JOURNAL OF NEW LIBRARIANSHIP
Vol. 9, No. 2, 2024
https://doi.org/10.33011/newlibs/17/13

Peer-Reviewed Article

Open-Source Migrations: Perspectives from an Early Career Librarian

Emma C. Beck, *University of Louisville*

ABSTRACT

This paper examines open-source migrations from the early career librarian perspective. After being on the implementation team for two open-source migrations in two years, the author noticed stark similarities in the two experiences. The paper will discuss the challenges all migrations face and challenges more specific to open-source systems. These include lack of resources, changing timelines, and lack of system sustainability. The paper will then move on to provide recommendations for other early career librarians who are implementing open-source systems. The paper will also offer ways colleagues and supervisors who are not working on the implementation can support these migrations and the people.

KEYWORDS

Open-source, migrations, entry level, Samvera, FOLIO

SUGGESTED CITATION

Beck, E. (2024). Open-source migrations: Perspectives from an early career librarian. *Journal of New Librarianship, 9*(2), 101–114. https://doi.org/10.33011/newlibs/17/13

A growing trend in libraries to adopt more open-source software, whether it is for digital repositories or the Integrated Library System (ILS), which is the modern-day library catalog. Open-source software is software where the source code is available for anyone to view and contribute. This allows for more flexibility and control over a system. The alternative is a proprietary system where a single company develops the system. This reduces the amount of control users have in regard to development, cost increases, and if a company chooses to stop supporting upgrades.

The trend of open-source migrations has been documented for more than a decade, noting the correlation between the mission of libraries to provide free and universally available information with free universally available source code (Morgan 2005).

The author of this article is in her early career and has been on the implementation team for two open-source system migrations within 18 months. Her first migration was an ILS migration at Mount Holyoke College (part of the Five Colleges Consortium) from Aleph to the library service platform Future of Libraries is Open (FOLIO). The second migration happened at the University of Louisville, her current place of employment. She started the job as the archives and special collections were finishing a migration of their digital collections from CONTENTdm to Samvera's Hyku.

Through this experience she saw similar challenges with both systems and communities. This paper will give a voice to those challenges and provide recommendations to early career librarians and other staff. The paper will start by addressing issues faced in most migrations, open source included, before moving on to problems acutely faced by early career professionals involved in open-source migrations. Following that, it will provide recommendations specific to early career librarians on how to navigate the open-source landscape and its potential difficulties. The paper will conclude with suggestions for supporting colleagues during open-source migrations.

This article considers an early career librarian to be a librarian with ten years of experience or less, including paraprofessional and pre-MLIS experience (Thomas et al., 2019).

Challenges in All Migrations

A literature review by Sharma and Khan (2021) found that open-source scholarship primarily focused on the benefits of open-source software. Because literature is dominated by case studies and progress, it can be difficult to accurately predict the challenges different institutions might face during their migrations. A more specific example of this is exemplified in the case study of Cornell launching FOLIO, which discusses their migration. It includes mention of 22 "showstopper" issues which delayed their migration yet went into no detail about the specific challenges and how they were resolved (Colt and Howell 2021). This dearth of scholarship on the potential obstacles of open-source migration can be especially challenging for early career colleagues, who may lack both the experience to respond to such issues, and the professional connections necessary to seek advice, and thus rely more heavily upon publications on the topic.

The gap in scholarship could be explained in various ways. Not all librarians are faculty, and thus do not focus on scholarship about their work. We could also theorize that librarians do not focus on these challenges in scholarship because the profession typically focuses on the positives in situations. Middle or later-career librarians rely on their network and various other communication channels when facing problems, especially during a migration. People are more transparent about challenges and recommendations when not in writing. This includes conferences and casual conversations where much of the daily work is discussed. For early-career librarians the network is not as wide, and they might have less access to conferences. This leads to a larger reliance on published scholarship, which is focused primarily on positives instead of constructive recommendations you hear in verbal conversations.

Customization

Customization is a double-edged sword. An often-cited benefit of open-source systems is the ability to alter the system, adding features that make it easier for users to navigate the interface. Each institution will have various user needs and collection needs depending on the number of collections, metadata, content, or format. Customization can promise solutions to all the niche needs of an institution. To do so, however, requires a lot of work and money. As Colt and Howell (2021) perfectly state, "One of the greatest benefits of FOLIO is that it is completely customizable through its settings, and one of the most difficult things about FOLIO is that it is completely customizable through its settings." This customization can go in both directions.

The core code of open-source systems is available on GitHub, a platform to share written code, but for any sort of customization you must have a software developer build the custom code. This cost to customize can be misleading because it does not appear on the price tag of the migration. The Five Colleges Consortium's migration to FOLIO highlighted this. As a consortium, each of the five colleges had unique needs, so the customizable nature of FOLIO was one of its primary appeals. However, the unique needs of each institution could only be fulfilled if time and money were put into the system. For example, each of the institutions needed to have separate billing systems. This was not originally possible, so it had to be customized.

Challenging Timeline

Most migrations, regardless of whether they are proprietary or open source, will experience delays. These delays include, but are not limited to, lack of funding, slow development, or systems issues. Instead of coming as a complete surprise, delays should be built into the timeline. During the Five College Consortium migration to FOLIO, the initial go-live date was summer 2020, but with the lack of system readiness, the migration was continually postponed until being implemented in July 2022. This resulted in confusion for staff and users, and reduced trust in the system before it was even implemented. Cornell experienced a similar delayed migration discussed in their case study (Colt & Howell, 2021).

The migration to Hyku at the University of Louisville was very similar to the timelines described above. The initial expected launch was sometime in 2021, but slow development meant the system migration was complete in August 2023. Obviously, the exact issues could not have been predicted, but discussing the possibility of setbacks sooner would have helped adjust expectations. One suggested method is using benchmarks to establish system readiness, which helped adjust the timeline in advance (Colt & Howell 2021).

Accessibility

Another setback many open-source systems have is the lack of accessibility built into the core code, which can create issues adding them in at a later date. The Americans with Disabilities Act (ADA) requires that websites, online courses, eBooks, and other technologies must be accessible to people with disabilities if the business is open to the public (Wall & Sarver, 2023). In addition to being legally required, accessible features benefit more than just those with disabilities. In the physical realm, we can see strollers benefiting from ramps built for wheelchair users. A website that is accessible for a screen reader is also typically more user friendly for all visitors.

Despite this clear ADA requirement and increased user experience, we still see development that does not comply with these standards. This lack of compliance is seen regularly as companies launch a product they claim to be “production ready” despite knowing it has accessibility issues (de Vries, 2023). If a system or product has known accessibility issues, it should not be considered ready.

The split between production ready and accessible is pervasive in the open-source community because the demand for development is not matched by developers (Klug et al., 2021). This is also true regarding accessibility; staff must prioritize development and unfortunately accessibility often takes the back burner. When multiple implementers experience this pressure, there is no one prioritizing the accessibility features.

Sustainability

System sustainability is another large concern with most open-source projects. In the open-source community, sustainability is the use of a product now and in the future by users and developers through changes (Liao et al., 2018). This can be challenging as interest and commitment vary from organization to organization and individual to individual. It is common to see a handful of core users leading the conversation, and thus development, in a given community. This is the same for service providers; there may be only one or two who will do software development work on the project. This can be an issue if the community is too heavily reliant on a single company to build the product.

In Hyku, there was only one company that could be hired for system development. They did not intend this to happen, but the lack of other software companies in the market means there is only one company to turn to for development. Thus, if prices are raised, there is no ability to negotiate or seek another quote. Open-source systems are built to allow anyone to assist in development, and while technically this is true, if there is only one company you can contract it can still function like a monopoly, regardless of whether it was intentional or accidental. A monopoly goes directly against the ethos of open source and is not a sustainable model.

This reliance on a company for development parallels to the reliance open-source systems have on the individuals who contribute time to the projects. If these colleagues were to leave the project or their jobs, the community would be negatively impacted (Klug et al., 2021). It is increasingly common for open-source projects to fund a paid position to coordinate contributions from community volunteers. A prime example of this is the Samvera community, which has pooled money to have a project manager to help continue development ("Community Leadership - The Community - Samvera").

In the FOLIO community there is a newly created position of FOLIO Developer Advocate which will work to increase communication and identify gaps in developer knowledge and work (FOLIO Developer Advocate Job Description). Knowing someone is there to propel the projects can help ease concerns that the project will fall by the wayside. A community-paid position can also mitigate the issues with service providers through a more cohesive development plan.

Challenges Specific to Early Career Employees

The challenges above are universal to open-source migrations and may also be present in proprietary systems. This section will focus on challenges specific to early career librarians. With how technical systems can be, the literature can be challenging to read, and the audience is typically more advanced users. The literature would benefit from more perspectives in the conversation. Early career librarians in all fields are underrepresented in literature. A literature review by Sharma and Khan (2021) found age and open-source systems was underrepresented in the field. Scholarship about open-source migrations primarily center around navigating and learning an open-source system. Having experience with two open-source migrations within the span of about a year demonstrated to the author that these sorts of challenges are not unique to one system and as such would likely be encountered by other early career librarians.

When introduced to an open-source community, newcomers will quickly realize how complex and full of background the conversations are. This could be with colleagues on the implementation team or in the special interest groups (SIGs). No one is intentionally obtuse or exclusive, but it can be difficult to know how to begin participating in a community that predates your membership. The community members already have a working history, familiarity with the system(s) in use, and a knowledge of library technologies. For early career librarians, a robust understanding of library technologies is not yet there, let alone open-source technology. In SIG meetings, specialized terminology unique to the open-source system is regularly used, and the history and development of the system is often discussed. Project-specific jargon can be challenging to pick up on and learn, and newcomers would benefit from having documentation to help close the gap.

System development and history is exacerbated by outdated user documentation, introduction, and glossary pages. The lack of documentation in open-source communities extends far beyond the library sphere. A survey in 2017 of open-source contributors found that 93 percent of respondents saw documentation as a pervasive problem (Open Source Survey, 2017). Even if this is an issue for everyone, it provides an even bigger barrier for those first entering an open-source project (Mendez et al., 2018). Outdated pages cannot aid in learning a system without asking regular direct and specific questions. While non-existent information pages are a challenge, sometimes outdated pages can be even worse as the information they contain may no longer be accurate or relevant. For example, one way to learn a system is to see it already implemented and to see another instance of the system. However, if the only way to find other implementations is by a go-live date listed on the community's website, then you will probably receive incorrect information, because, as discussed above, migrations are regularly delayed.

Entry-level colleagues will all experience challenges, but these challenges grow and change when looking at racial and gender minorities. The race and gender divides are especially prominent in the open-source community (Lin & Besten, 2019). While white women dominate the field of librarianship, the open-source library community does not have the same demographics (Bosu & Sultana, 2019). The 2018 study by Mendez et al. found that a majority of the barriers they identified were exacerbated due to gender bias. In addition, they found there were unique "cultural differences" experienced by racial minorities.

Women and people of color are more affected by imposter syndrome. Imposter syndrome is an internalized feeling of not being successful despite objectively being a high achiever. It is especially prominent in early career professionals (Martinez & Forrey, 2019) and is a very similar feeling to being a phony or fraud. Imposter syndrome can leave employees, especially entry-level ones, with reduced confidence and feeling anxious (Lacey and Parlette-Stewart 2017). Imposter syndrome is a spectrum, so the effects can vary drastically by individual and is influenced by other factors such as race, gender, and socioeconomic status (Gottlieb 2023). All these feelings, of lack of confidence, and being an outsider can be exacerbated in an open-source community, which is a space dominated by men where complex terminology and acronyms are regularly used. This can have a negative effect on the individual by deterring them from engaging with the community and a negative effect on the community which would benefit from more diverse conversations and backgrounds.

Migrations force a pattern of "hurry up and wait." One week the workload is astronomical. In addition to daily responsibilities, a developer needs a feature tested and the sooner it is reviewed, the sooner they can work on the next steps. When contracting out, the institution does not have power over the timeline and if development has taken time to reach testing, there can be a pressure to test it as soon as possible as not to hold up development. Once the testing is complete and feedback is given, the waiting returns as the software engineers are working on the code or an upload needs to run. This abrupt cycle of intense periods of work interspersed with lighter periods of work will continue, so embrace the calmer periods.

Entry-level staff are often late additions to the implementation team and thus have no point of reference and often were not part of the decision-making process that landed them in the current open-source landscape. This lack of history with the previous system can be a benefit and a challenge. While it might be hard to be critical about the system without knowing the benefits and downfalls of the current system, it also means a bias from another platform does not obscure the feedback. This can lead to a more user-focused perspective which can improve user experience.

Recommendations for Open-Source Migrations

Immersing yourself in an open-source community is overwhelming. Acronyms and terminology are abundant, increasing the challenge of breaking into the community. Even after feeling established, months later a new project or communication channel will be brought to light. It is not a flaw to miss context, it is the nature of a large project. Keep interacting and asking questions, and with time knowledge will develop.

This section will go into detail about the following recommendations for entry-level staff migrating to open-source systems:

- Ask questions
- Do not assume you are the problem
- Familiarize yourself with the tools
- Use driver/navigator method
- Solicit feedback
- Speak up
- Get involved
- Find someone to talk to

Ask Questions

As discussed above, it is common for entry-level employees, regardless of the profession, to have imposter syndrome. Individuals who suffer from imposter syndrome might mistakenly assume that colleagues who have been with their institution longer, or with different expertise, will have already thought of everything worth implementing or customizing. This reasoning negates the influence entry-level employees can have on open-source systems, and thus on users who will use the system. In addition, imposter syndrome can affect everyone regardless of where they are in their career (Clark et al., 2014). Thus, developing habits working around these feelings will serve an individual through their work life.

Open-source systems are built by software engineers. Even former library employees who are helping develop the back end of the open-source system are looking through a different lens than back-end developers. Front-end and entry-level employees add a new and valuable perspective in the community, one that is often closer to the front-end user. Fresh perspectives are valuable in open-source communities, as are diverse experiences. Everyone can contribute something different to the conversation, even if it is only questions. Even obvious questions to you get overlooked by developers (Nowogrodzki, 2019). These questions will either add to the development, build your understanding, or increase communication about an area lacking.

When encountering an issue in the system, do not assume you are the problem. Especially with newer open-source systems, the problem might be a bug. Developers and system administrators tend to look at the system based on how they use it, whereas staff, especially front-end staff, use the system differently. Therefore, it is not uncommon for a feature to work as planned in a test, but not actually be tested according to how typical or different users might access said feature. If something is confusing, regardless of whether it is a bug, it should be improved. If you, a staff member, are having trouble, users or students will as well.

Learn the Tools

As you enter the open-source community, familiarize yourself with the tools that are used. Slack, GitHub, and Confluence are key technologies often used. GitHub is a tool to share written code with others who might find it helpful and adapt the code for a different use case (Piotrowski & Marzec, 2023). The ease of sharing code is particularly helpful in an open-source environment because it is so central to the mission in open-source communities. Slack is a communication channel, though it is not specific to open-source systems. It facilitates quick conversations, allowing people to chime in as necessary, in a way that is not always possible over email. Confluence is a platform often used to hold documentation for those working on a project, such as for an open-source system, institution, or department.

Some people enter open-source communities knowing Slack, GitHub, and/or Confluence. Some institutions do not use these tools; thus, staff are learning new communication methods in addition to the open-source system. If you are a beginner, do not feel deterred in having to learn new technologies to keep up with development. They often come easier to you the more you use them, and you do not need to be an expert to navigate these tools.

A specific training tool is the driver and navigator method, ideal for a larger group of people learning the system. This technique involves having someone unfamiliar with the system share their screen and move the mouse, as the person who is more familiar talks through the steps and tells the driver exactly where to click. This forces the presentation to slow down and ensures each step is explained clearly. The driver and mouse technique can quickly demonstrate where there are usability concerns that were not obvious before and can prove especially helpful when training staff.

Engagement

Solicit feedback from staff not working on the project. This applies to any migration; implementation teams can get tunnel vision and lose perspective. With system familiarity, you understand why certain decisions were made and thus no longer see the same issues. You never know which feature that seems so clear to back-end staff could use explanation to front-end staff or public users (Nowogrodzki, 2019). Occasionally including other colleagues in the conversation will increase usability, especially if their ideas can be brought back to the community.

For entry-level employees battling imposter syndrome, speaking up in open-source spaces can be a challenge, but getting involved is critical. Not only does sharing perspectives help you learn the system and the community, but open-source communities are built around engagement (Klug et al., 2021). They are founded on volunteer work, service, and altruism (Baytiyeh & Pfaffman, 2019). Use your strengths to find a way to contribute, which will help you learn and will help the community develop.

Involvement can start as joining meetings, which will slowly expose you to more calls for service, whether it is writing documentation or writing code. The author found her strengths in organization and assisting with administrative duties, an area the Samvera Metadata Interest Group (SMIG) needed assistance. The pre-existing co-facilitator knew the history of the project development and documentation and thus can answer technical answers the community might ask. The co-facilitators have different experiences and strengths but lead to a well-rounded team.

While involvement can be a larger commitment in the case of a co-facilitator, it can also be a smaller commitment, like helping with testing. In the FOLIO community, there are quarterly bug tests to review features. Developers do not test the system in the same way as users, thus front-end testers are critical to the product working. In addition, testing features from a fresh perspective is more akin to the experience patrons might have. Open source will not work on its own. It works only when we all work.

Support System

Finally, find someone to talk to. This could be a mentor, a supervisor, a colleague, or a peer either at your institution or not. They do not need to have the same job or experience as you. They do not need to help navigate the landscape. They do not need to even understand the work you are doing, but having a support system to lean on will help navigate challenges. There is a massive amount of research that shows the benefits of mentorship, both officially and unofficially in the short and long run (Ehrich et al., 2004). Leaning into this and building your network will help you in the short and long term.

Ways to Provide Support during a Migration

For those not currently on the implementation team, there are ample ways to support colleagues undertaking this work. This section will focus on the ways colleagues not working on an implementation team can support their colleagues who are. Migrations take a massive amount of work. Finding the time to do both a migration and your job is a challenge. One way to support colleagues actively working on a migration is by offering to offload some of their other work. For example, if a committee needs a departmental representative, someone else stepping up is appreciated.

Regardless of how the migration goes, it always uses more time and resources than anticipated. Helping to advocate for more resources – be it money or time – goes a long way. In line with advocating for resources, it also helps to dispel the myth that open source is cheaper, or a cost cutting solution. Open source can be just as expensive, if not more expensive than proprietary technology. It is a common refrain when talking about open-source software to hear “free” as in “free speech,” not “free beer” (Free Software Foundation, 2024). Open-source systems are not just a one-time gift, they are a commitment and a lifestyle. When people assume there will be a budget surplus with a migration, they minimize issues the migration team will face.

At some institutions, migrations are a top-down decision. When this happens, increasing transparency from administrators and supervisors provides clarity about the nature of the work. This openness also builds trust and can help smooth the transition. Communicating about what open source is, why it is important, and bringing an overall awareness to the open-source mission will build some of that trust (Singh, 2013). The ethos of open source is central to a lot of work, and the more colleagues understand that the more likely they will be invested in a positive outcome.

Finally, to all the colleagues, supervisors, and administrators out there, celebrate your implementation team! Celebrate big and small accomplishments. Open-source migrations are a beast. They are challenging and drag on for years. Celebrating the arduous work helps everyone recognize the magnitude of the project.

Conclusion

In conclusion, the open-source landscape can be incredibly challenging for anyone to navigate, but it provides unique issues to entry-level library employees. These challenges, such as lack of institutional knowledge or history in an open-source community, are exacerbated by the outdated or lack of documentation. This forces people, especially early career librarians, to ask questions as they arise. While many of these issues are unlikely to improve on their own, we must still work to make open-source systems more accessible regarding both ADA compliance and community usability. When accessibility is improved, all users benefit. An individual can mitigate these challenges by getting involved in the community, asking questions, and building a support system. There are no quick fixes to these problems, but with time and individual commitment, open-source systems can be more welcoming to newcomers.

References

Baytiyeh, H., & Pfaffman, J. (2010). Open source software: A community of altruists. *Computers in Human Behavior*, 26(6), 1345–1354. https://doi.org/10.1016/j.chb.2010.04.008

Berry, S. (2023, March 31). *What is ADA compliance? (And what does ADA compliance mean for your website?)*. WebFX. https://www.webfx.com/blog/marketing/what-is-ada-compliance/

Bosu, A., & Sultana, K. Z. (2019). Diversity and inclusion in open source software (OSS) projects: Where do we stand? In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM*), 1–11. https://doi.org/10.1109/ESEM.2019.8870179

Clark, M., Vardeman, K., & Barba, S. (2014). Perceived inadequacy: A study of the imposter phenomenon among college and research librarians. *College & Research Libraries, 75*(3), 255-271. https://doi.org/10.5860/crl12-423

Colt, M., & Howell, D. (2021). Cornell Library FOLIO case study. *International Journal of Librarianship*, *6*(2), 13-20. https://doi.org/10.23974/ijol.2021.vol6.2.205

Community Leadership—The Community—Samvera. (n.d.). Samvera. Retrieved December 15, 2023, from https://samvera.org/the-community/community-leadership

De Joode, R. V. W. (2004). Innovation in open source communities through processes of variation and selection. *Knowledge, Technology & Policy*, *16*(4), 30–45. https://doi.org/10.1007/s12130-004-1013-4

de Vries, H. (2023). Ableist interactions. *Hidde's Blog*. https://hidde.blog/interactions-about-accessibility/

Ehrich, L. C., Hansford, B., & Tennent, L. (2004). Formal mentoring programs in education and other professions: A review of the literature. *Educational Administration Quarterly*, *40*(4), 518–540. https://doi.org/10.1177/0013161X04267118

Farrell, J. (2023). How to make data open? Stop overlooking librarians. *Nature*, *624*(7991), 227–227. https://doi.org/10.1038/d41586-023-03935-1

FOLIO Developer Advocate Job Description. (n.d.). Retrieved June 4, 2024, from https://docs.google.com/document/d/1esuP2nNke051LJ9PcmfiuG8z-E4ZlJyxMsNGtJQU-jk/edit#heading=h.8i5ugj2ekwv9

Free Software Foundation. (2024, January 1). What is free software? *GNU Project*. https://www.gnu.org/philosophy/free-sw.html

Geiger, R. S., Varoquaux, N., Mazel-Cabasse, C., & Holdgraf, C. (2018). The types, roles, and practices of documentation in data analytics open source software libraries: A collaborative ethnography of documentation work. *Computer Supported Cooperative Work (CSCW)*, *27*(3–6), 767–802. https://doi.org/10.1007/s10606-018-9333-1

Gottlieb, M. (2023). When I say … imposter syndrome. *Medical Education*, *57*(11), 1008–1009. https://doi.org/10.1111/medu.15160

Guidance on Web Accessibility and the ADA. (2023, December 1). ADA.Gov. https://www.ada.gov/resources/web-guidance/

Jones, S., Lampert, C., Lapworth, E., & Shaw, S. (2023). Islandora for archival access and discovery. *The Code4Lib Journal*, *58*. https://journal.code4lib.org/articles/17929

Klug, D., Bogart, C., & Herbsleb, J. D. (2021). "They can only ever guide:" How an open source software community uses roadmaps to coordinate effort. In *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1), 1–28. https://doi.org/10.1145/3449232

Lacey, S., & Parlette-Stewart, M. (2017). Jumping into the deep: Imposter syndrome, defining success and the new librarian. *Partnership: The Canadian Journal of Library and Information Practice and Research*, *12*(1). https://doi.org/10.21083/partnership.v12i1.3979

Leonard, E. (2018). Dream the impossible dream: A case study of U.S. Federal website accessibility standards compliance at Seton Hall University Libraries. *International Information & Library Review*, *50*(1), 34–39. https://doi.org/10.1080/10572317.2018.1422908

Liao, Z., Deng, L., Fan, X., Zhang, Y., Liu, H., Qi, X., & Zhou, Y. (2018). Empirical research on the evaluation model and method of sustainability of the open source ecosystem. *Symmetry*, *10*(12), 747. https://doi.org/10.3390/sym10120747

Lin, Y., & Den Besten, M. (2019). Gendered work culture in free/libre open source software development. *Gender, Work & Organization*, *26*(7), 1017–1031. https://doi.org/10.1111/gwao.12255

Locke, J. L., & Wright, B. (2022). History can be open source: Democratic dreams and the rise of digital history. *The American Historical Review*, *126*(4), 1485–1511. https://doi.org/10.1093/ahr/rhab534

Martinez, J., & Forrey, M. (2019). Overcoming imposter syndrome: The adventures of two new instruction librarians. *Reference Services Review*, *47*(3), 331–342. https://doi.org/10.1108/RSR-03-2019-0021

Mendez, C., Susmita Padala, H., Steine-Hanson, Z., Hilderbrand, C., Horvath, A., Hill, C. G., Simpson, L., Patil, N., Sarma, A., & Burnett, M. (2018). Open-source barriers to entry, revisited: A sociotechnical perspective. In *Proceedings of ICSE '18: 40th International*

Conference on Software Engineering, *ICSE '18*. https://doi.org/10.1145/3180155.3180241

Morgan, E. L. (2005). *Open source software in libraries*. LITA. http://infomotions.com/musings/ossnlibraries-lita/

Nowogrodzki, A. (2019). How to support open-source software and stay sane. *Nature*, *571*(7763), 133–134. https://doi.org/10.1038/d41586-019-02046-0

Open Source Survey. (2017). Open source survey. *GitHub Open Source*. https://opensourcesurvey.org/2017/

Ostergaard, K. (2015). Accessibility from scratch: One library's journey to prioritize the accessibility of electronic information resources. *The Serials Librarian*, *69*(2), 155–168. https://doi.org/10.1080/0361526X.2015.1069777

Palmer, A., & Choi, N. (2014). The current state of library open source software research: A descriptive literature review and classification. *Library Hi Tech*, *32*(1), 11–27. https://doi.org/10.1108/LHT-05-2013-0056

Piotrowski, D. M., & Marzec, P. (2023). Digital curation and open-source software in LAM-related publications. *Journal of Librarianship and Information Science*, *55*(4), 935–947. https://doi.org/10.1177/09610006221113372

Sharma, J., and Khan, S. (2021). Open source software adoption in libraries – A literature review study. *Library Philosophy and Practice*. https://digitalcommons.unl.edu/libphilprac/6383

Singh, V. (2013). Experiences of migrating to an open- source integrated library system. *Information Technology and Libraries*, *32*(1): 36-53. https://doi.org/10.6017/ital.v32i1.2268

Thomas, C., Trucks, E., & Kouns, H. B. (2019). Preparing early career librarians for leadership and management: A feminist critique. *In the Library with the Lead Pipe*. https://www.inthelibrarywiththeleadpipe.org/2019/early-career-leadership-and-management/

Wall, P. S., and Sarver, L. (2003). Disabled student access in an era of technology. *The Internet and Higher Education*, *6*(3), 277–284. https://doi.org/10.1016/S1096-7516(03)00046-0