

André Lahmann und Claas Kazzer

Barrierefreiheit in Bibliothekskatalogen

Die Verstetigung von Arbeitsergebnissen durch deren Rückfluss in die Community

An der Universitätsbibliothek Leipzig werden im Rahmen der *finc*-Nutzergemeinschaft Bibliothekskataloge mit der Discovery-Software VuFind® realisiert, deren Barrierefreiheit seit 2019 stark vorangetrieben wird. In diesem Erfahrungsbericht soll gezeigt werden, wie im Nachgang der erfolgreichen Überarbeitung der *finc*-Discovery-Portale hinsichtlich ihrer Barrierefreiheit nicht nur die Projektergebnisse, sondern auch eine verstetigte Praxis etabliert wurde, die Barrierefreiheit als einen weiteren, regulären Gesichtspunkt der Softwareentwicklung (neben UI/UX, Performance, Sicherheit, Features etc.) integriert. Nach einem kurzen Abriss des Projektverlaufs werden die notwendigen Workflows und die dadurch entstehenden Aufwände für das Zurückspielen barrierefreier Verbesserungen in den VuFind®-Community-Code erläutert und anschließend anhand von Fallbeispielen illustriert.

Die Rahmenbedingungen

Die Universitätsbibliothek Leipzig¹ stellt als technischer Betreiber im Rahmen der *finc*-Nutzergemeinschaft² Institutionen in ganz Deutschland 17 Discovery-Portale zur Verfügung. Hervorgegangen ist die *finc*-Nutzergemeinschaft aus einem 2011 mit Fördermitteln des Europäischen Fond für regionale Entwicklung (EFRE) an der Universitätsbibliothek Leipzig gestarteten Projekt, dessen Ziel es war, für 11 wissenschaftliche Bibliotheken in Sachsen mit der Open-Source-Software VuFind³ Discovery-Portale aufzubauen. Über das Web-Frontend hinaus, das den für die Nutzer*innen zugänglichen Teil des Katalogs ausmacht, werden der konsortial genutzte Suchmaschinenindex (Solr), die Anbindung unterschiedlicher Lokalsysteme (bisher Libero, Bibliotheca und Aleph) sowie Nutzungsstatistiken als Dienste angeboten. Stand Ende 2023 umfasst die *finc*-Nutzergemeinschaft 22 Mitglieder, die einen oder mehrere der angebotenen Dienste nutzen.

Die für die Discovery-Portale verwendete Open-Source-Software VuFind® wurde an der Villanova University in den US) entwickelt und wird seitdem von einer interna-

1 Universitätsbibliothek Leipzig: <https://www.ub.uni-leipzig.de> (31.05.2024).

2 Finc: Was ist *finc*? <https://finc.info/about> (31.05.2024).

3 VuFind: <https://vufind.org> (31.05.2024).

tionalen Community ständig erweitert und verbessert. So fließen auch die an der UB Leipzig entwickelten Features und Erweiterungen in einem kontinuierlichen Austausch mit der VuFind®-Community zurück und stehen damit der internationalen VuFind®-Anwendergemeinschaft zur Verfügung. Während Aspekte der Barrierefreiheit in den ersten Jahren der Entwicklung kaum Beachtung fanden, wird in den letzten Jahren intensiv daran gearbeitet, die Weboberfläche möglichst barrierearm zu gestalten. Zum Inkrafttreten der EU-Richtlinie 2016/2102 am 23. September 2020 sollten die finc-Discovery-Portale auf Barrierefreiheit geprüft und diese weitestgehend hergestellt werden. Auf internationaler Ebene – insbesondere in den USA und Kanada – sind erst kürzlich Regelungen in Kraft getreten, die Barrierefreiheit für Webportale öffentlicher Einrichtungen vorschreiben.⁴ Wie auch bei bibliothekarischen Features oder technologischen Verbesserungen der VuFind®-Software unterstützt sich die internationale Community gegenseitig durch das Einbringen barrierefreier Lösungen in den gemeinsam gepflegten Programmcode, sodass eine Nachnutzung auch niederschwellig möglich ist.

In diesem Erfahrungsbericht soll gezeigt werden, wie im Nachgang der erfolgreichen Überarbeitung der finc-Discovery-Portale hinsichtlich ihrer Barrierefreiheit nicht nur die Projektergebnisse, sondern auch eine verstetigte Praxis etabliert wurde, die Barrierefreiheit als einen weiteren, regulären Gesichtspunkt der Softwareentwicklung (neben UI/UX, Performance, Sicherheit, Features etc.) integriert. Nach einem kurzen Abriss des Projektverlaufs werden die notwendigen Workflows und dadurch entstehenden Aufwände für das Zurückspielen barrierefreier Verbesserungen in den VuFind®-Community-Code erläutert und anschließend anhand von Fallbeispielen illustriert.

Projekt „Implementierung barrierefreier Zugänge für bibliographische Kataloge sächsischer Hochschulbibliotheken“

Zielstellung des durch Mittel des Staatsministeriums für Wissenschaft und Kunst (SMWK) Sachsens ermöglichten Projekts „Implementierung barrierefreier Zugänge für bibliographische Kataloge sächsischer Hochschulbibliotheken“ an der UB Leipzig war die Herstellung der Barrierefreiheit in den Bibliothekskatalogen 13 sächsischer Hochschulbibliotheken der finc-Nutzergemeinschaft im Zeitraum 2019–2021. Zur Unterstüt-

⁴ Einen guten Überblick für die Situation in den USA und Kanada gaben Stephanie Leary und Andrea Buntz Neiman auf dem VuFind® Summit 2023 in Chicago. Die Folien des Vortrags in englischer Sprache können unter folgender URL abgerufen werden: https://vufind.org/wiki/community:conferences:summit_2023#stephanie_leary_and_andrea_buntz_neiman_equinox_open_library_initiative_-_layers_of_accessibility_in_open_sourcecreating_an_accessible_theme_for_vufind (31.05.2024).

zung des Projekts wurde auf die Beratung durch das Kompetenzzentrum für barrierefreie Informations- und Kommunikationsangebote in Sachsen (dzb lesen – BIKOSAX⁵) zurückgegriffen, sowie für jedes finc-Discovery-Portal eine abschließende Prüfung durch BIKOSAX durchgeführt. Zum Abschluss war vorgesehen, die im Rahmen des Projekts erarbeiteten Verbesserungen der finc-Discovery-Portale an die internationale VuFind®-Community zurückzuspielen, um eine globale Nachnutzung zu ermöglichen.

Projektverlauf

Trotz einiger Herausforderungen zu Projektbeginn, wie der generell schwierigen Personalgewinnung im IT-Bereich, insbesondere bei der Anforderung „Kenntnisse in barrierefreier Webseitenentwicklung“, sowie den im Laufe des Projekts aufkommenden Einschränkungen durch die COVID-19-Pandemie, konnte das Projekt mit nur geringem zeitlichen Verzug Ende 2021 abgeschlossen werden. In jedem der 13 zu überarbeitenden finc-Discovery-Portale wurde eine Vielzahl von Änderungen vorgenommen:

- Es wurde der semantische Aufbau der Webseiten überarbeitet und klar strukturiert, sodass sich aus der Struktur der visuell erfahrbare Inhalt erschließen lässt.
- Das Layout der Webseiten wurde insbesondere für Menschen mit Sehbeeinträchtigungen durch besseren Kontrast, Schriftgröße, Vergrößerbarkeit (Zoom) etc. optimiert. Hierbei wurde auch das Responsive Design der Webseiten verbessert, also die dynamische Anpassung des Webseitenlayouts an den Dimensionen des Ausgabegeräts. Die Verwendung von Screenreadern auf den Webseiten wurde dadurch verbessert, dass erläuternde Texte (z. B. durch ARIA-Codes) konsequent angewandt wurden, wo sie für die Bedienung und das Verständnis der Webseite notwendig sind. Weiterhin wurde eine konsequente Bedienbarkeit der Webseiten über die Tastatur umgesetzt.
- Wies ein finc-Discovery-Portal individuelle Features auf, mussten diese auch auf Barrierefreiheit geprüft und ggf. umgebaut werden. Zuletzt wurden in Zusammenarbeit mit den Bibliotheken Erklärungen zur Barrierefreiheit des jeweiligen finc-Discovery-Portals integriert.

In einem letzten Schritt wurden die Umsetzungen in jedem finc-Discovery-Portal durch BIKOSAX geprüft und ein Kurzbericht angefertigt. Die darin noch identifizierten Mängel bzgl. der Barrierefreiheit wurden in einer abschließenden Überarbeitung behoben und das finc-Discovery-Portal anschließend in den produktiven Betrieb überführt.

⁵ Dzb lesen: Digitale Barrierefreiheit unterstützt von BIKOSAX. <https://www.dzblesen.de/auftraege/digitales> (31.05.2024).

Erfahrungen aus der Projektlaufzeit

Wie eingangs erwähnt stellte sich der Projektauftritt aufgrund der allgemeinen Personalsituation und insbesondere wegen fehlender Vorkenntnisse zu barrierefreier Webseitenentwicklung als schwierig dar. So zeigte sich auch, dass das Thema Barrierefreiheit insgesamt einen hohen Einarbeitungsaufwand erfordert, der nicht zuletzt dadurch gekennzeichnet ist, sich eine notwendige Grundeinstellung oder ein Mindset anzueignen, dass Barrierefreiheit bei jeder Weiterentwicklung der Software miteinbezieht. Dabei sind eindeutige Lösungen häufig nicht sofort und in einigen Fällen sogar überhaupt nicht erreichbar. Eine pauschale, ausführliche Auszeichnung aller Elemente einer Webseite zum Beispiel würde dazu führen, dass die über einen Screenreader vermittelten Informationen womöglich die Nutzer*innen überfordern. Nur stellte sich dann bei der notwendigen Reduktion der Informationen die Frage nach der Relevanz für Menschen mit Behinderungen, vor allem, wenn es innerhalb des Entwicklungsteams keine Person mit Einschränkungen gab. Zwar können von den Entwickler*innen Tools wie Screenreader genutzt werden, um sich einen Eindruck vermitteln zu lassen, wie eine Webseite mittels Screenreader erfahrbar ist – es dürfte aber zweifelhaft sein, dass ein solches technologiegestütztes Hineinversetzen einer tatsächlichen Situation für die Nutzung assistiver Tools entspricht.

Dementsprechend zeigte sich im Projektverlauf also, dass Barrierefreiheit viele Diskussionen zu potenziellen Lösungen innerhalb des Projektteams, Austausch mit externen Projektpartnern (wie Bibliotheken oder BIKOSAX), aber auch ein Abwägen zwischen Vorgaben, bspw. des Corporate Designs der Einrichtungen und Anforderungen der Barrierefreiheit, erfordert. Diese zeitintensiven Prozesse müssen darüber hinaus angemessen dokumentiert werden, damit gefundene Lösungen auch zu einem späteren Zeitpunkt nachvollziehbar bleiben – d. h. zu dokumentieren, weshalb die konkret vorliegende Lösung gewählt wurde und welche anderen sich als nicht optimal erwiesen haben.

Die Ergebnisse dieser zum Teil langwierigen Prozesse aus Nachhaltigkeitsgründen nicht nur den finc Discovery-Portalen zu Gute kommen zu lassen, sondern auch der internationalen VuFind®-Community, war deshalb auch ein erklärtes Ziel des Projekts. Bereits während der Schlussphase des Projekts wurde mit dem Zurückspielen der barrierefreien Lösungen an die VuFind®-Community begonnen und dies wird seitdem fortgeführt. Das Projekt ist damit zwar abgeschlossen, die Verbesserung der Barrierefreiheit setzt sich jedoch in der Interaktion mit der VuFind®-Community stetig fort.

Fortsetzung durch Zurückspielen der Verbesserungen zur Barrierefreiheit an VuFind®-Community

Pull Requests

Die Verwendung und aktive Weiterentwicklung von Open-Source-Software zeichnet sich nicht durch Kostenersparnis, sondern durch Kompetenzgewinn und Flexibilität aus. Schließlich werden nicht anfallende Lizenzierungskosten durch die stattdessen benötigten Personalmittel zumindest aufgebraucht, allerdings gewinnt die Organisation durch das eingesetzte Personal die Kompetenz und Flexibilität auf geänderte Anforderungen zu reagieren und diese in den eigenen Produkten auch abzubilden. Ein weiterer, nicht zu vernachlässigender Vorteil von Open-Source-Software ist darüber hinaus die Verteilung der Last der notwendigen, stetigen Weiterentwicklung auf mehrere Schultern, sollte die betreffende Open-Source-Software über eine vitale Community verfügen. Im Falle von VuFind® trifft letzteres eindeutig zu, sodass die Einbringung der barrierefreien Lösungen aus dem besprochenen Projekt nicht nur den Effekt der Nachhaltigkeit im Sinne einer Nachnutzung haben sollte, sondern auch den der nachhaltigen und verteilten Weiterentwicklung der barrierefreien Lösungen. Da die Weiterentwicklungen der Software VuFind® auch regelmäßig in die finc-Discovery-Portale integriert werden, fließen so bereits in den finc-Discovery-Portalen umgesetzte Verbesserungen der Barrierefreiheit aus dem VuFind®-Community Code zurück und führen zu einer Reduzierung der für die finc-Discovery-Portale notwendigen Anpassungen in VuFind®.

Damit eine Änderung an VuFind® (sei es nun bzgl. der Barrierefreiheit oder eines sonstigen Features) „upstream“ übernommen wird (d. h. in den Quellcode von VuFind® integriert), muss ein Prozess durchlaufen werden. Der Quellcode von VuFind® wird auf dem Webportal Github verwaltet. Dabei kommt das Versionierungstool „git“ zum Einsatz, das sämtliche Änderungen an den Quellcodedateien von VuFind® transparent protokolliert. Soll nun eine Änderung upstream übernommen werden, wird die Änderung im Quellcode vorgenommen und der Änderungswunsch als „Pull Request“ auf Github hochgeladen (s. Abb. 1, Schritte 1–3).

Die im Pull Request enthaltenen Änderungen werden anschließend von Entwickler*innen der VuFind®-Community begutachtet und kommentiert (s. Abb. 1, Schritt 4). In diesem Schritt zeigen sich die gleichen Erfahrungen aus dem Projekt: Selbst bei kleinen Änderungen sind zum Teil umfangreiche Diskussionen notwendig, da Lösungen zur Erhöhung der Barrierefreiheit selten eindeutig sind. Die zeitintensiven Abstimmungen mit der Community, dem Team, externen Expert*innen usw. führen außerdem häufig zu anderen Ergebnissen als dem ursprünglich gestellten Pull Request. Ist ein Konsens für den Pull Request gefunden, wird dieser dann letztendlich im finalen Schritt in den Quellcode von VuFind® übernommen (s. Abb. 1, Schritt 5). Beim nächsten Update der finc-Dis-

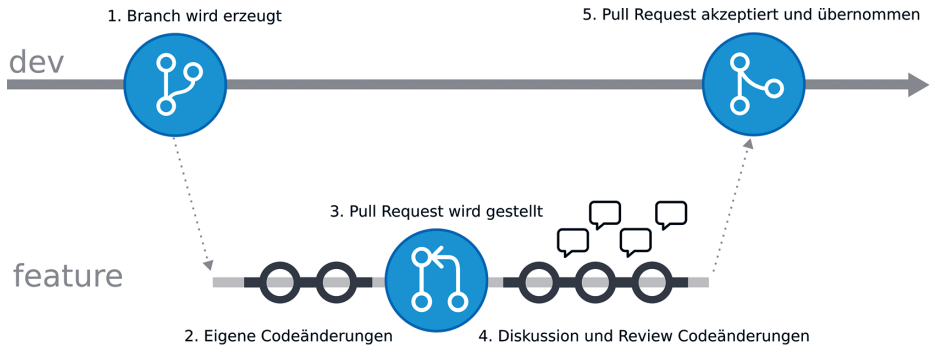


Abb. 1: Workflow des Zurückspiels von Änderungen in VuFind® auf Github.

covery-Portale mit der aktuellen VuFind®-Version würden die Codeänderungen aus dem Pull Request dann wieder „downstream“ in die finc-Discovery-Portale einfließen.

Erfahrungen aus zwei Jahren Pull Requests zu Barrierefreiheit

Wie bereits erwähnt, ist dieser Prozess häufig langwierig, da gefundene Lösungen mit der internationalen Community nochmals ausgehandelt werden müssen. Nicht selten kommt dabei ein anderes Ergebnis heraus, als das initial intendierte. Meistens stellen die Resultate jedoch eine weitere Verbesserung dar. Kritisch ist allerdings der durch den langwierigen Aushandlungsprozess notwendige Koordinationsaufwand, weshalb in der VuFind®-Community bereits die Notwendigkeit einer für Barrierefreiheit verantwortlichen Person mit entsprechendem Entscheidungsmandat diskutiert wurde.

Wie sich hier also auch zeigt, muss Barrierefreiheit in Software als zentrale und fortwährende Herausforderung verstanden werden. Barrierefreiheit ist ein Prozess, der bei der Softwareentwicklung berücksichtigt werden muss. Dementsprechend muss Barrierefreiheit in automatisierten Tests geprüft, sowie in den Workflows verankert werden. All das erfordert zusätzliche Ressourcen, vor allem personell. Dabei kann die Nutzung von Open-Source-Software vorteilhaft sein, da hier eine Verteilung der benötigten Ressourcen auf mehrere Schultern möglich ist – eine vitale Community vorausgesetzt. Das folgende Fallbeispiel illustriert den erläuterten Workflow sowie die daraus resultierenden Erfahrungen abschließend.

Fallbeispiel – Pull Request zur Verbesserung der Facetten

Im Frühjahr 2022 nahmen meine Kolleg*innen und ich die Verbesserung der „Facetten“ in Angriff. Als Facetten bezeichnen wir Bereiche, in denen nachträglich Filter zu einer

Suche ausgewählt werden können. Meist werden diese Bereiche optisch in einer Randleiste platziert (s. Abb. 2). Suche ich zum Beispiel nach dem Begriff „Barrierefreiheit“ und bekomme zu unscharfe Ergebnisse, kann ich die Suche dadurch weiter eingrenzen, dass ich zum Beispiel den Facettenwert „Online-Ressourcen“ wähle. Damit werden mir dann nur Ergebnisse geliefert, auf die sowohl der Suchbegriff, als auch der Facettenwert zutrifft. Facettenwerte können kombiniert werden. So könnte ich in unserem Beispiel die Treffschärfe weiter erhöhen, indem ich zusätzliche Formatfacetten, wie „Artikel, E-Artikel“ oder „Buch, E-Book“ auswähle, usw. Natürlich kann ich eine Suche auch einschränken, indem ich Facettenwerte ausschließe, also z. B. die Werte „Online-Ressourcen“ mit „nicht Buch, E-Book“ kombiniere, sodass mir alle Medien angezeigt werden, die zwar Online-Ressourcen sind, aber keine elektronischen Bücher.

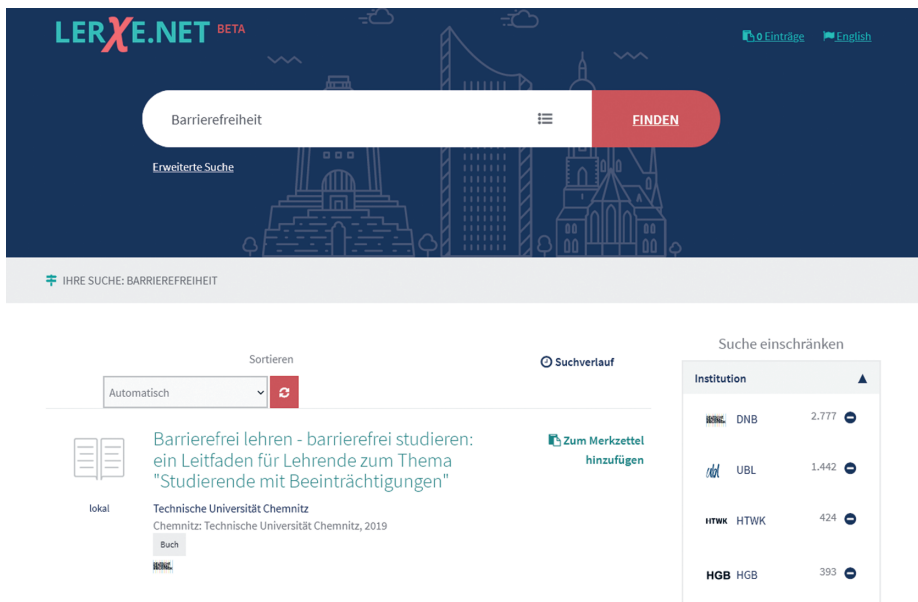


Abb. 2: Oberer Teil der Ergebnisseite des Leipziger Recherche-Netzwerks mit rechts angeordneten Facetten.

Die Tester*innen der Prüfstelle für Barrierefreiheit am dzb lesen, Team BIKOSAX, hatten Folgendes kritisiert: Beim Aktivieren oder Deaktivieren eines Facettenwerts wird die Seite neu geladen, jedoch ohne Warnung. Durch das Neu-Laden wird der Wert zwar in die Suche integriert, aber mein Fokus (oder der meines Bildschirmlesegeräts) geht verloren, ohne dass ich das beim Bedienen der Facetten wissen kann. Wenn mein Fokus z. B. gerade auf dem Facettenwert „Online-Ressourcen“ war und ich diesen aktiviert habe, ist mein Fokus nach dem Neu-Laden wieder am Seitenanfang. Das ist sowohl verwirrend, als möglicherweise auch frustrierend.

Unsere Idee: Wir versehen die Facetten mit einem Hinweistext.

Gesagt, getan: Als Erstes setzten wir die Änderungen in den Katalogen der Universität Leipzig und des Leipziger Recherche-Netzwerks um, um sie auch live testen zu können. Dann programmierte ich die Änderungen in den Basis-Code der VuFind®-Community und reichte den Änderungswunsch – Pull Request – ein. Der Kopfbereich eines Pull Requests sieht aus wie in der Abb. 3 und nennt ein paar interessante Zahlen (hier rot umrandet).

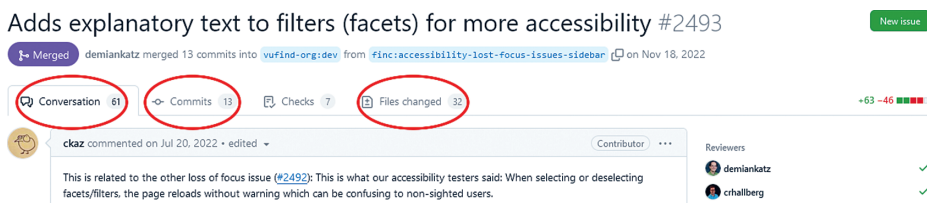


Abb. 3: Bildschirmfoto des oberen Teils des Pull Requests zur Verbesserung der Facetten auf der Kommunikationsplattform der VuFind-Community.

Um eine Lösung zu finden, mit der alle Beteiligten zufrieden waren, brauchte es diesmal 61 Diskussionsbeiträge und 13 Code-Änderungen in 32 Dateien. Der gesamte Prozess dauerte 4 Monate, von Juli bis November, was verglichen mit anderen Pull Requests recht wenig ist, zumal die Sommerferien dazwischen lagen.

Besucht man heute (Stand November 2023) den Katalog des Leipziger Recherche-Netzwerks,⁶ kann man feststellen, dass Bildschirmlesegeräte hier teilweise noch viel zu „schwatzhaft“ sind. Grund dafür ist, dass wir in diesem Katalog noch eine ältere Version der Katalog-Software und damit unter anderem auch die Originalversion des Pull Requests verbaut haben. In dieser war ich eindeutig über das Ziel hinausgeschossen. Hier liest der Screenreader, wenn man zum Beispiel den Facettenwert „DNB“ für „Deutsche Nationalbibliothek“ fokussiert:

Hauptsprungmarke – Suche einschränken – Überschrift Ebene 2 – Institution – Zum Einschränken der Ergebnisse können Sie unter folgenden Filtern wählen – schwarzes, nach oben weisendes Dreieck – ausgeklappt – Linkliste – DNB – Seite wird bei Auswahl des Filters DNB neu geladen – Link – Seite wird bei Auswahl des Filters DNB neu geladen – zwei 777 Treffer – Filter DNB ausschließen – Seite wird neu geladen, wenn der Filter DNB aus den Suchergebnissen ausgeschlossen wird

Und das wiederholt sich bei jedem Facetten-Wert! Je mehr Facetten ein Katalog also hat, desto mehr zusätzlicher, unsichtbarer Text taucht auf der Seite auf.

Meine Absicht war, dass der Screenreader über das Verhalten der Website informieren sollte, egal welche Facette ausgewählt oder ausgeschlossen wird. Dass das viel zu viel Text ergibt, fiel erst während der Diskussion des Pull Requests so richtig auf.

⁶ Lerxe: Ein Recherchenetzwerk für den Wissenschaftsraum Leipzig. <https://lerxe.net/> (31.05.2024).

Nach einer längeren Debatte, ob meine die beste Lösung sei oder wie man dem Problem noch besser beikommen könnte, hatte mein Kollege Chris Hallberg von der Villanova University in Pennsylvania (USA) die Idee, nur die Facetten-Hauptüberschrift mit dem Hinweis auszustatten. Perfekt!

Auch wenn die teils langen Diskussionen und vielen kleinen Nachbesserungsforderungen in der Community manchmal anstrengend sind, zeigt sich hier ein sehr positiver Effekt. Das Team ist international und seine Mitglieder bringen ihre eigenen, manchmal sehr unterschiedlichen Sichtweisen ein. Daher die teils langen Debatten. Ohne die Diskussion jedoch wäre es entweder zu keiner Verbesserung der Facetten gekommen oder meine erste Version wäre nur bei uns lokal eingebaut worden. Nach der Auseinandersetzung mit der Community und weiteren kleinen Verbesserungen – also nach den erwähnten 61 Diskussionsbeiträgen und 13 Code-Änderungen –, steht der erklärende Text jetzt nur einmal auf der Seite. Jetzt liest der Screenreader in unserem Beispiel:

Hauptsprungmarke – Suche einschränken – Überschrift Ebene 2 – Institution – Seite wird neu geladen, wenn Filter aktiviert oder ausgeschlossen wird

Im Anschluss an die erfolgreiche Übernahme in den Community-Code müssen wir die gefundene Lösung natürlich noch in die restlichen Kataloge einbauen, die wir betreuen. Das kann in dringenden Fällen direkt passieren oder aber spätestens, wenn im Zuge der turnusmäßigen Updates die Software-Version in den Katalog eingespielt wird, die den Pull Request enthält.

Auf genau diesem Weg ist die Änderung inzwischen in mehreren Katalogen in den verschiedensten Ecken der Welt angekommen, ohne dass sich deren Programmierer notwendigerweise überhaupt mit Barrierearmut auseinandersetzen mussten. Dazu zählen (natürlich) der Beispiel-Katalog von VuFind®, aber auch der Katalog der Uni Michigan,⁷ oder der der Midlands State University in Zimbabwe.⁸ Weitere werden folgen.

Lassen Sie mich noch einmal zusammenfassen: Die vorgestellte Änderung bewirkt nur, dass man beim Aktivieren oder Deaktivieren von Facettenwerten gewarnt wird, dass die Seite neu laden wird. Es ist also eine minimale Verbesserung. Durch die Diskussionen in diesem und einem weiteren Pull Request wurde klar, dass es sinnvoll wäre, eine Version der Ergebnisliste zu erstellen, bei der nur die Liste selbst aktualisiert und nicht die ganze Seite neu geladen wird. Damit geht der Fokus bei Aus- oder Abwahl einer Facette nicht mehr verloren. Dieser sehr komplexen Änderung hat sich einer der engagiertesten VuFind®-Entwickler aus Finnland angenommen.

Was dieses Beispiel ansatzweise zeigt, ist, wie der Prozess des Zurückspielens von Code an eine internationale Nutzergemeinschaft funktioniert. An eine Nutzergemeinschaft, wohlgemerkt, die quelloffene Software benutzt – also Software, bei der jede

⁷ Michigan State University: MSU Libraries. <https://catalog.lib.msu.edu/> (31.05.2024).

⁸ Midlands State University: Libraries. <https://vufind.library.msu.ac.zw/vufind/> (31.05.2024).

Person den Quellcode einsehen und bearbeiten kann. Der Prozess ist oft langwierig, aber er lohnt sich letztlich sowohl für alle direkt Beteiligten, als auch für alle, die die verbesserte Software schon heute nutzen oder in Zukunft nutzen werden.

Autoren

Claas Kazzer (Abschnitt 4) ist Oberflächenprogrammierer an der Universitätsbibliothek Leipzig. Er arbeitet mit den Schwerpunkten Usability und Barrierefreiheit an der Entwicklung der finc-Kataloge mit und trägt die Erkenntnisse und Lösungen des Leipziger Teams aktiv in die VuFind®-Community.

André Lahmann (Abschnitte 2 und 3) ist wissenschaftlicher Bibliothekar und Produktmanager der finc-Kataloge an der Universitätsbibliothek Leipzig. Er koordinierte das Projekt „Implementierung barrierefreier Zugänge für bibliographische Kataloge sächsischer Hochschulbibliotheken“ von 2019–2021 an der UB Leipzig und überführte den Aspekt Barrierefreiheit in die kontinuierliche Weiterentwicklung der finc-Kataloge.