



Mit InnerSource von OSS-Projekten lernen

Als Unternehmen von den Erfahrungen der Open-Source-Szene profitieren: InnerSource zeigt, wie die professionelle Zusammenarbeit auch remote gelingt, und setzt dabei vor allem auf asynchrone Kommunikation und bewährte OSS-Strukturen.

Von Isabel Drost-Fromm

■ Remote-first – oder zumindest hybrid: Die letzten beiden Jahre haben gezeigt, dass nicht nur hippe Start-ups aus dem Bay-Area-Umfeld verteilt arbeiten können, sondern auch etablierte deutsche Unternehmen. Dennoch flammen dieser Tage vielerorts wieder Diskussionen auf, ob und wie man zur Büropräsenz zurückkehren sollte. Dabei sollte Firmen mit einem Fokus auf Softwareentwicklung der Schritt zur Remote-Arbeit relativ leichtfallen: Sie können auf die reichhaltigen Erfahrungen der verteilten Teams im Open-Source-Umfeld zurückgreifen. Dort ist man es schließlich gewohnt, gemeinsam an einem Produkt zu arbeiten, ohne dafür physisch zusammenkommen zu müssen.

Entweder suchen sich die Unternehmen Unterstützung bei Mitarbeiterinnen und Mitarbeitern, die auf persönliche Erfahrungen in Open-Source-Projekten zurückblicken können. Oder sie nutzen die Werkzeuge von InnerSource Commons, einer Stiftung, die dabei hilft, Best Practices aus der Open-Source-Projektarbeit in den Unternehmenskontext zu übertragen (siehe ix.de/z6g5). Dass der InnerSource-Weg auch in der Praxis funktioniert, zeigt die Erfahrung der Autorin im eigenen Unternehmen in den vergangenen beiden Jahren.

Die InnerSource-Bewegung wurde von Open-Source-Experten gegründet, motiviert vom Wunsch, Open-Source-Arbeitsweisen mehr Menschen nahezubringen. Schließlich greifen fast alle Entwicklerinnen im Alltag auf Open-Source-Komponenten zurück, doch nur ein kleiner Teil ist in den Projekten selbst aktiv. InnerSource unterstützt Interessierte dabei, sich in Open-Source-Projekten leichter einzubringen, indem die etablierten Arbeitsweisen quasi inhouse schon erprobt werden. Ganz nebenbei bringt Inner-

Source damit die bewährten Remote-first-Arbeitsweisen ins Unternehmen. Auf diesen Aspekt konzentriert sich dieser Artikel.

Wann InnerSource sinnvoll ist

Klassische agile Softwareentwicklung möchte Abhängigkeiten zwischen Teams minimieren. Nur so können Teams autonom im Sinne ihrer Kunden Entscheidungen treffen. Trotzdem lassen sich in der Praxis nicht alle Abhängigkeiten vermeiden. Sind dann Änderungen über Teamgrenzen hinweg notwendig, bleiben oft nur zwei Möglichkeiten: Entweder wird die Änderung als Feature Request ins Backlog des anderen Teams eingebracht und dann die Umsetzung abgewartet. Oder man löst das Problem über einen Workaround. InnerSource eröffnet einen Mittelweg: Das Team mit dem Änderungswunsch kann anbieten, selbst bei der Umsetzung zu unterstützen – ganz so, wie man es auch von Nutzern von Open-Source-Komponenten erwartet.

Sollen Teams auf diese Weise gemeinsam an Komponenten arbeiten, erfordert das eine entsprechende Kommunika-

tionsweise: Ähnlich wie Open-Source-Projekte bringen InnerSource-Projekte Menschen zusammen, die in unterschiedlichen Teams und Kontexten arbeiten. Selbst wenn alle Beteiligten in einer Zeitzone leben, ist es oft schwierig, Slots für Meetings zu finden, zu denen alle Zeit haben. Daher arbeiten InnerSource-Projekte vorwiegend asynchron.

Die genutzten Kommunikationsmedien müssen es daher erlauben, Nachrichten zu archivieren, zu durchsuchen und zu verlinken – so entsteht quasi nebenbei ein Archiv der „Passive Documentation“. Technisch kann es sich dabei um archivierte Mailinglisten oder um Chat-systeme wie Slack handeln, solange es um unstrukturierte Abstimmungen geht. Sobald konkreter über bestimmte Features

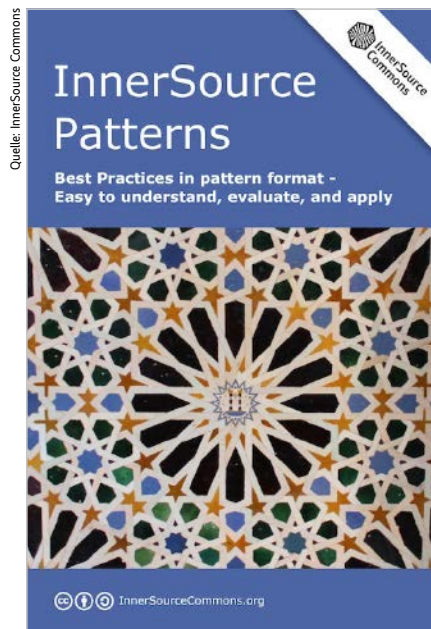
diskutiert wird, bieten sich Ticketsysteme an. Während diese üblicherweise aber nur dazu dienen, einen Überblick über die aktuelle

Arbeit zu geben, werden sie in InnerSource-Projekten sehr viel stärker zur Abstimmung genutzt: Welches Feature soll warum umgesetzt werden? Warum und wie wird die Änderung priorisiert? Das geht bis hin zu Diskussionen rund um das Design der Umsetzung.



IX-TRACT

- ▶ Die Zusammenarbeit in Open-Source-Projekten findet fast ausschließlich physisch getrennt statt. Softwarefirmen, die sich von der Büropräsenz verabschieden möchten, können von diesen Erfahrungen profitieren.
- ▶ InnerSource ist ein Ansatz, Open-Source-Strukturen in den Unternehmenskontext zu überführen.
- ▶ Zentral ist dabei die asynchrone Kommunikation. Dennoch sollten persönliche Zusammentreffen nicht gänzlich unterbleiben.



Quelle: InnerSource Commons

Das online kostenlos zugängliche E-Book „InnerSource Patterns“ vermittelt Best Practices beim Transfer von Open-Source-Strukturen (siehe ix.de/z6g5).

gewinnen. Insofern bilden die Trusted Committer eines InnerSource-Projektes quasi das Host-Team.

Personen, die erst am Anfang ihrer Projektmitarbeit stehen, werden dagegen als Contributor bezeichnet. Üblicherweise setzen sie das Projekt selbst ein und haben Dinge gefunden, die sie gern ändern würden. Genau wie im Open-Source-Kontext führen sie die Änderungen selbst durch – entweder mithilfe existierender Dokumentation oder mit Unterstützung und Mentoring der Trusted Committer.

Das Mentoring findet dabei aber nicht eins zu eins statt, sondern in Form von offener, transparenter Kommunikation auf den Kommunikationskanälen des Projektes – schließlich sind auch Open-Source-Contributors angehalten, öffentliche Kommunikationskanäle zu nutzen, um Fragen zu stellen. Die Vorteile liegen auf der Hand: Im Gegensatz zur direkten Kommunikation können sehr viel mehr Menschen aus ihrer persönlichen Perspektive auf die Frage eingehen und zur

Während der Entwicklung können alle beteiligten Personen so transparent nachverfolgen, was warum wie entschieden wurde, und gegebenenfalls ihre Perspektive einbringen – unabhängig davon, wann sie Zeit haben, sich mit dem Thema auseinanderzusetzen. Nach der Umsetzung bietet die so archivierte Diskussion die Möglichkeit, die Entscheidungsfindung nachzuverfolgen. So vermeidet man auch die oft nach Jahren gestellte Frage: „Warum haben wir das damals noch mal so gemacht und nicht anders – und sind die damaligen Annahmen heute überhaupt noch gültig?“

Da InnerSource-Projekte häufig Menschen aus unterschiedlichen Teams zusammenbringen, ist es wichtig, sich für das Projekt auf gemeinsame Entwicklungsstandards zu einigen. Diese fassen die Beteiligten analog zu Open-Source-Projekten üblicherweise in README- und CONTRIBUTIONS-Dokumenten zusammen. Um auch dabei eine asynchrone Entscheidungsfindung zu ermöglichen und ermüdend lange Meetings zu vermeiden, sollte man diese Dokumente wie Quellcode behandeln und mit den gleichen Prozessen weiterentwickeln.

Rollen aus Open-Source entlehnt

Es bleibt die Frage, wer über die Standards der Zusammenarbeit entscheidet. Bei Open-Source-Software sind das in der Regel die Maintainer der Projekte. Im InnerSource-Projekt bildet die Rolle des Trusted Committer diese Aufgaben ab: Trusted Committer sind nicht nur für das Committed von Code zuständig, sondern auch für die strategische Ausrichtung des Projektes verantwortlich sowie für die Ausgestaltung des Miteinanders, für Codequalität und Security, aber auch dafür, neue Mitstreiter für das Projekt zu

Problemlösung beitragen. Zudem haben so auch andere potenzielle Contributors die Möglichkeit, die Antwort zu lesen.

Betrachtet man die beschriebene Arbeitsweise von InnerSource-Projekten, wird schnell klar, dass der Wechsel hin zu remote-first damit offensichtlich und sehr einfach wird: Schließlich liegt der Fokus der Kommunikation zum Projekt selbst bereits auf asynchronem Austausch und löst damit ein zentrales Problem der verteilten Softwareentwicklung.

Dennoch bleibt – wie bei Open Source auch – der Bedarf nach persönlichem Kontakt: Vor allem, wenn es um Vertrauensbildung und Aufbau des Host-Teams geht, aber auch dann, wenn Contributors stärker an das Team gebunden werden sollen, helfen Treffen, bei denen die Beteiligten physisch zusammenkommen.

Dabei sollte der Fokus aber eher auf dem gemeinsamen Erleben und dem Zwischenmenschlichen liegen – wie auch bei Open-Source-Konferenzen der Hallway-Track oft wichtiger ist als die eigentlichen Konferenz-Vorträge.

Fazit

Mit InnerSource gelingt es sehr einfach, von einer eher bürolastigen Präsenzkultur in Richtung remote-first und zeitweise sogar remote-only zu gehen, wie die beruflichen Erfahrungen der Autorin zeigen. Für einen tieferen Einblick empfehlen sich die Onlinetrainings der InnerSource Commons Foundation. (jvo@ix.de)

Quellen

Mehr zum InnerSource-Einstieg unter ix.de/z6g5

Isabel Drost-Fromm

ist Java-Entwicklerin, Mitglied der Apache Software Foundation und Open Source Strategist bei der Europace AG. 