□□ 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.

Revision #10
Created 10 September 2024 08:24:32 by Test Editor
Updated 12 September 2024 10:44:40 by Test Editor