



Raphaël Pinson (aka raphink) ist Infrastruktur-Entwickler und Trainer bei Camptocamp. Seine Fachgebiete sind Automatisierung und die Implementierung von DevOps-Praktiken.

Open Source, Standards und die technische Schuld

Viele Unternehmen nutzen zunehmend Open Source Software, wofür es unzählige gute Gründe gibt. Allerdings ist es immer noch nicht üblich, dass Unternehmen auch einen Beitrag an die Weiterentwicklung leisten. Nach wie vor sehen die meisten Firmen Open Source als reinen Verbrauchernutzen an.

Warum sollten Sie zu Open Source Software beitragen?

Früher war ich immer der Meinung, dass die besten Argumente dafür, zu bestehenden Projekten beizutragen, die Wartung und Kompatibilität sind. Wenn ich ein Projekt privat forke und ihm Funktionalität hinzufüge, besteht das Risiko, dass meine Änderungen im Laufe der Zeit immer schwerer zu warten sind. Wenn jedoch weitere Entwickler*innen über meine Änderungen informiert sind, minimiert sich dieses Risiko stark.

Wenn ich also meine Änderungen dem Communityprojekt beisteuere, wird sichergestellt, dass diese auch über die Zeit mit dem Basiscode kompatibel bleiben. Es könnte sogar Verbesserungen an meinem Code geben, wenn mehr Leute in der Zukunft auf meinen Änderungen aufbauen.

Heute glaube ich jedoch, dass dieses Beispiel ein spezieller Fall einer allgemeineren Regel ist, die mehr pragmatische Gründe umfasst, Code als Open Source beizutragen. Dies hängt mit dem Konzept der technischen Schuld zusammen: die Idee, dass technische Entscheidungen versteckte Kosten (eine "Schuld") mit sich bringen, die in der Zukunft bezahlt werden müssen, um den aktuellen Stand der Technik einzuholen.

Wie kann ich die Schuld minimieren?

Die Minimierung von technischen Schulden ist ein umfangreiches Thema. Ich bin allerdings überzeugt, dass dieses Risiko signifikant reduziert werden kann, wenn man sich an Standards hält. Je strikter sich ein Projekt an Industriestandards hält, desto unwahrscheinlicher ist es, dass es in der Zukunft auf einen anderen technologischen Stack portiert werden muss.

Was ist, wenn die Standards nicht meinen Anforderungen entsprechen?

Wird man mit einer fehlenden Funktion konfrontiert, reagiert ein Grossteil der Menschen mit dem Bau einer spezifischen Komponente gemäss Ihrem Anwendungsfall. Dies ist allerdings ein eher riskantes Vorgehen, wie es auch der Strategie-Theoretiker Simon Wardley beschreibt. So wird die Komponente in die Genesis-Phase verschoben und damit unberechenbarer, weniger standardisiert und anfälliger für das Entstehen technischer Schulden.

Es gibt allerdings eine andere Möglichkeit. Wenn ich ein valides, jedoch unerfülltes Bedürfnis habe, dann könnten andere Menschen dieses Bedürfnis in der Zukunft ebenfalls haben. Wenn dem so ist, dann wird irgendwann, irgendwo, einen neuen Standard für dieses Bedürfnis schaffen. Setzt sich dieser neue Standard durch, dann wird die Schuld meiner spezifischen Komponente offensichtlich werden.

Die Erstellung und Nutzung von Standards in der Entwicklung von Open Source Technologien hilft, die Entstehung einer technischen Schuld zu vermeiden.

Was wäre also, wenn ich - anstatt eine spezielle Komponente zu bauen, um den fehlenden Standard auszugleichen - den neuen Standard selbst festlegen würde? Mit Open Source können Sie genau das tun! Es gibt Ihnen die Möglichkeit, der erste zu sein, der eine offene Implementierung für einen generischen Bedarf anbietet und somit einen neuen Standard schaffen kann. Wenn dieser neue Standard sich durchsetzt, haben Sie nicht nur Ihr Problem gelöst, sondern auch keine technischen Schulden angehäuft und sind den anderen Benutzern immer einen Schritt voraus.

Moment, wir sind kein FAANG!

Es ist offensichtlich, dass es sich die Mehrheit der Unternehmen nicht leisten kann, sich mit IETF-RFCs zu beschäftigen oder ISO-Standards an ihre Bedürfnisse anpassen.

Ein Standard muss jedoch nicht so kompliziert sein. Nehmen wir an, ich verwende dieses beliebte CLI Tool und ich muss eine Option spezifizieren, die noch nicht existiert. Ich könnte etwas hacken, um die benötigten Optionen zu erzeugen. Oder ich könnte das Tool patchen und ein neues Flag für meine Bedürfnisse hinzufügen, und diese Änderung zurück in das Projekt einbringen. Die Chancen stehen wir bereits erwähnt gut, dass jemand anderes diese Option auch braucht.

Somit wird jedes Mal, wenn diese Option benötigt wird, mein neues Flag verwendet und man hat zu einem neuen Standard beigetragen und gleichzeitig keine technischen Schulden erzeugt.

Es kommt nicht auf die Grösse der Schritte an, sondern auf die Richtung, in die Sie sie gehen.

