



Know-how Chancen und Risiken von Open Source

Geld ist nicht alles

von Stefan Schaffner



Quelle: Corina Bocu - 123RF

In wirtschaftlich anspruchsvollen Zeiten ist eine gründliche Vorbereitung einer Investitionsentscheidung unabdingbar. Im Zuge dessen sind auch Überlegungen notwendig, ob die künftige Software als Open Source konzipiert sein soll oder nicht. Denn nicht nur in Sachen Lizenzkosten bringen offen zugängliche Varianten interessante Vorteile mit sich. Aber auch ungeahnte Fallstricke lauern. Dieser Beitrag zeigt Chancen und Risiken für Unternehmen auf, die sich für Open Source-Anwendungen entscheiden.

Grundsätzliche Vorbehalte bezüglich Qualität und Verlässlichkeit einer Open Source Lösung muss ein Unternehmen heutzutage nicht mehr haben. Dennoch sollten künftige Anwender in der Entscheidungsphase sehr genau beleuchten, welche Lösung für den geplanten Einsatzzweck optimal ist und welche Rahmenbedingungen gegeben sind.

Quelloffenheit bietet zahlreiche Vorteile

Setzen Unternehmen auf Open Source-Werkzeuge, können sie in der Regel innerhalb kürzester Zeit profitieren – nicht nur auf rein wirtschaftlicher Ebene wie etwa durch geringe Lizenzkosten. Vielmehr liegt der entscheidende Vorteil der Nutzung offener Anwendungen in der naturgemäßen Offenheit des Quellcodes und der damit unter Umständen verbundenen Dokumentation. Auf diese Weise kann die Expertise einer Vielzahl von Fachentwicklern aus der Community in die Lösung einfließen. Impulse für neue Funktionen sind hierbei ebenso willkommen wie das Finden und Korrigieren von Fehlern innerhalb der Gruppe. Auch im Sinne der Nachhaltigkeit hat die offene Form die Nase vorn. Immerhin sind proprietäre Anwendungen abhängig von der Lebensdauer ihrer anbietenden Firmen. Open Source-Software hingegen wird von den Anwendern und deren Weiterentwicklungsbestreben getra-

gen. Ob der Urheber und sein Unternehmen noch Marktpräsenz zeigen oder nicht, spielt hier eher eine untergeordnete Rolle.

Offene Quellcodes punkten auch in Interoperabilitätsfragen, Usability und Schnittstellenoffenheit. Auch hier liegt der Vorteil in der Natur der Sache: Viele Experten kennen eben auch viele Schnittstellen und Anwendungsbrücken, die sie in der Regel auch in die Lösung integrieren. Davon profitiert natürlich in erster Linie der Anwender. Selbstverständlich sind Anbieter von Closed Source-Varianten auch bestrebt, beispielsweise im Sinne der Interoperabilität, hohe Standards zu setzen. Gleichwohl können sie mit den Möglichkeiten einer Open Source-Software oft nicht Schritt halten.

Wollen Unternehmen eigene Anpassungen an der Anwendung vornehmen, sind sie ebenfalls besser beraten, wenn sie auf offene Quellcodes setzen. Immerhin haben sie so die Möglichkeit, die Dinge ihren Bedürfnissen folgend mitzugestalten – ohne dafür vom Anbieter einer geschlossenen Lösung zur Kasse gebeten zu werden. Ohnehin sind die Nutzer von Open Source-Anwendungen flexibler. Beispielsweise müssen sie keine Obsoleszenz einplanen und auf neuere Versionen updaten, wenn sie es nicht möchten. Darüber hinaus sind offene Quellcodes dafür

bekannt, dass sie inhomogene IT-Infrastrukturen und "exotische" Plattformen deutlich besser unterstützen können als Closed Source-Lösungen. Auch die Mitarbeit mehrerer Unternehmen oder Institutionen wird durch den frei verfügbaren Quellcode deutlich einfacher. Und: Die Anwender haben nur sehr geringe bis keine Lock-in-Effekte zu befürchten, da sie in der Regel keine Anbieterbindung und damit einhergehende Lizenzkosten haben und somit schneller die Anwendung wechseln können.

Vorhandene Ressentiments in den Themen "Sicherheit" und "Zuverlässigkeit" von Open Source haben die Erfahrungen der letzten Jahre ebenfalls entkräftet. Die Vielzahl der – meist auch namentlich bekannten – Entwickler der Community haben stets Einblick in den Quelltext. Probleme können so folglich zeitnah erkannt und durch den Einsatz intelligenter Kollaborationssysteme auch ebenso schnell behoben werden. Getreu dem Motto "Viele Augen sehen viel" partizipieren die Anwender beim Einsatz einer solchen Lösung also gleich mehrfach. Die große Gruppe der Entwickler und deren Identifizierbarkeit sorgen darüber hinaus dafür, dass keine schädliche Software in Umlauf gebracht wird. Wird eine Sicherheitslücke bekannt, wird die Community in der Regel umgehend via Internet informiert und



häufig binnen kurzer Frist auch eine Korrektur vorgenommen. Da sicheres Programmieren und die Integration von Sicherheitsfunktionen im Bereich der Open Source-Software einen hohen Stellenwert haben, sind solche Lösungen in der Regel auch weniger von Viren betroffen.

Wo Licht ist, ist auch Schatten

Auch wenn die Vorteile einer Open Source-Variante beeindruckend mögen – die Medaille hat zwei Seiten. Beispielsweise sind die TCO ein wesentlicher Punkt, der möglicherweise gegen den Einsatz einer offen gestalteten Lösung sprechen könnte. Bei der Kostenbetrachtung fallen Lizenzkosten kaum ins Gewicht. Jedoch sind, insbesondere bei komplexeren Anwendungen, die notwendigen Aufwände für die Konzeption, Inbetriebnahme, Wartung und den Betrieb zu beleuchten. Viele Lösungsanbieter im Open Source-Bereich bauen genau hierauf ihr Businessmodell auf: Freie Verfügbarkeit der Software, jedoch kostenpflichtige Dienstleistungen für Beratung und Inbetriebnahme.

Auch die Offenheit der Lösung hat ihre Schattenseiten. Immerhin ist die Qualität der "Mitarbeit" der Experten unterschiedlich hoch – sowohl in Entwicklungs- als auch in Dokumentationsfragen. Überhaupt ist es mit der Expertise der Fachleute so eine Sache: Schließlich ist sie stark von der Agilität und der Responsiveness der Community abhängig. "Schläft" die Gemeinschaft, "schläft" folglich auch die Lösung – was im Grunde eine andere, neue Art von Abhängigkeit für den Anwender bedeutet. Die haben sie zwar bei einer geschlossenen Software auch, aber in diesem Szenario haben Unternehmen in der Regel feste Service- und Wartungsverträge mit den Anbietern abgeschlossen und sind auf diese Weise etwas abgesichert. Die Pflege oder der Support eigener individueller OSS-Lösungen verursacht im Vergleich dazu oft sogar höhere Kosten als der Einsatz von Standard-Software proprietärer Lösungen.

Besonders kritisch wird es aber in Sicherheitsfragen: Beispielsweise garantieren Open Source-Entwicklungen keine "security through obscurity". Bei proprietärer Software kann kein Unbefugter hinter die

Fassade schauen und eine Sicherheitslücke anhand des Quellcodes erkennen. Diese Art der Anwendungssicherheit baut schlicht darauf, dass die Funktionsweise der Lösung geheim bleibt und Sicherheitslücken somit schwerer zu finden sind. Der Anbieter verheimlicht also Funktionsweisen und hofft so, nicht geknackt zu werden. Bei OSS geht das nicht. Schaut ein erfahrener Hacker in die Quellen, sieht er die typischen Lücken sofort. Das wissen auf der anderen Seite aber natürlich auch die Programmierer und bauen diese Lücken erst gar nicht in das System ein.

Der Segen der Community, dass Funktionen schnell durch die Heerschar von Entwicklern umgesetzt werden, kann auch zum Fluch werden. Wenn sich niemand in der Gruppe für die eigene Idee begeistern lässt, stehen Unternehmen in der Regel vor einem Problem. Bei einer proprietären Lösung wenden sich IT-Verantwortliche bei individuellen Wünschen einfach an den Hersteller und bitten (gegen Geld) um die Erweiterung der Lösung. Das geht bei OSS nicht so einfach. Ist die Community träge, uninteressiert oder schlicht nicht mehr vorhanden, sitzen die Anwender schnell auf dem Trockenen. Dann müssen sie in der Regel selbst ran und hoffen, dass die eigenen Entwickler mit dem Code und der Dokumentation umgehen können.

Diese Problematik trifft im Übrigen auch auf andere Situationen zu wie beispielsweise beim Wechsel auf ein neues Betriebssystem. Erwärmt sich niemand in der Community für die Portierung auf das neue OS, steht die Lösung vor dem Aus oder der Nutzer muss auf einer alten OS-Version bleiben. Überhaupt gilt es in Flexibilitätsfragen das Für und Wider genau abzuwägen. Beispielsweise stehen Unternehmen mit Open Source-Anwendungen schnell im Regen, wenn sie als Betriebssystem Linux einsetzen wollen, da viele Spezialanwendungen nicht für Linux erhältlich sind und folglich ersetzt oder neu entwickelt werden müssen.

Befragt man Anwender nach Ihren Erfahrungen, so zeigt sich, dass die größten Hindernisse Probleme bei der Integration mit den vorhandenen Softwarelösungen und

der Mangel an qualifiziertem Personal sind. Fehlende oder fehlerhafte Funktionen werden zwar auch genannt, sind aber ebenso wie mögliche Probleme in der Kontaktaufnahme mit dem Support eher nachrangig.

Open Source Best Practice

Es gibt also gute Argumente für und gute Argumente gegen den Einsatz offener Lösungen. Alles in allem bleibt es wie so oft eine Einzelfallentscheidung, wann eine Open Source-Variante sinnvoll ist oder nicht. Erfahrungsgemäß gibt es bestimmte Szenarien, wo der Einsatz offener Anwendungen überaus sinnvoll ist und die Nutzer spürbar von den beschriebenen Vorteilen profitieren.

Im Bereich der Serverinfrastruktur und der Datenbanken etwa haben sich Open Source-Produkte bereits zu einem Standard mit breiter Akzeptanz entwickelt. Hierzu zählt der Linux-Server in vielen Bereichen der Unternehmens-IT genauso wie die MySQL-Datenbank im Bereich der Webanwendungen. Proprietäre Anwendungen sind hier inzwischen in der Minderheit. Für diese Systeme haben Unternehmen oftmals bereits interne Kompetenzen aufgebaut. Alternativ stellt der Markt eine Fülle von Experten für die entsprechenden Anwendungen bereit.

Eine Orientierungshilfe für Unternehmen gibt bereits die Klassifizierung des Einsatzbereiches: Häufig werden OSS-Lösungen im Bereich der Serverinfrastruktur, des Internets, der Datenbanken, der Netzwerkinfrastruktur, der Sicherheit und des Managements eingesetzt. Weniger häufig finden sich diese Lösungen bei Collaboration, Dokumentenmanagement sowie bei ERP- und CRM-Systemen.

Für die Evaluierung einer Open Source-Lösung ist die nachhaltige Kostensenkung oftmals ein wesentlicher Treiber. Dabei sollten Unternehmen zunächst den Blick auf die Lizenzkosten richten. Neben den einmaligen Anschaffungskosten können sich Anwender bei Open Source auch vor den seitens der Anbieter getriebenen kostenpflichtigen Release-Wechseln und sich somit vor tiefen Abhängigkeiten gegenüber dem Softwareanbieter wie unnötigen Updates, willkürlichen Preiser-



höhungen und fehlenden Integrationsmöglichkeiten schützen.

Von diesen – wenn auch starken – monetären Argumenten allein sollten sich Unternehmen jedoch nicht leiten lassen. Entscheidend ist, dass sich die Lösung in die Geschäftsprozesse einfügt und die Anforderungen für die eigene Branche erfüllt sind. Oftmals sind hierfür branchenspezifische Module erforderlich, die im Bereich des OSS nicht flächendeckend und vor allem vollständig verfügbar sind. Darüber hinaus ist zu beachten, dass die Integration der Open Source-Lösung in die Unternehmensstrukturen möglich sein muss und zu den eigenen IT Standards passt.

Open Source-Checkliste

Folgende Fragestellungen können helfen, eine Open Source-Lösung für den Einsatz im eigenen Unternehmen zu validieren:

- **Relevanz des Produkts in meiner Branche:** Entwicklungsteams von ausgereiften Lösungen stellen oftmals Listen von Anwendern zur Verfügung, die das Produkt einsetzen. Anhand dieser Information lässt sich prüfen, ob die Lösung hält, was sie verspricht, und ob sie in der eigenen Branche anwendbar ist.
- **Funktionsgarantie/Support für den Unternehmenseinsatz:** Eine Open Source-Lizenz beinhaltet weder eine Funktionsgarantie noch garantierten Support. Funktionierende Communities sind hier jedoch oftmals sehr gut organisiert. Auch wenn der Support auf freiwilliger Basis bereitgestellt wird, finden IT-Verantwortliche für jegliche Fragestellung einen Spezialisten in der Entwicklergemeinschaft. Sofern das für den Unternehmenseinsatz keine ausreichende Sicherheit bietet, sollte die Verfügbarkeit von Drittanbietern für diese Leistungen geprüft werden.
- **Funktionierende Community:** Eine Lösung besteht nicht nur aus dem Quellcode, sondern steht oder fällt mit einer funktionierenden Gemeinschaft von Entwicklern, die sich ständig mit dem Produkt auseinandersetzen. Mehr Augen entdecken mehr Fehler und entwickeln breit aufgestellte Funktionen und Schnittstellen. Eine aktive Community ist ein wesentlicher Indikator für die Qualität und Beständigkeit einer Lösung.

- **Ausreichende Prozesse zur Sicherung der Qualität:** Betrachten wir die großen und bekannten Open Source-Projekte, stehen diese in punkto Qualitätssicherung den konventionellen Softwareherstellern in nichts nach. Dennoch sollte die IT-Abteilung prüfen, ob die vorhandenen Prozesse der Qualitätssicherung im eigenen Haus diese Lücke schließen können oder ob auch in diesem Bereich externe Dienstleister hinzugezogen werden sollten. Insbesondere bei unternehmensspezifischen Modifikationen der Lösung müssen externe und interne Qualitätssicherungsmaßnahmen ineinandergreifen.
- **Einfluss auf die Weiterentwicklung:** Open Source-Projekte werden seitens der Entwicklerteams unterschiedlich organisiert. Daraus resultiert auch der Grad der Einbeziehung der Anwender. Idealerweise werden in strategische Entscheidungen die Entwickler, Nutzer und die Drittanbieter mit einbezogen. Dabei gilt der Grundsatz: Bei steigender Nutzeranzahl und deren Einflussnahme wächst die Chance, dass die Lösung auf das eigene Unternehmen passt.
- **Bereitgestellte Dokumentation:** Da sich viele Entwicklerteams primär auf den Quellcode, die Fehlerbeseitigung und das Hinzufügen von Funktionen konzentrieren, bleiben die Pflege des Benutzerhandbuchs und teilweise auch die Quellcode-Dokumentation als reine Pflichtaufgabe oft hinter der Lösung zurück. Achten Sie daher stets darauf, dass die unternehmensinternen Richtlinien erfüllt bleiben. Sorgen Sie notfalls durch Erstellung von Administrations- und Anwenderdokumentation dafür, dass Ihre Richtlinien erfüllt werden.
- **Regelmäßige Aktualisierungen:** Hierbei geht es weniger um das Verbarmachen von neuen Funktionen, sondern um die Verlässlichkeit und die Geschwindigkeit, in der Sicherheitskorrekturen (Security Patches) bereitgestellt werden. Dies stellt ein wesentliches Qualitätsmerkmal einer Lösung dar. Praktischerweise kann jeder Anwender – ausreichende Entwicklerkenntnis vorausgesetzt – selbst Korrekturen am Quellcode vornehmen und diese dann der Entwicklergemeinschaft zeitnah zur Verfügung stellen.

- **Anpassbarkeit an die Unternehmensanforderungen:** Ein wesentlicher Vorteil von Open Source liegt in ihrer Anpassbarkeit an die eigenen Anforderungen, da der Quellcode frei verfügbar ist. Im Rahmen der Evaluierung gilt es folglich zu prüfen, wie gut die grundlegende Architektur und die Schnittstellen dokumentiert sind und in welcher Qualität der Code geschrieben ist. Auch wenn Anpassungen einem Drittanbieter überlassen werden sollen, wird die Qualität seiner Arbeit wesentlich von diesen Faktoren abhängig bleiben.
- **Passt die Lösung zur Unternehmensgröße:** Neben der Funktionalität einer Lösung ist die Frage der Skalierbarkeit essenziell. Folglich sollten IT-Verantwortliche bei der Auswahl der Lösung auf die zugrundeliegenden Technologien achten. Auf den ersten Blick ist oftmals nicht zu erkennen, ob die Lösung auch bei der eigenen Unternehmensgröße noch stabil und performant einsetzbar ist.

Fazit

Open Source-Software ist zweifelsohne zu einem wichtigen Bestandteil in Unternehmen geworden. Das ist mit Blick auf die zahlreichen Vorteile wenig verwunderlich. Immerhin sparen Anwender nachweislich Kosten und profitieren von einer engagierten und kompetenten Community, die die Lösung nach vorn bringt. Auf der anderen Seite ist der Einsatz von OSS auch mit Risiken verbunden, wie etwa die Angreifbarkeit der Lösung oder Flexibilitätseinbußen. Insofern ist und bleibt es wie in vielen anderen IT-Themen auch eine Abwägungsfrage, ob und inwieweit ein Unternehmen auf Open Source setzt. Viele tun es bereits, insbesondere in den Bereichen Serverinfrastruktur und Datenbanken. Dabei gilt es aber unterschiedliche Rahmenbedingungen zu beachten. Ob und inwieweit eine offene Lösung zum Einsatz kommen kann und sollte, ist beispielsweise auch eine Frage der Unternehmensstrukturen und der bestehenden IT-Landschaft. Mitunter können externe Berater hier mit Hilfe ihrer Erfahrung eine wesentliche Stütze sein, die richtigen Entscheidungen zu treffen. (j/p)

Stefan Schaffner ist Geschäftsführer der ASS it-systemhaus GmbH.