

Wie alle anderen Branchen auch sind Verlage heutzutage auf Software angewiesen. Zu den strategischeren Werkzeugen gehören dabei Satzprogramme, E-Book-Lesesysteme, Web-Browser, Schriften, Office-Programme, Prüf- und Konvertiertools, Content-Management-Systeme und Verwaltungssoftware für Produktplanung und Metadaten.

Software, gerade wenn sie strategisch ist, bedeutet Risiko, und das nicht nur wegen zufälliger Bugs. Ein wesentliches Risiko ergibt sich aus der Abhängigkeit vom Softwareanbieter. Ist er groß und börsennotiert, macht er was er will, zum Beispiel die Lizenz auf ein viel teureres Abomodell umstellen. Ist er hingegen klein, kann es geschehen, dass er plötzlich den Betrieb, Vertrieb oder Support einstellt. Als Beispiel mag ein Formeleditor für ein bekanntes Satzprogramm dienen, dessen Anbieter Anfang 2011 plötzlich den Vertrieb einstellte.

Ein Teil der Risikovorsorge besteht darin, einen *Lock-In* zu vermeiden, also dass man bestimmte Tools nicht durch äquivalente Tools anderer Anbieter substituieren kann – wegen eines proprietären Datenformats, wegen einzigartiger Funktionalität oder weil umworbene Dritte (Autoren, Leser) im jeweiligen Software-Ökosystem zu Hause sind.

So gibt es im Bereich der Manuskriptbearbeitung zwar einen Platzhirschen – Word –, doch muss man sich kaum Sorgen machen, sollte Microsoft seine Lizenzpolitik radikal verschlimmern. Das Format ist offen, und mit LibreOffice gibt es vollwertigen Ersatz. Einen Lock-In gibt es höchstens bei angepassten Nutzeroberflächen, insb. bei interaktiven Anwendungen mit VBA-Skripting.

Beim Satzprogramm InDesign stehen die Chancen momentan schlechter, eine Alternative zu finden. Einige Verlage möchten das, aber Quark z.B. ist keine Alternative, denn dies ist beim Datenformat viel weniger offen, und preislich kaum eine Verbesserung. Das von uns verwendete TeX ist kein 1:1-Ersatz – man kann zwar relativ standardisierte Layouts damit automatisiert befüllen, aber interaktiv gestalten kann man damit kaum. Da der Zug andererseits immer mehr in Richtung Standardisierung geht und es z.B. mit Speedata ein TeX-basiertes Open-Source-Tool gibt, um komplexere Designentscheidungen automatisiert treffen zu können, gibt es möglicherweise doch einen Ansatz. Es handelt sich dann aber um andere (standardisiertere) Inputdaten und Layouts, man kann

also nicht einfach InDesign mit Speedata ersetzen. Ersatz durch Word oder LibreOffice ist auch möglich, aber nur mit signifikanten Abstrichen (kein CMYK, keine Farbprofile, keine Farbseparationen, ...).

Ein Kandidat für eben diese 1:1-Ersetzung von InDesign möchte das Open-Source-Tool Scribus¹ sein. Wir haben damit leider kaum praktische Erfahrung. Auf den ersten Blick fehlen fundamentale Features wie verankerte Textrahmen, Registerhaltigkeit und automatisch erstellte Verzeichnisse, aber das Dateiformat (Scribus-XML) und die Scriptbarkeit (Python) sind schon mal sehr zugänglich, und außerdem lassen sich Formeln mittels LaTeX integrieren. In einem verwandten Software-Bereich, der 3D-Modellierung, gräbt inzwischen ein ehemals belächeltes Open-Source-Tool namens Blender den proprietären Platzhirschen das Wasser ab.

Kommerziell und Open Source – kein Widerspruch

Doch wie kann Open Source eine größere Sicherheit bieten, wenn Freiwillige ohne kommerzielles Interesse entwickeln? Was, wenn sie das Interesse verlieren? Zunächst einmal ist die Annahme hinter dieser Frage in Frage zu stellen: Es gibt Open Source mit kommerziellem Interesse. Es gibt eine Firma mit Milliardenumsatz, RedHat, mit einem Open-Source-Kernprodukt. Anstatt für Lizenzen zahlen die Kunden eben für Weiterentwicklung, kundenspezifische Anpassungen und Bereitstellung.

Der entscheidende Punkt ist: da der Quellcode offen zugänglich ist und Open-Source-Lizenzen genau dies gestatten, kann man die Software notfalls forken, d.h., einen Klon weiterentwickeln (lassen).

Kommerzielle Weiterentwicklung kann finanziert werden, indem Firmen wie Red Hat Deckungsbeiträge aus anderen Dienstleistungen und Produktverkäufen verwenden, um Entwickler zu bezahlen. Es gibt aber auch das Modell, bei dem Kunden gezielt für die Entwicklung von Features zahlen, die danach Teil des Open-Source-Angebots werden – je nach Lizenz teilweise sogar werden müssen.

Dieses Konzept lief bei le-tex dem Gerechtigkeitsgefühl einiger Kunden und Kollegen entgegen: Wir sollen zahlen, und alle anderen, incl. unsere Wettbewerber, dürfen es dann nutzen? Dem kann man

1 <http://www.scribus.net/>

entgegen, dass man im Gegenzug von der ganzen existierenden Funktionalität profitiert, für die bereits andere gezahlt haben.

Anbieter wie BaseX (XML-Datenbank für Content Management) setzen auf beide Komponenten: Querfinanzierung der Entwicklung über Beratungsumsätze und gezieltes Sponsoring von Features. Dabei ist das komplette Produkt stets quelloffen unter der liberalen BSD-Lizenz.

Ein weiteres Erlösmodell sind duale Lizenzen, wie sie z.B. von Alfresco (Content Management) angeboten werden. Nutzer, für die das System missionskritisch ist, die garantierten Support benötigen oder die bestimmte Enterprise-Features benötigen, müssen eine kommerzielle Lizenz erwerben.

Wir haben für unser Prüf- und Konvertierframework transpect unter einer BSD-Lizenz veröffentlicht, es gibt also keine duale Lizenz. Wir haben aber eine weitere ErlösKomponente erschlossen, die unsere Kunden mit ihrem Interesse an Produktionssicherheit sehr gut verstehen:

transpect-Installationen bestehen aus zahlreichen Modulen, die kundenspezifisch zusammengestellt werden. Da die Module ständig weiterentwickelt werden, möchten die Kunden die neuesten Bugfixes und Funktionalitäten nutzen. Dabei kann es aber zu Inkompatibilitäten kommen und infolgedessen zu unerwünschten Konvertierungsergebnissen oder Abbrüchen. Wir bieten deswegen an, dass wir anhand einer Auswahl an Inputdateien bei jeder Aktualisierung von Basismodulen testen, ob die kundenspezifische Konverterkonfiguration weiterhin dieselben Ergebnisse zeitigt. Ist dies nicht der Fall, setzen sich die Verantwortlichen der Kundeninstallation und der Basismodule ins benehmen und schauen, ob es sich um eine gewünschte Änderung handelt oder um unerwünschte Nebeneffekte. Letztere versuchen sie zu vermeiden; ist dies nicht möglich, schlägt die Verantwortliche für die Kundeninstallation dem Kunden eine Anpassung der Installation im Rahmen des Supports vor. Einsteilen produziert der Kunde mit kompatiblen Modulen weiter. Dieses Modell ist mit einer Wartungspauschale bei kostenpflichtiger Enterprise-Software zu vergleichen. Der Deckungsbeitrag aus dieser Überwachungspauschale finanziert die Basisentwicklung mit.

Kooperation

Ein Phänomen, das wir bei unserem Prüf- und Konvertiertool transpect sehen: es tun sich Verlage (in dem Fall: C.H. Beck, Beltz, Campus und Unionsverlag) zusammen, um einen Standardworkflow (in dem Fall: docx und IDML nach XML bzw. docx nach

IDML konvertieren und aus dem XML EPUB erzeugen) einrichten zu lassen. Die Kosten für den einzelnen Verlag sind dann überschaubar, und das o.g. Ungerechtigkeitsgefühl ist weniger stark ausgeprägt. Nutzen andere Verlage diesen Workflow, sinken für die Initiatoren mittelfristig die Wartungs- und Weiterentwicklungskosten, denn einige dieser „Trittbrettfahrer“ werden später ebenfalls Features benötigen und finanzieren, die von allgemeinem Interesse sind.

Was aber tun, wenn man ein Open-Source-Tool wie Scribus interessant findet, aber Features fehlen und kein kommerzieller Support angeboten wird? Am besten nimmt man mit den Entwicklern Kontakt auf und fragt danach. Kürzlich haben wir das Crowdfunding einer Privatperson für native MathML-Unterstützung in Browsern und EPUB-Readern mit 500 € unterstützt.² Wir haben aber keine in der Bilanz aktivierbare Lizenz oder Individualsoftware erhalten. Der Initiator konnte uns jedoch eine Rechnung für Beratung ausstellen, so dass wir es immerhin als Betriebsausgaben absetzen konnten.

Prinzipiell günstiger ist die Situation, wenn eine gemeinnützige Organisation die Entwicklung koordiniert und Zahlungen an die Entwickler weiterleitet. Bei Organisationen wie der Apache Software Foundation ist dabei jedoch problematisch, dass sie nur in den USA als gemeinnützig anerkannt sind und Zuwendungen nicht abgesetzt werden können. Als Partner in Deutschland könnte die Document Foundation in Frage kommen, welche die LibreOffice-Entwicklung verantwortet.³

Zusammenfassend: Gerade bei „Infrastruktursoftware“ hat sich die Erkenntnis durchgesetzt, dass alle ungefähr dieselben Herausforderungen zu meistern haben und sich kein Verlag hier mit technologischen Alleinstellungsmerkmalen differenzieren muss. Deswegen ist Kooperation von Verlagen, Entwicklern wie le-tex und anderen Dienstleistern sinnvoll.

Kontakt

Gerrit Imsieke (gerrit.imsieke@le-tex.de)
<https://twitter.com/letexml>

Kolophon

Dieses Handout wurde mit LibreOffice 4.3.0.4 erstellt. Die Grundschrift ist Crimson Text, und die Überschriften sind aus der Ek Mukta gesetzt. Beide sind als Open-Source-Schriften von Google Fonts erhältlich.⁴

² <http://www.ulule.com/mathematics-ebooks/>

³ <http://www.documentfoundation.org/>

⁴ <https://www.google.com/fonts>