notion**: My personal favorite. It is an all-in-one collaborative application that brings together, on a single platform: documents, databases, public websites, knowledge bases, project management systems and beautiful notes.
- **Zoom**: For synchronous meetings and events.
- **Google Drive**: For generating and storing documents.

Do you have advice for the cooperation between volunteers and paid staff?

We don't have a lot of advice on this front from our team, but we recently held a small group discussion around this in one of our Support Squad meetings, where a volunteer from Code for Romania shared their thoughts. You can watch the recording of that discussion here (in English).

How much self-determination do individual community members have (or how much is controlled centrally?)

Code for All is here to help connect civic tech organizations and individuals, and provide resources and opportunities to help their work. Our member organizations work completely independently from us, and have absolute self-determination in their work. We do not influence or control any aspect of their independent work, but act as a source of support to help advance, scale or replicate their projects.

You also work on the fundraising side of things - are there any insights from that part of your work that could help other people who try and build a sustainable civic tech/public interest tech project?

We have an article that covers some highlights around this: Challenges & Tips for Funding Civic Tech. We also held a sustainability workshop recently for some of our members, and the insights they learned are shared here.

□□ Software for and by communities - interview with Dark Crystal Social Key Recovery System / CoBox

Mu and Peg worked on the projects Dark Crystal Social Key Recovery System and CoBox. In these and other projects they are working with a unique collaborative approach and a strong focus on community topics. They are sharing their knowledge and experiences on the topic in this interview.

> *In your projects - CoBox and Dark Crystal Social Key Recovery System - one can observe two different approaches towards the topic of community, a) the way the software is functioning because both projects are distributed software, i. e. they rely on a group of people using them; b) the developers form a collective/cooperative. Therefore, it seems like community is very important to you. How did you come up with the community approach for your software? Do you feel like distributed backups and collaborative clouds are more secure and otherwise advantageous - or is there even more to it?*

There's a lot to it! Of course we are organised as a collective - which is a big part of it - but we will share more about that below. In terms of the software itself, the community approach isn't valuable to us along a single vector - there are multiple dimensions to it.

For Dark Crystal, the project has its roots in the needs of a very concrete community, around the 'Secure Scuttlebutt' social network. Many 'peers' (as they are called within the community, to reflect the p2p architecture of the protocol) had expressed frustration when losing access to their account. So we designed a social recovery system through open discussion on that platform. Even at the very early design stages there was a lot of input from the people who would later use it.

In terms of whether the decentralised architecture of the recovery system made it more secure, yes we would argue that distributing responsibility across a trusted group has security advantages over the traditional consumer-provider model. But we think the co-design / community-centered approach to software can be very useful regardless of what architecture you decide on.

For CoBox, the community aspects were multiple and resulted from a coincidence of threads coming together around an architectural idea. All members of the team involved in developing the project idea had long histories in the European cooperative and collective action movements and

had experienced first-hand the frustration of there being few values-aligned digital tools available, forcing groups to rely on infrastructure that was in contradiction with their lived values.

At the time of its conception there was also increasing public awareness around surveillance capitalism and the personal, political and environmental dangers of massive centralised digital infrastructure, and therefore more tech funding was being directed towards supporting the underlying protocols on top of which such an alternative digital infrastructure could be built. Having stumbled across DAT, which was one of these underlying protocols, we decided to build on top of it to create the infrastructure we felt was missing from the communities we participated in. To do so, we deliberately set out to engage this community in the process of collaborative design.

So as we have come to understand it, 'community' can manifest in a few different ways. It can manifest infrastructurally in the software, socially in the team, and also in the input that spans these two components, so that both Dark Crystal and CoBox were developed in collaboration with the people they were being built for.

A fourth manifestation of community that we haven't discussed above, but will do below, is the community of developers who are building complementary digital infrastructures, to address parallel needs and based upon similar values. This has been where Dark Crystal ultimately found the most traction, as while the UX for distributed key backup came directly from the participation of end-users, it is other tech projects, needing to integrate a key-backup feature into their applications, who have become the primary consumers of this tool.

> *For a lot of users your approach might be unusual, too. How do those get involved, how do you “target” them? Is there for instance some sort of critical mass of users that you need in order for the community to grow from within, i.e. without you having to actively acquire new people?*

We are acutely aware of the difference between the information available to big corporations for determining preferences and the tools they have to collect usage-data vs the capacities that small community projects have to do so. It is therefore significantly easier for a big corporation to create an 'ideal' product without requiring much conscious participation from any of their users (through, for example, mass A/B testing and the like). As a small team working on a project and attempting to use tools and practices that align with a particular set of values, it is much more important to speak with people directly, to find out what they want and need, and often these interviews and testing sessions reveal ideas and perspectives that had never occurred to us before.

In terms of finding users to help us design the product, in the case of CoBox (in which we more actively had to 'target' people) we used the channels that the cooperative movements we were involved with had already set-up. We posted on forums, spoke at a community event and contacted people we personally knew from within that milieu. However, we are aware that unlike the data that corporates are able to amass, our process requires a commitment of time from end-users, who are all busy with their own projects and work and community building activities. We have until now always managed to compensate users for their participation, and we make an effort to plan the sessions with an eye towards efficiency and respect.

If the tools being built accurately identify a need and solve the problem in an effective way then what we have seen is that a kind of network effect develops off the back of that. Because of the open-source nature of development, projects are able to tap into each others' work, rather than a particular feature being siloed into a single application as a 'USP' over the offer of a competing app. This is what happened with Dark Crystal, where we are still being approached by projects that have heard about the key backup tool through word of mouth or shared communication channels and would like to integrate it into their application for the benefit of their users.

So yes, there comes a point when the community around a project exists without needing much active publicity. But rather than a critical number of users, it is probably a critical level of interest - you can have a small number of people who are very enthusiastic or actively involved, and it is enough to build a project around.

> *How do you organize your work as a cooperative/collective? Is it very different from a traditional firm? How did you come up with it and why did you decide to organize your group in that way?*

Our decision to work this way is rooted in our belief that work can and should be fulfilling, rather than alienating and exploitative. And in our understanding (which is ever adapting and developing) this derives from a feeling of shared ownership, agency, belonging and mutual respect between individuals who are standing on level ground.

Workplace dynamics can be really debilitating when there is no possibility to feel heard and engaged in your work, or when that work conflicts too sharply with your personal values and with what you would like to see exist in the world. At least one aspect of which is the case, as far as we know, in many workplaces, technical or not. Of course we recognise that especially among tech workers there can be some appeal to forgoing the above for work that is extremely well compensated. But such work can often also conflict with other personal and social values, making the compromise even less appealing in sum.

Internally, we try to organise in a way that there are not big power imbalances in terms of making decisions or accessing resources like servers or the bank account. There are, of course, many complexities to working this way. One challenge for a group that works like this is the process by which someone joins or leaves the group. When someone who is helping on a particular project becomes more involved, there needs to be some formal process which makes it clear at what point they become a member.

> *Maybe you could also tell us a bit about your approach to design and what your decision process looks like in that area. E. g., on the CoBox website you write about the peer-design process - what is that?*

Big corporations can do massive and continuous user-testing, unbeknownst to the user. Sometimes this is ugly and manipulative (for example, Facebook's psychological impact testing), but it doesn't have to be. When you do not have the capacity to do this kind of research, you have to rely on smaller and more personal processes. This is emphasised in the start-up community as well, where the available resources are still grand in comparison to what we have access to, but much smaller

than what established companies have access to. The co-design process has a lot of benefits. It not only allows you to design and create a product that the people you are building for actually want - taking into account their more subtle and difficult to identify needs, but it also allows you to very consciously choose to prioritise particular communities who may be underserved by other products available.

There are of course also challenges with co-design and user-testing. For example, when you are developing a new architecture, and developing a UX for it, the fact that it is quite far removed from what is already familiar, comfortable to and expected by people can be a challenge. The UX-design coaching that Dark Crystal received from Simply Secure as part of our funding was really helpful towards addressing this challenge, because distributed key backup or recovery involves some concepts which are quite unconventional. It was therefore difficult for us to find terminology and UI patterns that were not too confusing and the design coaching helped a lot with that.

Another challenge that we faced is that sometimes what users want doesn't align at all with what you want, or what you thought they would want. This happened to us during the second development phase of CoBox. After our first release, we had a clear roadmap that we thought prioritised what users would need, in order to begin using the tool. Our co-design interviews revealed that this was not the main or most important barrier to usage at all, and an issue that we had completely overlooked was far more prominent in their decision-making process about when or whether to begin using the app. Fortunately we were able to pivot in this instance, to prioritise addressing the issue that they raised.

□□ Interview mit Tobias Zwick (OpenStreetMap)

Tobias Zwick ist der Autor von StreetComplete, einer Android App, mit der man ohne weitere Kenntnisse direkt etwas zur OpenStreetMap (OSM) beitragen kann. Dank der Förderung durch den Prototype Fund (Runde 8) konnte er die Entwicklung 2020/2021 intensivieren. Mittlerweile wird die App von mehr als 22.000 Menschen genutzt, was sie nach Anzahl der Nutzer*innen zurzeit zum zweitbeliebtesten Editor für OpenStreetMap überhaupt macht. Auch erhält das Projekt schon von Beginn an (seit 2017) zahlreiche Beiträge von Mitwirkenden.

> *Was, denkst du, zieht Leute in die OSM Community? Und warst du vorher selbst in der OSM Community aktiv?*

Was den Zugang zu freier Information angeht, ist OpenStreetMap sicherlich ähnlich wie Wikipedia. Allein der Gedanke, hoffentlich hilfreiche Informationen für andere Menschen frei zur Verfügung zu stellen, ist sicherlich für viele Motivation genug.

Mir macht es zudem Freude, die weißen Flecken auf der Karte zu füllen. Insbesondere wenn man vor Ort ist, fühlt man sich dabei regelrecht wie ein Entdecker. Gleichzeitig ist es motivierend, das Resultat meiner Arbeit hinterher direkt auf der Karte zu sehen - man hat etwas geschafft! Ich hatte darüber 2013 in meinem damaligen Reiseblog geschrieben.

Letztlich ist es auch ein netter und zumeist wenig fordernder Zeitvertreib, denn um zur OpenStreetMap beizutragen muss man kein*e Expert*in sein.

> *Wie bist du bei der Einbindung der Community vorgegangen? Reicht der Anschluss an OSM oder bist du selbst darüber hinaus aktiv geworden, um Menschen zu erreichen?*

Vor-Ort Informationen beizutragen war schon immer recht aufwändig und benötigt technisches Verständnis zu OSM. Die App ändert das. Das heißt, der initiale Erfolg der App, überhaupt wesentlich genutzt zu werden, ist mir dadurch, dass ich zufällig den richtigen Riecher hatte, was dem Ökosystem fehlt, quasi in den Schoß gefallen.

Dieses Projekt ist insofern speziell, als dass es nicht für sich allein steht oder stehen will, sondern sich in eine größere Community integriert. Potenzielle Nutzer*innen gibt es also viele. Das schränkt jedoch ein, was man machen kann - und was nicht. Die App muss sich der bestehenden Community unterordnen, um akzeptiert zu werden und nicht auf den Prüfstand zu kommen. Das musste ich von Anfang an beachten.

Damit die App und die Beitragenden in der Community nicht als Fremdkörper aufgefasst werden, war es allein schon wichtig, die OpenStreetMap-Community von Anfang an einzubinden. Dazu gehört unter anderem eine Präsenz in verschiedenen Community-Channels, eine ausführliche Beschreibung der App im Wiki mit Unterseiten, damit auch für Leute, die die App nicht nutzen, nachvollziehbar bleibt, was mit dieser App getan wird.

Die Einbindung in die Entwicklung, die meiner Erfahrung nach am besten und als erstes funktioniert, ist Crowdsourcing der Übersetzung. Dank einem dezenten Hinweis auf der Projektseite und der verlinkten Wikiseite wurde die App bisher in 46 Sprachen übersetzt - zu 100 % von Beitragenden.

Letztlich hat die App aber auch neue Zielgruppen erreicht, die bisher noch wenig mit OpenStreetMap am Hut hatten: Leute, denen das Beitragen bisher zu technisch oder zu aufwändig erschien oder auch Leute, die sonst eventuell eher Pokemon Go oder Geocaching als Zeitvertreib betrieben hätten.

Diese neuen Gruppen zukünftig (mehr) in die OpenStreetMap-Community im Ganzen zu integrieren, würde sowohl die OpenStreetMap Community bereichern als auch die Akzeptanz für die App und ihren Anwendungsfall weiter steigern.

Hast du Tipps für Leute, die mit ihrem Projekt eine Community erreichen wollen?

Ja, habe ich. Aber ein Disclaimer: Was für mich funktioniert hat, funktioniert vielleicht nicht für Andere. Also, Ratschläge immer mit Vorsicht genießen!

Wie schon erwähnt ist StreetComplete etwas speziell, weil es sich in eine größere Community integriert. Dadurch wurden gleich zu Anfang viele Leute erreicht. Sich einer bestehenden Community anzuschließen oder daraus hervorzugehen ist möglicherweise auch für andere Projekte ein hilfreicher Ansatz.

Die erste und größte Hürde ist natürlich aber erst einmal, ein Produkt zu entwickeln, das auch wirklich genutzt wird. Eine Idee kann ein Hit oder ein Miss sein. Generell hat ein Projekt, das ein bestehendes erfolgreiches Produkt erweitert oder ergänzt, natürlich bessere Chancen, genutzt zu werden.

Daher - ganz im Sinne des Prototype Funds - ist es wichtig zuerst ein Minimum-Viable-Produkt (MVP) zu erstellen und dieses publik zu machen. Nur wenn das auf positives Feedback stößt, lohnt es sich, die Idee weiterzuverfolgen. Wichtig ist, dass der Umfang (Scope) des Produkts ganz klar auf das Minimum abgegrenzt ist und in sich geschlossen ist. **Ein Produkt, das unfertig wirkt, will niemand benutzen**.

Deswegen dokumentiere öffentlich, was Ziel des Projekts ist, was zum Scope gehört - und was nicht, sodass potenzielle Beitragende wissen, woran sie sind (Beispiel in StreetComplete). Was ist erwünscht und was nicht? Eine öffentliche Dokumentation hilft auch, selbst in den selbst gesetzten Schranken zu bleiben, also Feature-Creep und mehrere gleichzeitige Baustellen zu vermeiden. Meiner Erfahrung nach ist es am besten, wenn das Projekt immer in sich geschlossen und "fertig"

wirkt. Baustellen wirken auf Dauer abschreckend für Nutzende und auch Mitwirkende.

Wenn es konkrete Dinge gibt, die Beitragende tun können um dem Projekt zu helfen, dann schreibe auch das auf. Die Energie von (potenziellen) Beitragenden muss gelenkt werden: Sie müssen wissen, was sie konkret tun können (als Beispiel CONTRIBUTING.md in StreetComplete). Sonst ist die Hürde viel zu hoch.

Wenn es in deinem Projekt einen öffentlichen Issue-Tracker (z. B. GitHub) gibt, könntest du auch Labels einführen, mit denen du bestimmte Tickets markieren kannst, z. B. "needs PR", "help wanted", "good first contribution" usw.

Das ist alles, was mir an konkreten Tipps einfällt. Es gibt noch einige weitere Richtlinien, an die ich mich halte, wie zum Beispiel leicht lesbaren aber modernen und gut dokumentierten Code zu schreiben, damit mögliche Beitragende einen einfacheren Einstieg haben. Aber ob das nun wirklich entscheidend ist, weiß ich auch nicht.

Die oben genannte Dokumentation auf der anderen Seite hat garantiert einen Unterschied gemacht (am Anfang des Projektes fehlte diese nämlich und das hat sich bemerkbar gemacht). Es ist eben wichtig, dass die Entscheidungen, die man als Leiter in einem öffentlichen Projekt fällt, transparent und nachvollziehbar bleiben. Man kann nicht allen alles recht machen, aber versuchen zu erreichen, dass sich niemand verprellt fühlt und alle, die etwas beitragen, ein gutes Gefühl dabei haben.

> *Ist die Einbindung in die Community ausreichend, um StreetComplete langfristig zu halten oder braucht es noch mehr Ressourcen?*

Tja, das langfristige Warten von reinen FOSS-Projekten ohne Geschäftsmodell ist natürlich immer so eine Sache. Es hilft, bei der Entwicklung darauf zu achten, dass das Projekt möglichst wartungsfrei durch Automatisierung und Nutzung von stabilen Frameworks überleben kann. Wenn das Projekt genug genutzt wird, finden sich mit der Zeit ggf. auch Mitstreiter*innen, die dabei helfen, das Projekt aktuell zu halten.

Letzteres ist bei StreetComplete der Fall, allerdings nützt das eher der Erweiterung des Projektes um weitere Features als der Wartung an sich.

Außerdem sind sowohl OpenStreetMap als auch Android als Ökosystem sehr instabile Plattformen, sodass der Wartungsaufwand sehr hoch ausfällt. Instabil ist Android in dem Sinne, dass sich u. a. die empfohlenen Frameworks ändern, sie teilweise ganz und gar als veraltet markiert werden, weil sich das Verhalten von Android von Version zu Version ändert und weil die App den sich stetig ändernden (und immer strenger) ausgelegten Richtlinien des Play Stores entsprechen muss.

OpenStreetMap ist außerdem instabil in dem Sinne, dass der Konsens darüber, wie etwas richtig einzutragen ist, sich ständig wandelt und erweitert, weswegen die Logik entsprechend ständig angepasst oder zumindest in Diskussionen verteidigt werden muss.

Dennoch würde ich sagen, rein die Wartung der App ist dank der vielen Spenden die ich über Liberapay, Patreon und Github Sponsors bekomme, gesichert, selbst wenn ich keine Freizeit mehr in dieses Projekt stecken kann.

□□ Link collection community building and management

To read, listen & watch

- *Building a sustainable open source community: training and certifications*: https://www.linuxfoundation.org/blog/building-a-sustainable-open-source-community-training-and-certifications/
- Linux Foundation, *Participating in Open Source Communities*, https://www.linuxfoundation.org/tools/participating-in-open-source-communities/
- *Coaching for Open Source Communities 2.0:* https://medium.com/@leslielauralive/coaching-for-open-source-communities-2-0-b29b84a43322 and https://archive.fosdem.org/2019/schedule/event/community_coaching_open_source_communities_2/
- #MoreThanCode's tips on community building: https://morethancode.cc/report/executivesummary/
- Open Source Guide, *Building Welcoming Communities*: https://opensource.guide/building-community/

Methods & tools

- Some people want just the basic infos, others are interested in more details. Writing/Communicating in an *inverted pyramid-style* helps: Give the most basic information first, put details in later sections that can be easily ignored if there is no strong interest in them: https://en.wikipedia.org/wiki/Inverted_pyramid_(journalism)
- The *Wheel of Successfuly Practice* gives an overview of practices that help to grow and maintain communities: https://preparecenter.org/resource/valuing-what-works-success-factors-in-disaster-preparedness/
- Read up on *Legitimate peripheral participation* to understand how newcomers can become part of the community: https://en.wikipedia.org/wiki/Legitimate_peripheral_participation and https://www.fordes.de/posts/LegitimatePeripheralParticipationWikipedia.html
- Automatization tools for communication:

- MailerLite for newsletters - reasonable free tier http://mailerlite.com
- Buttondown for newsletters - automations and a generous free plan https://buttondown.com
- Mattermost, discord, slack, whatsapp groups, signal, Matrix (Some channels are more or less attractive to some communities – discord is a bit more gamer-y, matrix more for devs etc.)
- List from the Civic Field Tech Guide with tools: https://directory.civictech.guide/listing-category/internal-communication-and-coordination

For networking opportunites take also a look at our link collection on project presentation and communication: https://kb.prototypefund.de/Themen/prasentation-und-kommunikation/page/link-collection-project-presentation-and-communication

□□ Maxi Richt und Benjamin Seibel zu Förderung, Communities und digitalem Ehrenamt

In der dritten Episode des Public Interest Podcast geht es um die strukturelle Seite der Public-Interest-Tech-Entwicklung, also um Förderung, Communities und digitales Ehrenamt.

Neben einem Input zum Thema sprechen wir mit Maxi Richt aus der Code-for-Germany-Community über seine Erfahrungen in der digitalen Zivilgesellschaft. Außerdem reden wir mit Benjamin Seibel vom CityLAB Berlin darüber, was nachhaltige Förderung und eine größere Akzeptanz von Open Source bewirken können.

□□ Kommunikationsstrategien für Community-Building und -Management

*Gute Kommunikation ist das A und O, wenn ihr Nutzer*innen erreichen und eine Entwickler*innen-Community rund um euer FOSS-Projekt aufbauen wollt. Personen, die die Software nutzen möchten, müssen verstehen, wie sie funktioniert. Und Personen, die zur Entwicklung beitragen wollen, müssen wissen, ob das gewünscht ist, wie das geht und was gebraucht wird.*

Im Folgenden findet ihr eine Reihe von Kommunikationsstrategien, mit denen ihr anderen helft, eure FOSS-Projekte sicher und langfristig zu nutzen, zur Entwicklung beizutragen oder da weiterzumachen, wo ihr aufgehört habt.

Von Anfang an:

Gerade, wenn ihr vorhabt, eine Community rund um euer Projekt aufzubauen und miteinzubeziehen, kann es vorteilhaft sein, den Code noch während der Entwicklung zu veröffentlichen. Zum einen ist es nämlich ganz im Sinne des Open-Source-Gedankens, dass Versuche und Lernprozesse offen und gemeinsam mit anderen geschehen. Zum anderen kann ein aktives Repository mehr Aufmerksamkeit für euer Projekt generieren, da es guten Einblick in eure Projektziele und Entwicklungsschritte gewährt. So stehen mehr Infos für potenzielle Mitwirkende bereit. Generell kann es sogar für Dritte attraktiver sein mitzumachen, wenn das Projekt noch in der Entwicklung ist und sich viel mehr Mitgestaltungsmöglichkeiten bieten, als wenn nur noch Maintenance ansteht.

Oft wird die Sorge vor Kritik am Code und damit am Projekt als Grund genannt, mit der Veröffentlichung bis zu einem bestimmten Meilenstein zu warten. Dabei sind Hinweise auf mögliche Probleme oder Bugs vielfach konstruktiv und durchaus hilfreich: Wer weiß, wie lange diese Fehler sonst unentdeckt geblieben wären.

Der Code allein ist natürlich nur die halbe Miete, wenn du Nutzende und Mitstreiter*innen gewinnen möchtest. Rund um euer Projekt existieren noch eine ganze Menge weitere Infos, die für andere hilfreich sind, um zu verstehen, worum es euch geht, wie sie euer Projekt nutzen und wo sie euch vielleicht unterstützen können. Je nachdem, was euer Ziel ist, können die folgenden Tipps hilfreich sein, Nutzenden und potenziellen Mitentwickler*innen alle Informationen zur Verfügung zu stellen, die sie für Nutzung oder Einstieg brauchen:

Informationen für Nutzende:

- Stellt Informationen in einer Form bereit, die für die Nutzenden intuitiv ist, z. B. i**n Form eines Wikis**. Die App StreetComplete hat etwa ein Wiki mithilfe der Open-Source-Software MediaWiki erstellt, die auch für Nicht-Entwickler*innen vertraut sowie leicht nutzbar ist. Zudem ist das Wiki im allgemeinen OpenStreetMap-Wiki verknüpft und ist so leicht auffindbar. (Übrigens hat Tobias von StreetComplete in einem Interview, das ihr in der Knowledge Base findet, über die Arbeit mit der Community berichtet.)
- Wenn das Projekt noch nicht ausreichend sicher ist oder wichtige Funktionen fehlen, sind **Informationen zum aktuellen Entwicklungsstand und möglicherweise auftretenden Problemen** sehr hilfreich für potenzielle Nutzer*innen. Über einen Hinweis auf eurer Webseite oder im Appstore könnt ihr Endnutzer*innen erreichen und gleichzeitig um Feedback bei Problemen bitten, wie es etwa OpenAndroidInstaller tut.
- Änderungen, Neuerungen oder Fehlerbehebungen bei der Veröffentlichung von neuen Softwareversionen lassen sich gut in Form von **Release Notes** im Repository oder auf der Projektwebseite kommunizieren und sind besonders für technisch versiertere Nutzer*innen hilfreich. Es gibt dafür keine Standards, deswegen schaut euch am besten an, wie andere Projekte solche Infos bereitstellen. Eine gute Anlaufstelle ist da zum Beispiel die Webseite von gget.

Informationen für Entwickler*innen:

- Auf **Veranstaltungen**, über **Social Media**, in **Artikeln** oder **Blogbeiträgen** über eure Projektfortschritte zu berichten, ist ein gängiger Weg, um Aufmerksamkeit für euer Projekt zu gewinnen. Auch in einem frühen Entwicklungsstadium kann sich das schon lohnen – auch wenn es vermeintlich wenig zu berichten gibt: andere Entwickler*innen können etwa wertvolles Feedback geben oder sogar Mithilfe anbieten. Wenn ihr auf Veranstaltungen unterwegs seid, kann es außerdem nützlich sein, nicht nur im Vorfeld über eure Kanäle darüber zu informieren, sondern auch danach, wie Polychat es etwa tut. So können im Nachgang Interessierte Aufzeichnungen eurer Vorträge finden und einen Eindruck gewinnen, wo man euch in Zukunft treffen kann. Unser Tipp: Überlegt euch nicht nur, wo ihr eure Zielgruppe am besten erreicht, sondern auch, welches Format zu euch passt. Manche sind für die große Bühne geboren, aber vielleicht fühlt ihr euch ja bei einem informelleren Meetup mit kurzer Pitchrunde am Anfang wohler. Event-Tipps findet ihr u. a. hier in der Knowledge Base.
- **Informationen über eure geplanten Entwicklungsschritte** können Entwickler*innen helfen, euer Projekt auch dann zu verfolgen, wenn es noch in einer recht frühen Entwicklungsphase steckt, und mögliche Anknüpfungspunkte für eine Mitarbeit zu identifizieren. Eine Checkliste lässt sich etwa gut im Repository, aber auch auf der Projektwebseite unterbringen und kann neben der Übersicht zu den Entwicklungsplänen zeitgleich auch den Fortschritt dokumentieren. Solltet ihr irgendwann an einen Punkt kommen, an dem ihr das Projekt abgeben möchtet, kann so eine Übersicht außerdem eine gute Basis für ein Übergabedokument bieten. Ein gutes Beispiel dafür bietet das Projekt PsyLink, das in der Checkliste nicht nur die reine Entwicklung der Software plant, sondern auch über die Themen Hardware, Dokumentation, Community und Marketing informiert.

- **Contributing Guidelines** geben Hinweise dazu, wie Entwickler*innen zu einem Projekt beitragen können. Bei OpenMLS seht ihr ganz gut, was sie alles beinhalten können: Informationen, welchen Stilrichtlinien Beiträge folgen sollten, wie Beiträge im Repository hinzugefügt werden können und welchen Regeln der anschließende Review-Prozess folgt. Laura von gget hat hier in der Knowledge Base außerdem Hinweise für Contributing Guidelines zusammengefasst.
- Neben Informationen, wie der technische Ablauf für Beiträge ist, kann ein **Code of Conduct** (CoC) dabei helfen, Erwartungen an die Art der Kommunikation und die Zusammenarbeit im Projekt zu erklären. Gerade wenn ein Projekt noch in einer frühen Entwicklungsphase ist und größere Mängel kritisiert werden, kann ein CoC einen Kommunikationsstandard setzen. Auch hier gibt es natürlich vielfältige Möglichkeiten, wie so ein CoC aussieht und was er alles beinhaltet. Das Team vom Projekt transcribee hat sich beispielsweise am verbreiteten Contributor Covenant orientiert und legt in seinem CoC eine Reihe von Verhaltensstandards sowie deren Durchsetzung fest.
- Direkte Beiträge zum Code sind oftmals nicht die einzige Möglichkeit, an einem Projekt mitzuarbeiten. Um die **Arbeit aller Community-Mitglieder öffentlich anzuerkennen**, ist es daher sinnvoll, sie auch als Beitragende aufzuführen. Während das auf der Webseite einfach ist, kann es je nach Plattform im Repository schon schwieriger sein. Anbieter wie GitHub legen den Fokus stark auf den technischen Teil der Software-Entwicklung, sodass nur die Entwickler*innen als Beitragende aufgelistet werden. Projekte wie UrbanAnalyst zum Beispiel haben aber auch Beitragende, die etwa durch Issues an der Weiterentwicklung mitwirken. Deshalb nutzt das Projekt den Bot „All Contributors“ für GitHub, mit dem sich zusätzliche Beitragende leicht zur Übersicht im Readme hinzufügen lassen.

Wo fange ich an?

Nicht alle Kommunikationswege, die wir hier vorstellen, sind für alle Projekte gleich wichtig. Überlegt euch, was für euch und eure Zielgruppe sinnvoll sein könnte. Dabei können euch die folgenden Fragen helfen, ein klareres Bild zu zeichnen:

- Was für eine Community habe ich oder will ich? Geht es mir eher um Endnutzer*innen oder Menschen, die aktiv an der Entwicklung des Projekts mitwirken sollen?
- Was wünsche ich mir von der Community, bzw. was kann sie zum Projekt beitragen? Das kann von der regelmäßigen Nutzung deiner Anwendung über Feedback hin zu Mitarbeit bei der Weiterentwicklung eine ganze Menge sein.
- Was braucht sie an Informationen, damit sie diesen Beitrag zum Projekt leisten kann?
- Wo kann ich sie am besten erreichen, sowohl digital – zum Beispiel über meine Webseite oder das Repository – als auch auf Veranstaltungen?

Und wenn‘s nicht so läuft wie geplant?

Dann ist gute Kommunikation umso wichtiger!

- Bei Bugs oder Sicherheitsproblemen solltet ihr **transparent auf allen Kanälen kommunizieren**, was diese sind und ggf. **vorbeugende Schritte** treffen, damit das Problem keine weiteren Schäden anrichtet. Zum Beispiel, indem ihr die App durch einen Dummy im Store ersetzt.
- Es kann viele Gründe dafür geben, dass sich das Projekt im Ganzen nicht so entwickelt wie gewünscht: weil etwa der Ansatz nicht funktioniert oder durch technische Neuerungen (bald) überholt ist. Der Code und die Erkenntnisse, die ihr bei der Entwicklung gewonnen habt, können trotzdem einen Mehrwert für andere darstellen. Vielleicht hat ja jemand eine ähnliche Idee und kann von der Vorarbeit oder euren Erfahrungen, was funktioniert und was nicht, profitieren. Die könnt ihr beispielsweise in der **Dokumentation** oder einem **Whitepaper** zusammenfassen, wie es das Projekt audilu getan hat.
- Zu guter Letzt: Gerade wenn ihr vielleicht am Projekt weiterarbeiten möchtet, aber schlichtweg die Zeit dafür fehlt, kann **gute Kommunikation und Dokumentation zum Stand des Projekts** andere motivieren, Aufgaben zu übernehmen und in das Projekt einzusteigen oder sogar ganz zu übernehmen.

□□ Gründung eines (gemeinnützigen) Vereins als FOSS-Projekt? Eine Übersicht von Tweasel for Web

Der Datenanfragen.de e. V. ist ein deutscher gemeinnütziger Verein, der es sich zur Aufgabe gemacht hat, Menschen bei der Ausübung ihres Rechts auf Datenschutz zu helfen. Wir hatten ihn 2018 gegründet, um eine rechtliche Struktur für den Betrieb der gleichnamigen Webseite zu schaffen, auf welcher Nutzer_innen Anfragen generieren können, um ihre Betroffenenrechte aus der Datenschutz-Grundverordnung zu nutzen und so Auskunft über die Daten, welche Unternehmen und andere Organisationen über sie verarbeiten, zu erhalten, falsche Daten berichtigen zu lassen, nicht mehr nötige Daten löschen zu lassen, oder dem Versand von Direktwerbung zu widersprechen.

Seit einigen Jahren beschäftigen wir uns darüber hinaus viel mit Tracking und Überwachung durch mobile Apps und Webseiten. Dazu betreiben wir beispielsweise viel Forschung und entwickeln im Projekt Tweasel Infrastruktur, welche die Erkennung von Tracking sowie rechtliche Beschwerde dagegen automatisiert. In der 16. Runde des Prototype Funds werden unsere beiden Vorstandsvorsitzenden gefördert, um ein Tweasel-Addon für Web-Browser zu entwickeln.

Neben diesem technischen Teil machen wir aber auch viel „klassischere“ Vereinsarbeit wie Veranstaltungen, Öffentlichkeitsarbeit und Informationsmaterial oder Beteiligung an Konsultationsverfahren von Aufsichtsbehörden.

Über die Jahre haben wir einiges an Erfahrungen gemacht und gelernt, was an so einer Vereinsstruktur gut funktioniert und was nicht. Diese Erfahrungen wollen wir an dieser Stelle mit anderen FOSS-Projekten teilen, die auch über eine Vereinsgründung nachdenken. Dabei gehen wir nicht darauf ein, wie eine Vereinsgründung konkret abläuft, sondern verweisen hierfür auf den tollen Vortrag „Vereine gründen für das Fediverse“ von Claudius und w4tsn, der sich zwar spezifisch an Vereine für Fediverse-Instanzen richtet, aber zu großen Teilen übertragbar ist. Der

Vortrag bietet auch ein gutes Grundwissen zu der Thematik, das wir im Folgenden voraussetzen.

Unser Weg zum Verein

Die Arbeit an Datenanfragen.de hatten wir, wie sicherlich bei vielen anderen Projekten auch, zunächst ohne irgendeine feste Organisationsstruktur als Privatpersonen gestartet. Als wir erkannt hatten, dass uns das Projekt wichtig ist und wir es längerfristig betreiben wollen, war uns schnell klar, dass wir eine rechtliche Struktur brauchen, welche offizielle Betreiberin des Projekts ist.

Hierfür gab es einige Gründe. Zunächst war uns wichtig, das Projekt möglichst unabhängig von den Initiator_innen zu machen und eine Übertragung der Leitungsrollen auf andere Personen in der Zukunft zu ermöglichen. Wir haben uns explizit auch gewünscht, dass sich die Community am Projekt beteiligen und demokratisch an Entscheidungen mitwirken kann. Darüber war uns aber auch vor allem eine Haftungsbeschränkung wichtig, damit wir und andere ehrenamtlich am Projekt arbeiten können, ohne Angst haben zu müssen, persönlich in die Haftung genommen zu werden. Und schließlich wollten wir dafür sorgen, dass das Projekt dauerhaft gemeinwohlorientiert betrieben wird und niemals Gewinn damit erzielt wird.

In Anbetracht dieser Beweggründe war ein eingetragener Verein tatsächlich die einzige Rechtsform, die für uns infrage kam. Praktischerweise sind beim (gemeinnützigen) Verein auch die Gründungskosten vergleichsweise niedrig. Aber insbesondere über die Gründung eines Unternehmens haben wir gar nicht nachgedacht, weil es uns wichtig war (und immer noch ist), kein Geld mit unseren Nutzer_innen zu verdienen.

Aus gleichen Gründen haben wir uns auch für eine Gemeinnützigkeit entschieden. Wir wollten uns bewusst den deutlich höheren Anforderungen und der strikteren Prüfung durch das Finanzamt unterwerfen, damit Menschen uns vertrauen können, dass wir wirklich ausschließlich gemeinwohlorientiert agieren.Anfangs hatten wir etwas naiv angenommen, dass ein Projekt zum Datenschutz, der ja schließlich ein Grundrecht ist, selbstverständlich gemeinnützig sein kann. Zum Glück gibt es die Möglichkeit, den Satzungsentwurf vom Finanzamt kostenlos prüfen zu lassen, wodurch wir gelernt haben, dass aus dem Vereinszweck sehr klar hervorgehen muss, welche der gemeinnützigen Zwecken aus der leider sehr starren und recht willkürlichen Liste aus § 52 Abs. 2 AO denn konkret gefördert werden und wie.

Welche Rechtsform im Endeffekt die richtige ist (oder ob überhaupt eine nötig ist), ist aber natürlich sehr abhängig vom jeweiligen Projekt und sollte individuell bedacht werden. Und gerade die Entscheidung für oder gegen eine Gemeinnützigkeit sollte gut überlegt sein. Soll die eigene Arbeit an der Software beispielsweise durch Consulting dazu finanziert werden, ist sicherlich eine wirtschaftliche Rechtsform sinnvoll. Auch eine gUG oder gGmbH kann attraktiv sein, wenn das Projekt zwar gemeinwohlorientiert geleitet, Entscheidungen aber nicht unbedingt demokratisch von den Mitgliedern gefasst werden sollen.

Vorteile

- Finanzielles (bei Gemeinnützigkeit)
- Spenden sammeln mit Steueranreiz

- Keine Steuern für Einnahmen
- Gemeinnützigkeit ist oft Voraussetzung für Fördermittel
- Haftungsbeschränkung
- Haftung liegt (meistens) beim Verein, Schutz für Ehrenamtliche
- Kollektive Organisation
- Möglichkeit der Formalisierung von Regeln
- Keine Einzelperson hat Macht über die Projektassets (je nach Satzung)
- Projekt ist von Einzelpersonen unabhängig(er) und lässt sich besser weitergeben
- Kontrollmöglichkeit der Leadership
- Ansehen
- Institution verleiht Seriösität
- Vertrauen in gemeinnützige Organisationen

Nachteile

- Verwaltungsaufwand und Kosten
- Eintragung kostet Geld.
- Relative starre Struktur, bspw. erfordern Satzungsänderungen ziemlichen Aufwand und Kosten. (Kann aber gleichzeitig auch Vorteil sein.)
- (insbesondere bei Gemeinnützigkeit) Admin-Overhead, z. B. für Mitgliederversammlungen und Jahresberichte
- Man braucht Vereinsverwaltungs- und Buchhaltungssoftware, aber wenig gute Optionen (mit FOSS und Datenschutz).
- Privatsphäreeinbußen
- Name, Adresse und Geburtsdatum von Vorstandsmitgliedern werden im Vereinsregister veröffentlicht
- Beschränkungen der Handlungsfähigkeit
- Enge und „intransparente“ Beschränkungen durch das Gemeinnützigkeitsrecht, insbesondere bei Aktivismus häufig unklar, was erlaubt ist, Risiko vorauseilenden Gehorsams

Zu bedenken

- Schwierige Frage: Verhältnis zwischen Leadership im Verein und Maintenance-Aufgaben „im Code“
- Einmal für Gemeinnützigkeit entschieden, kommt man nur noch sehr schwer und ggf. teuer wieder heraus.

Tipps

- Es gibt online etliche Vorlagen für Satzungen, Protokolle, etc.
- Guckt Euch möglichst viele an und kombiniert die Aspekte, die Euch gefallen. Gerade bei der Satzung lohnt es sich, sich anfangs mehr Gedanken zu machen als weniger.
- Bedient Euch gerne auch bei uns, unsere Satzung und alle Ordnungen sind - wie auch dieser Beitrag - CC0: https://www.datenanfragen.de/verein/transparenz
- Notar_innen beglaubigen auch nur Unterschriften, ihr braucht nicht deren Rundumpaket.

Ressourcen

- Vortrag, der die Schritte zur Vereinsgründung erklärt: https://media.ccc.de/v/2024-375-vereine-grnden-fr-das-fediverse
- Aktuelle rechtliche Themen
  - Vereinfacher: https://www.youtube.com/@Vereinfacher
  - socialnet-Newsletter: https://www.socialnet.de/newsletter/
- DSEE
  - Beratung: https://www.deutsche-stiftung-engagement-und-ehrenamt.de/kontakt/
  - Webinare: https://www.youtube.com/@DS-Engagement-und-Ehrenamt
  - Förderdatenbank: https://foerderdatenbank.d-s-e-e.de/
- Zum Verhältnis von Gemeinnützigkeit und politischer Betätigung
  - Allianz Rechtssicherheit für politische Willensbildung: https://www.zivilgesellschaft-ist-gemeinnuetzig.de/
  - Leitfaden von der GFF: https://freiheitsrechte.org/gemeinnuetzigkeit-infomaterial
- Initiative Transparente Zivilgesellschaft: https://www.transparency.de/mitmachen/initiative-transparente-zivilgesellschaft/

□□ Communities für Prototype – Evaluation der 13. und 14. Förderrunde

Um zu lernen, was Projekte ausmacht, die beim Prototype Fund angenommen und erfolgreich umgesetzt werden, und welche Art der Förderung sie am besten unterstützt, sprechen wir regelmäßig mit unseren Geförderten. Bei der Evalaution der Förderrunden 13 und 14 wollten wir wissen, welche Rolle Community Building für die Förderprojekte des Prototype Fund spielt. Das sind unsere Erkenntnisse:

1. Communities sind ein Ideal

Zum Ideal der Open-Source-Softwareentwicklung gehört neben einer Veröffentlichung des Codes unter freier und offener Lizenz traditionell auch, dass die Entwicklung von einer Community getragen wird. Dieses Ideal teilen viele Geförderte aus Runden 13 und 14 und schrieben der Community eine zentrale Bedeutung für die Open-Source-Entwicklung zu. Etwas seltener wurde die Existenz einer Community um ein Open-Source-Projekt zwar als grundsätzlich wichtig, aber z. B. bei kleinen oder kommerziellen Projekten als nicht erforderlich eingeschätzt. Auch für den Prototype Fund hat Community Building einen hohen Stellenwert. Geförderte werden deshalb durch Coaching- und Vernetzungsanbote dabei unterstützt.

2. Neue Communities wachsen langsam

Eine neue Community aufzubauen ist schwierig und braucht Zeit. Geförderte aus Runden 13 und 14 fanden es besonders herausfordernd, Personen mit der richtigen Expertise zu finden. Einige Monate nach Ende der Förderung wurden – gemäß der verfügbaren Informationen in Repositorien – nur sechs der 46 abgeschlossenen Projekte aus Runden 13 und 14 von mehr als fünf Personen entwickelt. Bei drei dieser sechs Projekte bestanden die Repositorien schon mindestens ein Jahr vor Beginn der Förderung durch den Prototype Fund. Da in Repositorien die Arbeit derjenigen, die nichttechnische Aufgaben übernehmen, häufig nicht sichtbar ist, ist diese Schätzung konservativ. Nichtsdestotrotz zeigen die Zahlen, dass der Aufbau einer größeren Community innerhalb von wenigen Monaten nur in wenigen Fällen gelingt.

3. Offenheit für neue Beitragende hat nicht immer Priorität

Obwohl viele Geförderte grundsätzlich eine Community aufbauen wollten und offen für neue Beitragende waren, räumten sie der Einbindung von neuen Personen zu Beginn der Förderung nicht immer Priorität ein. Auf die technische Umsetzung der Projekte aus Runden 13 und 14, wie z. B. die Wahl des Repositoriums, des genutzten Frameworks oder die Strukturierung des Projekts in Module, hatten Überlegungen zum Community Building nur selten einen Einfluss. Auch ihren Code veröffentlichten nicht alle der Geförderten von Beginn an, sondern warteten beispielsweise das Erreichen eines bestimmten Entwicklungsstadiums, erste öffentliche Präsentation ihres Projekts während der Förderzeit oder das Ende der Förderzeit ab. Grund dafür waren teilweise Bedenken, dass der Code einer öffentlichen Beurteilung noch nicht standhalten oder eine solche Öffnung mit einem unerwünschten Kontrollverlust über das Projekt einhergehen könnte.

4. Zusammenarbeit erfolgt meist ad hoc

Keines der Förderprojekte, deren Entwickler*innen für die Evaluation interviewt wurden, hatte formal etablierte Entscheidungsstrukturen. De facto wurden Entscheidungen ungefähr gleich häufig von einer Person allein oder konsensual in der Kernentwickler*innengruppe für das Projekt und, sofern vorhanden, für die Community entschieden. Informationen dazu, wie beigetragen werden kann, z. B. in Form von Contributing Guidelines, hatte zum Zeitpunkt der Evaluation knapp ein Drittel der Projekte veröffentlicht. Die Koordination der Zusammenarbeit, insbesondere mit neuen Beitragenden, empfanden viele Geförderte aus Runden 13 und 14 als Herausforderung.

Wie sollte Förderung für Community Building aussehen?

Bei der Gestaltung von Förderprogrammen ist zu beachten, dass der Aufbau einer Community langwierig ist und Strukturen in der Regel nur dann wirksam sind, wenn sie von der Community gemeinsam und über einen längeren Zeitraum aufgebaut werden. Fördermaßnahmen sollten deshalb in erster Linie darauf abzielen, ein Bewusstsein dafür zu schaffen, wie Projekte von Beginn an offen für neue Beitragende gestaltet werden können. Helfen kann dabei Input gleich zu Beginn der Förderung sowie der Austausch zu Best Practices unter Geförderten. Wichtig ist außerdem, dass Arbeitszeit für Community Building unterstützt wird und Community-Mitglieder, die nicht offiziell Teil der Förderung bzw. des Kernteams sind, in Förderstrukturen eingebunden werden können. In welchem Umfang das während der Förderzeit möglich ist, sollte klar verständlich erklärt sein.

Die ausführlichen Ergebnisse, Informationen zur Methodik der Evaluation findet ihr im Evaluationsbericht. Außerdem erfahrt ihr darin, wer sich beworben hat und gefördert wurde, zu

welchen Themen Projekte eingereicht und ausgewählt wurden, was aus ihnen geworden ist und was das Team des Prototype Fund in Zukunft noch besser machen könnte.