

Klein aber oho!

Einfache kooperative Softwareentwicklung in Bibliotheken – am Beispiel von beluga core

Matthias Finck

1 Einleitung

Softwareentwicklung spielt gerade im Zusammenhang mit Open Source Software in Bibliotheken eine immer größere Rolle. Dabei werden viele Entwicklungsvorhaben nicht von einer Bibliothek alleine umgesetzt, sondern in Kooperation mit anderen Häusern und Partnern. Dies geschieht entweder aus Überzeugung für den Weg in Richtung Open Source Lösungen oder auch weil die Ressourcen schlichtweg keinen alternativen Weg zulassen.

Das Ergebnis sind Entwicklungsprozesse, die einerseits eine enge, intensive Zusammenarbeit auf Augenhöhe erfordern, andererseits durch zeitliche, räumliche und vor allem organisatorische Verteiltheit geprägt sind – Prozesse, die konsensorientiert gestaltet werden müssen und wo doch jeder Partner im Rahmen teilweise sehr unterschiedlich verfügbarer Ressourcen seine Anforderungen an die Software umgesetzt sehen möchte.

Es gibt einige gut dokumentierte Beispiele großer internationaler und nationaler Projekte wie Koha, Quali-OLE oder Kitodo, die zeigen, dass kooperative Entwicklungsprozesse im Bereich bibliothekarischer Open Source Software erfolgreich umgesetzt werden können. Diese Projekte verfügen alle über ein professionelles Projektmanagement und teilweise erhebliche finanzielle Ressourcen zum Erhalt der organisatorischen und technischen Infrastruktur.

In diesem Beitrag soll anhand des Beispiels des Projekts beluga core gezeigt werden, dass kooperative Entwicklungsprozesse auch im Kleinen erfolgreich funktionieren können. Um gemeinsam erfolgreich Software zu entwickeln, braucht es nicht zwangsläufig globale Entwicklungsteams und fest etablierte Managementstrukturen. Mit Hilfe dieses Beispiels soll gezeigt werden, mit welchen organisatorischen und technischen Mitteln den Herausforderungen kooperativer Entwicklungsprozesse auch in kleinen Projekten begegnet werden kann. Dabei werden die konkreten Werkzeuge und Methoden vorgestellt, die in beluga

Kooperative Softwareentwicklungsprozesse sind ein zunehmendes Phänomen im Bibliothekswesen. Dabei entscheiden sich Bibliotheken aus unterschiedlichsten Motiven für einen gemeinsamen Entwicklungsprozess – von tiefer Überzeugung für Open Source Entwicklungen bis hin zu schlichtweg zu knappen Ressourcen für eine vollständige Eigenentwicklung. In diesem Beitrag soll mit dem Projekt beluga core ein kooperativer Softwareentwicklungsprozess der Staats- und Universitätsbibliothek Hamburg sowie der Universitätsbibliotheken Braunschweig, Lüneburg und Hildesheim vorgestellt werden, der mit Hilfe vergleichsweise geringer Ressourcen einfach und leichtgewichtig etabliert wurde. Das Beispiel soll motivieren, bereits bei kleinen Entwicklungsvorhaben über Entwicklungskooperationen nachzudenken, indem aufgezeigt wird, welche Vorteile für alle Partner daraus erwachsen.

Cooperative software development processes becoming more and more popular in librarianship. The reasons range from conviction for open source developments to low budgets in development divisions. This paper presents the case study beluga core – a collaborative software development process of the Hamburg State and University Library and the University Libraries of Braunschweig, Lüneburg and Hildesheim, which was established as a simple and lightweight process by using relatively low resources. The example should motivate to think about collaborative scenarios even for small development projects by showing the benefits for all partners.

core zum Einsatz kamen, und der Erfolg des kooperativen Prozesses reflektiert.

Die Frage, die dieser Beitrag beantworten will, ist, wie diese einfachen kooperativen Entwicklungsprozesse erfolgreich gestaltet werden können. Wie sehen effektive und vor allem effiziente Vorgehensweisen in der Kommunikation, Koordination und Kooperation aus? Und welche potenziellen Probleme gilt es zu vermeiden?

2 Kooperative Softwareentwicklung

Softwareentwicklungsprozesse bedingen eine intensive Zusammenarbeit aller Beteiligten und sind Prozesse, bei denen zeitliche, räumliche und organisatorische Verteiltheit eine zunehmende Rolle spielen. Die Entwicklung findet zudem typischerweise in Teams statt, die zwar ein gemeinsames Ziel – die Ent-

wicklung der Software – aber auch unterschiedlichste Qualifikationen oder verschiedenste organisatorische Verankerungen besitzen.¹ Damit betreffen wichtige Voraussetzungen für die erfolgreiche Entwicklung von komplexer Software nicht nur technische Faktoren, „sondern vor allem auch den Prozeß der Aufteilung der Entwicklungsaufgaben und -aktivitäten innerhalb eines Projektteams, den Aufbau von Kommunikations- und Koordinationsbeziehungen und die Unterstützung von kreativen Problemlösungsprozessen“.² Softwareentwicklung nicht nur als technischen Entwicklungsprozess, sondern als kooperativen Kommunikationsprozess^{3,4} oder gar Bildungsprozess⁵ zu verstehen, ist seit ca. 15 Jahren eine wichtige Perspektive der Informatik auf das Thema Entwicklungsprozesse. Aus dieser Perspektive heraus lässt sich der Begriff kooperative Softwareentwicklung mit den Worten von Josef Altmann und Gustav Pomberger wie folgt definieren:

„Kooperative Softwareentwicklung umfaßt die Abdeckung der Kommunikations- und Koordinationsbedarfe innerhalb eines Softwareentwicklungsprozesses, die für die Planung, Durchführung und Abstimmung aller aufgabenbezogenen, zeitlich und räumlich verteilten Aktivitäten erforderlich sind. Kooperative Softwareentwicklung umfaßt dementsprechend alle prozeß- und produktbezogenen Aktivitäten aller Beteiligten, deren gemeinsames Ziel die Erstellung eines Softwareproduktes ist.“⁶

Die drei charakteristischen Merkmale, die die beiden Autoren für kooperative Entwicklungsprozesse identifiziert haben, sind:

- *Die aufgabenbezogene Verteilung*, durch die mittels einer geeigneten Koordinationsstrategie die aufgabenbezogene Aufteilung von spezifischen Teilprozessen anhand der Fähigkeiten und Möglichkeiten der beteiligten Personen erfolgt.
- *Die zeitliche Verteilung*, die mit Hilfe von synchronen

und asynchronen Kommunikationsmitteln die Tätigkeiten sequentiell oder nebenläufig aufteilt.

- *Die räumliche Verteilung*, bei der zusätzlich eine Aufteilung nach organisatorischen oder wirtschaftlichen Gründen erfolgt.

Alle drei Merkmale benötigen die genaue Spezifikation von Kommunikations-, Koordinations- und Kooperationsbedarfen sowie die Bereitstellung einer Infrastruktur, die diese Bedarfe deckt.

3 Open Source Entwicklungen als Spezialfall für kooperative Softwareentwicklungen

Open Source Entwicklungen stellen einen Spezialfall im Hinblick auf kooperative Softwareentwicklung dar, die gerade im Bibliothekswesen immer mehr an Bedeutung gewinnen^{7,8} und für die in ganz besonderem Maße eine organisatorische und technische Unterstützung der drei genannten Merkmale notwendig sind.

Das liegt vor allem an der hohen Überschneidung der charakteristischen Merkmale von Open Source Entwicklungsprozessen und den allgemeinen Merkmalen von kooperativen Entwicklungsprozessen. Open Source Entwicklungsprozesse weisen nämlich mit Verteiltheit, Offenheit und Agilität⁹ typische Merkmale auf, die den Kommunikations-, Koordinations- und Kooperationsbedarf kooperativer Entwicklungsprozesse sogar noch verschärfen.

Da ist zum einen der Aspekt der Verteiltheit als zentrales Charakteristikum. Dieses Merkmal weist im Hinblick auf Open Source Entwicklungsprozesse weniger auf eine räumliche Verteiltheit der beteiligten Personen hin, sondern vielmehr auf eine zeitliche und vor allem organisatorische Verteiltheit¹⁰. Typisch für Open Source Entwicklungsprozesse sind parallel oder synchron verlaufende Entwicklungsaktivitäten im Rahmen von Entwicklungsprojekten, die von unterschiedlichen am Entwicklungsprozess beteiligten Organisationen ausgeführt werden. Diese müssen dann wechselseitig in Einklang gebracht werden. Damit existiert aufgrund dieses Merkmals eine sehr hohe Anforderungsüberdeckung mit kooperativen Softwareentwicklungsprozessen.

Zum anderen zeichnet Open Source Entwicklungs-

1 Tietze, Daniel A. / Schümmer, Till: Kooperative Softwareentwicklung, in Schwabe, Gerhard / Streitz, Norbert / Unland, Rainer (Hrsg.): Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten. Springer, Berlin / Heidelberg 2001, S. 264-275.

2 Altmann, Josef / Pomberger, Gustav: „Kooperative Softwareentwicklung: Konzepte, Modelle und Werkzeuge“ in Nüttgens, Markus / Scheer, August-Wilhelm (Hrsg.): Electronic Business Engineering. Physica-Verlag HD, 1999, S. 643-664.

3 Schewe, Bettina: Kooperative Softwareentwicklung. Deutscher Universitätsverlag, Wiesbaden 1999. DOI: 10.1007/978-3-322-85820-7_2

4 Altmann, Pomberger: Kooperative Softwareentwicklung: Konzepte, Modelle und Werkzeuge, 1999.

5 Zorn, Isabel / Büschenfeldt, Maika / Schelhowe, Heidi: Kooperative Softwareentwicklung einer Sekretariatsplattform als Bildungsprozess, in Herczeg, Michale / Kindschüller, Martin Christof (Hrsg.): Mensch und Computer 2008: Viel Mehr Interaktion. Oldenbourg Verlag, München 2008, S. 337-346.

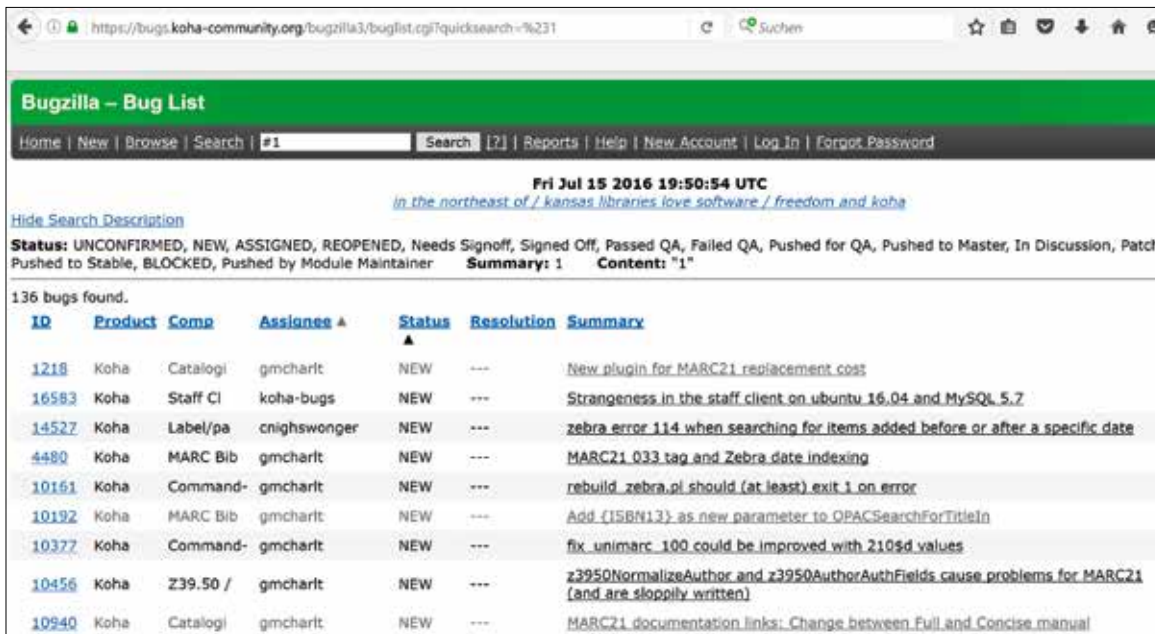
6 Altmann, Pomberger: Kooperative Softwareentwicklung: Konzepte, Modelle und Werkzeuge, 1999. S. 652.

7 Lohmeier, Felix / Seige, Leander: Open Source Bibliotheksmanagement. Vortrag auf der Konferenz der Leiterinnen und Leiter der sächsischen Hochschulbibliotheken, Dresden 2014, <http://de.slideshare.net/f.lohmeier/open-source-bibliotheksmanagement-mit-dswarm-amsl> [15. Juli 2016]

8 Mittelbach, Jens: Zur Zukunft von Bibliothekssoftware, in: BIS – Das Magazin der Bibliotheken in Sachsen 4 Nr. 1 (2011), S. 26-28.

9 Finck, Matthias: Usability-Engineering in der Open-Source-Softwareentwicklung – Perspektiven, Vorgehensweisen und Techniken, Diss. Hamburg, 2007.

10 ebda



ID	Product	Comp	Assignee	Status	Resolution	Summary
1218	Koha	Catalogi	gmcharlt	NEW	---	New plugin for MARC21 replacement cost
16583	Koha	Staff Cl	koha-bugs	NEW	---	Strangeness in the staff client on ubuntu 16.04 and MySQL 5.7
14527	Koha	Label/pa	cnighswonger	NEW	---	zebra error 114 when searching for items added before or after a specific date
4480	Koha	MARC Bib	gmcharlt	NEW	---	MARC21 033 tag and Zebra date indexing
10161	Koha	Command-	gmcharlt	NEW	---	rebuild_zebra.pl should (at least) exit 1 on error
10192	Koha	MARC Bib	gmcharlt	NEW	---	Add (ISBN13) as new parameter to OPACSearchForTitleIn
10377	Koha	Command-	gmcharlt	NEW	---	fix_unimarc_100 could be improved with 210\$d values
10456	Koha	Z39.50 /	gmcharlt	NEW	---	z3950NormalizeAuthor and z3950AuthorAuthFields cause problems for MARC21 (and are sloppily written)
10940	Koha	Catalogi	gmcharlt	NEW	---	MARC21 documentation links: Change between Full and Concise manual

Abbildung 1:
„Bugtracker der
Projekts Koha“

prozesse eine besondere Form der Offenheit aus. Sowohl in der Nutzung als auch in der Entwicklung muss ein gewisses Maß an Offenheit vorhanden sein. Offenheit in der Nutzung besagt, dass die in der Entwicklung intendierte Nutzung sich nicht auf einen Kontext beschränkt, und Offenheit in der Entwicklung bedeutet, dass es Dritten freisteht, sich in den Entwicklungsprozess einzubringen.¹¹ Dabei bedeutet Offenheit in der Entwicklung nicht, dass einzelne im Entwicklungsprozess verankerte Projekte offen sind, aber es ermöglicht die grundsätzliche Beteiligung, so dass viele Projekte sich zwangsläufig stärker öffnen, um z. B. Parallelentwicklungen zu vermeiden. Dieses Merkmal verstärkt die allgemeinen Anforderungen an kooperative Entwicklungsprozesse, da sich der Bedarf an Abstimmungsprozessen und -werkzeugen weiter erhöht.

4 Beispiele für große kooperative Softwareentwicklungen im Bibliothekswesen

Es ist aufgrund der Ausgangsvoraussetzungen deshalb kein Zufall, dass die prominenten Beispiele für kooperative Softwareentwicklungen in Bibliotheken in der Regel Open Source Entwicklungen betreffen. Natürlich ist kooperative Softwareentwicklung in gewisser Weise zentraler Bestandteil des deutschen Bibliothekswesens. Die großen Bibliotheksverbände in Deutschland, die das Kooperationsmotiv wie beim Kooperativen Bibliotheksverbund Berlin-Brandenburg (KOBV) sogar teilweise im Namen tragen, haben alle auch die zentrale Aufgabe, Softwareentwicklung koordiniert voranzutreiben. Somit ist der Abstimmungs-

bedarf zwischen gleichberechtigten Stakeholdern seit Jahren durch die Verbände etabliert. Allerdings handelt es sich bei den Entwicklungen in den Verbänden zumeist um zentralisierte Entwicklungsprozesse, die nur hinsichtlich der Anforderungsanalyse Merkmale eines kooperativen Entwicklungsprozesses aufweisen. Wirklich kooperative Entwicklungsprozesse mit den Herausforderungen auf allen Ebenen der Prozessgestaltung finden sich eher im Open Source Bereich.

Als ein Beispiel für einen großen kooperativen Entwicklungsprozess mag das webbasierte integrierte Open-Source-Bibliothekssystem Koha dienen. Das Bibliothekssystem, das über die wesentlichen Module für Katalogisierung, Ausleihe, Zeitschriftenverwaltung, Erwerbung und OPAC verfügt, wurde seit 1999 zunächst für den Horowhenua Library Trust in Neuseeland entwickelt und dann als Open Source für die Weiterentwicklung in einer internationalen Gemeinschaft freigegeben.¹² Das System ist mittlerweile von mehr als 250 Entwicklern beständig weiterentwickelt worden und weltweit in mehr als 2.500 Bibliotheken unterschiedlicher Größe und Ausrichtung im Einsatz.¹³ In Deutschland bietet das Bibliotheksservice-Zentrum Baden-Württemberg (BSZ) Hosting und Support von Koha an.¹⁴ Neben etablierten Organisations-

¹² <https://koha-community.org/about/history/> [15. Juli 2016]

¹³ Fischer, Katrin: Kooperative Softwareentwicklung - das Koha-Community-Modell. Vortragsfolien vom 15. BSZ-Kolloquium am 30.09.2014 in der Universität Stuttgart 2014. <https://swop.bsz-bw.de/frontdoor/index/index/docId/1008> [15. Juli 2016]

¹⁴ Rajski, Beate / Feldsien-Sudhaus, Inken / Horst, Dora / Katzner, Erika / Weier, Heiko / Zeumer, Tobias: Koha-Evaluation durch die Universitätsbibliothek der TUHH (= TUBdok). Technische Universität Hamburg-Harburg, Hamburg 2015, doi:10.15480/882.1236 [15. Juli 2016].

¹¹ ebda



Abbildung 2:
Website
des Projekts
Kitodo

strukturen wie gewählten Release Teams wird in dem Projekt auch eine komplexe technische Infrastruktur – bestehend aus vielen einzelnen Komponenten – zur Verfügung gestellt, um den Kommunikations- und Kooperationsbedarfen Rechnung zu tragen. Dazu gehören u.a. eine Versionsverwaltung, ein Bugtracker, ein Wiki, eine Mailingliste oder eine eigene Plattform für Übersetzer der Software.

Kuali OLE („Open Library Environment“) ist als eine ursprünglich in den USA entwickelte Open Source Software für ein integriertes Bibliotheksmanagement¹⁵ ein weiteres Beispiel für einen erfolgreichen kooperativen Entwicklungsprozess. Die Entwicklung wird mittlerweile maßgeblich koordiniert und getragen von der Kuali Foundation, die 2006 als „not-for-profit“-Organisation mit dem Ziel gegründet wurde, die Zusammenarbeit in der Entwicklung und Erweiterung von Unternehmenssoftware-Anwendungen für den universitären Bereich zu fördern.¹⁶ Die eigentliche Entwicklung von Kuali OLE begann im Jahr 2008 – zunächst noch ohne Beteiligung der Kuali Foundation. Unter der Führung der Duke University Libraries wurde ein integriertes Bibliotheksmanagementsystem entwickelt, das den Anspruch hatte, open-source-basiert und nach den Prinzipien „Flexibilität“, „Gemeinschaftseigentum“, „serviceorientated architecture“, „Integration der Unternehmensebene“, „Effizienz“ und „Nachhaltigkeit“ ausgerichtet zu sein.¹⁷ Eine Gemeinschaft aus wissenschaftlichen Bibliotheken, die in das neue System investieren wollten, entwickelte ein Modell für einen kooperativen Entwicklungsprozess, der auf einem partnerschaftlichen, nachhaltigen Geschäftsmodell auf Basis einer Non-Profit-Organisationsstruktur beruht. Ähnlich wie bei dem Projekt Koha gibt es eine

Vielzahl von Entwicklern, die durch eine umfangreiche technische Infrastruktur bestehend aus unterschiedlichsten Tools unterstützt werden.

Neben den großen internationalen Entwicklungsprojekten gibt es auch in Deutschland verschiedene erfolgreiche kooperative Entwicklungsprojekte. Als Beispiel sei hier das Projekt Kitodo¹⁸ genannt, das – früher unter dem Namen Goobi bekannt – die Produktion, Präsentation und Archivierung von digitalen Objekten unterstützt¹⁹. Kitodo ist Open Source und wird federführend

von Bibliotheken selbst entwickelt. Die Software wird mittlerweile von zahlreichen Bibliotheken nachgenutzt – insgesamt in über 40 Bibliotheken, Archiven und Museen – in Deutschland, Großbritannien, den Niederlanden, Österreich und Spanien. Wie in den beiden internationalen Beispielen hat sich auch bei Kitodo eine professionelle Struktur zur Organisation des kooperativen Entwicklungsprozesses gebildet. Im Interesse eines verbesserten Rechts- und Investitionsschutzes sowie einer höheren Organisationskraft der Anwendergemeinschaft haben 12 Bibliotheken und Unternehmen 2012 einen Verein gegründet, der das Release-Management professionell organisiert und die Kommunikation zwischen den einzelnen Mitgliedern koordiniert²⁰. Mittlerweile sind in dem Verein mehr als 30 Bibliotheken und professionelle Dienstleister organisiert, um den Entwicklungsprozess von Kitodo voranzutreiben. Entsprechend der anderen Beispiele existiert auch hier eine ganze Reihe technischer Tools wie ein Wiki, eine Versionsverwaltung oder eine Mailingliste, die die technische Infrastruktur für das Projekt bilden.

Alle drei Projekte sind langfristig angelegte Entwicklungsvorhaben, die mit entsprechenden Ressourcen als kooperative Entwicklungsprojekte aufgesetzt wurden. Mit dem Projekt beluga core wird im Folgenden ein Beispiel im Detail vorgestellt, in dem vier Bibliotheken kurzfristig und mit geringen Mitteln ebenfalls einen kooperativen Entwicklungsprozess aufgesetzt haben, um mit beluga eine bereits existierende Lösung auf Basis der Open Source Software vufind gemeinsam weiterzuentwickeln. In diesem Projekt wurden keine dauerhaften Kooperationsstrukturen angelegt – es war auch erst einmal nur auf ca. ein

¹⁵ <https://www.kuali.org/> [15. Juli 2016]

¹⁶ Kemner-Heek, Kirstin: Konzeption und Angebot zukünftiger Bibliotheksmanagementsysteme: Bestandsaufnahme und Analyse, Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft; 64, Köln 2012. <http://www.fbi.fh-koeln.de/institut/papers/kabi/volltexte/band064.pdf> [15. Juli 2016]

¹⁷ Ebd., S. 91.

¹⁸ <http://www.kitodo.org/> [15. Juli 2016]

¹⁹ Bonte, Achim: „Fünf Jahre Goobi – das Rückgrat der Massendigitalisierung der SLUB“. BIS – Das Magazin der Bibliotheken in Sachsen. 2 (2012), S. 84–86.

²⁰ Neuböck, Gregor: „Oberösterreichische Landesbibliothek: Neues Leben für alte Bücher – oder wie man ein Digitalisierungsprojekt auf Schiene bringt!“. Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen & Bibliothekare 66.1 (2013), S. 179–191.

Jahr ausgelegt. Als Fallbeispiel zeigt das Projekt beluga core aber exemplarisch, welches Mindestmaß an Tools und Projektmanagementstrukturen es auch im Kleinen für einen kooperativen Entwicklungsprozess braucht.

5 Das Projekt beluga core

Im Rahmen eines gemeinsamen Entwicklungsprojekts haben die wissenschaftlichen Bibliotheken in Hamburg ab 2007 begonnen, unter der Federführung der Staats- und Universitätsbibliothek Hamburg (SUB Hamburg) ein gemeinsames Discoverysystem mit dem Namen beluga aufzubauen.²¹ Beluga wurde konzipiert als eine mit Web 2.0-typischen Funktionen ausgestattete Rechercheplattform, die im Vergleich zum herkömmlichen Bibliothekskatalog die Recherche in Büchern, Zeitschriften und Aufsätzen in den Hamburger Bibliotheken vereinfachen und zudem die virtuellen Lernräume und sozialen Netzwerke mit Literaturinformationen und digitalen Texten versorgen sollte. Mittlerweile setzt beluga auf 11 Bibliothekssystemen auf und umfasst mehr als 13 Mio. Datensätze.²²

Schon beim Bau des ersten Prototyps entschied sich die SUB Hamburg aus strategischen Gründen, auf die Offenheit des Systems und des Entwicklungsprozesses sowie die Nachnutzbarkeit für die Verwendung von Open-Source-Technologien zu setzen²³. Seit 2011 kommt die Open-Source-Software vufind als technische Grundlage für beluga zum Einsatz, weil vufind sich als Open-Source-Alternative zu den proprietären Katalogsystemen international und auch deutschlandweit etabliert hatte.²⁴

Seit 2015 wird auch in diesem Entwicklungsprozess ein Konzept verfolgt, das darauf abzielt, möglichst vielen Bibliotheken eine niedrigschwellige Nachnutzung zu ermöglichen und somit den Entwicklungsprozess als kooperativen Entwicklungsprozess zu gestalten. Unter dem Arbeitstitel beluga core wurde dieser Entwicklungsprozess von der SUB Hamburg gemeinsam mit den Universitätsbibliotheken in Lüneburg, Hildesheim und Braunschweig begonnen. Dabei wurde der Prozess möglichst leichtgewichtig gehalten. Es werden vor allem drei Werkzeuge eingesetzt,

um die typischen Verteilungsaspekte eines kooperativen Entwicklungsprozesses mit minimalem Aufwand und trotzdem adäquat zu adressieren:

- **Gemeinsames Repository:** Auf der Basis von GIT²⁵ wurde ein gemeinsames Repository für das Projekt eingerichtet, in dem die einzelnen Entwicklungen der Standorte parallel vorangetrieben werden.
- **Gemeinsames Ticketsystem:** Mit redmine steht ein Ticketsystem zur Verfügung, über das Fehler und Weiterentwicklungswünsche dokumentiert und strukturiert werden.
- **Gemeinsame Kommunikationsplattform:** Mit CommSy²⁶ kommt eine einfache webbasierte Kooperationsplattform zum Einsatz, die es den am Prozess Beteiligten erlaubt zu kommunizieren, indem z.B. Diskussionsforen genutzt werden, sich zu koordinieren, indem z.B. die Terminverwaltung, -planung und -protokollierung über das System erfolgt, und zu kollaborieren, indem z.B. gemeinsame Dokumente für Entwicklungskonzepte erstellt werden.

Mit GIT und redmine existieren somit zwei Werkzeuge zur gemeinsamen Arbeit auf Entwicklungsebene und mit CommSy eines zur gemeinsamen Arbeit auf Ebene der konzeptionellen Zusammenarbeit auf persönlicher Ebene.

Gestützt wird dieser Entwicklungsprozess von Jour-Fixe-Terminen, die monatlich stattfinden. Von jedem Partner ist der Projektleiter an diesen Terminen anwesend, um dem Abstimmungsbedarf innerhalb des Teams gerecht zu werden. Diese Termine finden überwiegend in Face-to-Face-Meetings statt, um die zeitliche und räumliche Verteilung zumindest teilweise zu überwinden und um vor Ort die wesentlichen Planungsentscheidungen gemeinsam zu treffen. Über dem Projektleitertreffen existiert noch eine Lenkungsgruppe, bestehend aus Vertretern der Bibliotheksleitungsgruppen, die bei Bedarf zusammengerufen wird.

Alle Bibliothekspartner haben zu Beginn des Projekts ein kleineres Entwicklungsbudget bereitgestellt, das jeweils nicht ausreichend gewesen wäre, um ein solches Vorhaben eigenständig durchzuführen. Für das gemeinsame Projekt ist die Summe hingegen auskömmlich gewesen, um notwendige Anpassungen und Weiterentwicklungen vorzunehmen. Dazu wurde ein gemeinsamer Releaseplan abgestimmt und die Aufwände für die einzelnen Umsetzungsschritte den jeweiligen Häusern zugeordnet.

21 Christensen, Anne: Katalog 2.0 im Eigenbau: Das beluga-Projekt der Hamburger Bibliotheken, in: Bergmann, Julia / Danowski, Patrick (Hrsg.) Handbuch Bibliothek 2.0., De Gruyter Verlag, 2010, S. 317-332.

22 Maas, Jan: beluga – ein konsortiales Discoverysystem auch für Bibliotheken außerhalb Hamburgs. Vortragsfolien Bibliothekartag 2016. urn:nbn:de:0290-opus4-23625 [15. Juli 2016].

23 Christensen: Katalog 2.0 im Eigenbau: Das beluga-Projekt der Hamburger Bibliotheken, 2010.

24 Breeding, Marshall: „Open Discovery Interface“ American Libraries, 6/7 (2009), S. 40-41.

25 <https://git-scm.com/> [15. Juli 2016]

26 <https://www.commsy.net/> [15. Juli 2016]

Abbildung 3:
Entwicklungs-
wünsche zu
beluga im
Ticketsystem
redmine

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Projekt
1147	Unterstützung	Neu	Normal	Korrekturvorschläge ("Meinten Sie...") auch anzeigen, wenn Ergebnisse vorhanden		14.07.2016 16:01	beluga 3.2
1145	Feature	Neu	Normal	Zitierung: Einbau von externem Dienst statt Anzeige der 3 Stile		14.07.2016 14:51	beluga 3.2
1143	Unterstützung	Neu	Normal	beluga-bs: bibtip doch in left_column konfigurierbar machen		14.07.2016 14:29	beluga 3.2
1142	Feature	Neu	Normal	beluga-bs: Alle Nutzer-Session-Elemente löschen, wenn sich Nutzer abmeldet		14.07.2016 14:16	beluga 3.2
1141	Unterstützung	Neu	Normal	beluga-alle: Benutzerkonto - Gebühren: Endsumme anzeigen		14.07.2016 13:57	beluga 3.2
1140	Unterstützung	Neu	Normal	beluga-alle: Erweiterte Suche und Facettenfilter in beiden Reiter auch bei unterschiedlich konfigurierten Indizes und Filtern		14.07.2016 13:39	beluga 3.2
1139	Feature	Neu	Normal	beluga-alle: Erweiterung Facetten-Funktionen		14.07.2016 13:34	beluga 3.2
1138	Unterstützung	Neu	Normal	beluga-alle: Detailsseite - Eingabefeld für x-ten		14.07.2016 13:19	beluga 3.2

Fazit

Nach weniger als einem Jahr sind die Weiterentwicklungen von beluga für die drei nachnutzenden Bibliotheken einsatzfähig. Alle vier Bibliotheken haben durch den kooperativen Entwicklungsprozess Doppelarbeit in ihren Häusern vermieden und so jetzt einen technischen Stand von beluga geschaffen, der durch eine alleinige Initiative weder zeitlich noch finanziell auf gleichem Niveau umsetzbar gewesen wäre.

Der gemeinsame Entwicklungsprozess bindet im Bereich der Zusammenarbeit sicherlich einiges an zeitlichen Ressourcen – insbesondere die regelmäßig stattfindenden Jour-Fix-Meetings. Aber durch das intensive Maß an Abstimmung verringert sich der gesamte Entwicklungsaufwand, da bereits in der Konzeptionsphase Fehler durch das Acht-Augen-Prinzip deutlich stärker vermieden werden. Und vor allem verringert sind der in jeder Bibliothek notwendige lokale Entwicklungsaufwand.

Die relative Offenheit und Flexibilität des Prozesses erlaubt es auch, dass jedes Haus sich in seiner ihm möglichen Geschwindigkeit in den Prozess einbringen kann. Sollte aufgrund personeller Kapazitätsengpässe eine Bibliothek für z.B. ein paar Monate sich nicht so intensiv in dem Prozess einbringen können, dann steht dem Haus am Ende des Zeitraums dennoch eine Weiterentwicklung zur Verfügung. Lediglich das Maß der Einflussnahme verringert sich.

Insgesamt zeigt das Projekt beluga core, dass es sich auch im Kleinen lohnt, auf kooperative Entwicklungsprozesse zu setzen. Auch ohne die weltweit unterstützende Community und eine professionell instanziierte

Projektstruktur hat diese Form der Zusammenarbeit Vorteile

- in Bezug auf die finanziellen Ressourcen in jedem beteiligten Haus,
- in Bezug auf die Softwarequalität und
- in Bezug auf die reduzierten Nachbesserungsaufwände in der Entwicklung durch die Perspektivenvielfalt in der Konzeption.

Das Projekt beluga core kann deshalb als positives Beispiel dafür dienen, dass es nur wenig Starthilfe braucht, um so einen einfachen kooperativen Entwicklungsprozess zu initiieren, und dass man mit wenigen personellen, finanziellen und technischen Hilfsmitteln einen solchen Prozess erfolgreich etablieren kann. Aber auch wenn die Ressourcen deutlich geringer ausfallen als bei einer individuellen Umsetzungsstrategie der einzelnen Häuser, so muss allen Beteiligten dennoch klar sein, dass alle drei Ressourcentypen (Geld, Personal und Zeit) im ausreichenden Maße zur Verfügung gestellt werden müssen. **I**



Prof. Dr. Matthias Finck

Professor für Usability Engineering
an der NORDAKADEMIE –
Hochschule der Wirtschaft und
Inhaber und Geschäftsführer der
effective WEBWORK GmbH
Köllner Chaussee 11
25337 Elmshorn
matthias.finck@nordakademie.de