Agile Open-Source Discovery: Blacklight with Ebsco Discovery Service

In 2013, after several years of an unfocused discovery strategy relying on licensed, proprietary software, University of Alberta Libraries (UAL) developed a new, more focused discovery strategy. This strategy proposed implementing an open-source discovery solution, as well as adopting processes, methodologies, and tools from the world of agile software development.[1] Along with new IT governance processes, this signaled a major change on the part of our systems department, encompassing not only hardware and software infrastructure, but also our thinking around configuration, software design, testing, and user-experience (UX). Moving in this new direction also necessitated changes in work culture, including changes in workflow, team composition, collaboration, and decision-making processes.

The Discovery Systems Librarian position was created in September 2011. At first, the position's role was ambiguous: the discovery strategy at the time amounted to "license as many discovery systems as we can afford, and put them all on the library home page." At the time we presented the traditional OPAC as the default library search, OCLC's WorldCat Local and Ebsco's Discovery Service (EDS). That fall, we licensed Ex Libris' Primo system, to be implemented in 2012. The presentation of three different discovery systems on the library website was not intuitive, and a separate flowchart was required to help users figure out which system they should use in a given situation.

Initially, the new Discovery Systems Librarian role was to implement Primo, which I began to do early in 2012. 2012/2013, however, turned out to be quite disruptive. In April, the Chief Librarian stepped down and the provincial government, which funds most post-secondary education in Alberta, heavily cut the university's budget. By the time a new Chief Librarian had been found, the library had already implemented some budget reduction measures, including a voluntary severance program, reduced hours and temporary closure of some library branches, and centralization/consolidation of collections. As part of this process, the library's Information Technology Services (ITS) unit decided to cancel the unfinished Primo implementation and, in order to make the discovery services on offer more intuitive (as well as to

[1] For an overview of agile, see http://en.wikipedia.org/wiki/Agile_software_development

save money), also canceled our WorldCat Local license. In addition, it was recognized that in 2013 an ARL library should not be presenting a traditional OPAC as its default search interface, so in the autumn of 2013, we replaced the OPAC with an Ebsco Discovery Service search box, presenting the traditional OPAC, along with Google Scholar, as alternative search tools.

The switch to presenting EDS as the default library search was a step in the right direction, but it still did not fulfill all our student and researcher needs. A clear strategy still needed to be developed not only to give us direction as we continued to work on discovery services, but also in order to deal with data silos that had grown organically over the years. These collections included, for example, non-upgraded bibliographic records, a database of maps, historical curricula records acquired through web archiving, and small collections belonging to research groups on campus. These silos were not candidates for inclusion in licensed discovery systems for a variety of reasons. Some of them contained data that was not structured enough (e.g. metadata without a metadata schema) or not rich enough (e.g. stub or basic MARC records). Conversely, some data, such as that contained in our digitized and born-digital collections, had both richness and structure that would be flattened and lost when mapped to the schemas available in proprietary systems. Many discovery systems only handle MARC and Dublin Core records, mapping other metadata schemas to Dublin Core when necessary, which wouldn't allow us to take advantage of our full MODS records. I developed a new discovery strategy report (Popowich 2013) over the spring and summer of 2013, which included a roadmap for incorporating all our data silos in a single index and search interface, and would in the end provide a single search interface for our students, faculty, researchers, and staff. The report was approved in September 2013 by the head of the systems unit, the co-chairs of the library's web architecture team, the associate university librarians on the Senior Leadership Team, as well as the Chief Librarian. I therefore had a clear mandate to move forward with implementing the recommendations of the report.

In preparing the discovery report, I was strongly guided by Lorcan Dempsey's *Educause Review* article entitled "Thirteen Ways of Looking at Libraries, Discovery, and the Catalog: Scale, Workflow,

Attention." (Dempsey 2013) This article provides a framework for thinking about discovery systems and services that takes into account recent changes in technology and research behaviour.

Underpinning much of Dempsey's discussion is what he calls the "network level" or "network scale." On one hand, the network can be thought of as a higher-level system of a particular kind of institution (the network of libraries and record stores, for example, as opposed to any particular library or record store), but on the other hand the network is also the layer of services and artifacts that provide context to individual institutions. It is in this layer, Dempsey argues, that people work: library users today operate at the level of the network instead of at the level of their local, individual, library. In practice, the network is the world wide web: a layer above individual institutions which both links libraries together, but also links them to other kinds of organizations and services. Dempsey writes that, in recent years,

> access and discovery have [...] scaled to the level of the network: they are web scale. If I want to know if a particular book exists I may look in Google Book Search or in Amazon, or in a social reading site, in a library aggregation like WorldCat, and so on. My options have multiplied and the breadth of interest in the catalog is diminished: it provides access only to a part of what I am potentially interested in.[2]

This concept -- that discovery occurs at the network level instead of the institutional level -- has profound consequences not only for discovery, but also for online library services in general (link resolving and proxying, for example). We are starting to see some of these consequences playing out in library technology, but the idea raises many interesting questions about the library as web-presence or service provider. Does a single unified web-presence still make sense for our users? How does network-level usage affect how we employ analytics and statistics, especially social-analytics? What are the privacy implications of tracking usage at the network level? How do we compete with other network-level services while still maintaining a core focus on the needs of our constituents? Do we even need to

[2] Dempsey, para. 11. [Change these footnotes to refer to the print version].

"compete," or is there another model which might fit the library's mission and strategy more closely? Dempsey does not provide concrete answers to these questions, but does illustrate ways of thinking about the problem, some of which are perennial, and some of which are radically new. Dempsey outlines a few ways in which library discovery can adapt to the requirements of the network scale:

- provide simple search interfaces which lead to rich result sets
- integrate disparate library services into a single, network-level system
- harness alternative metadata sources and methodologies (e.g. crowdsourcing)
- be present in the user's workflow, rather than requiring users to come to us

In providing simple search interfaces that lead to rich results, a discovery system would conform to users' expectations of the network (in user experience, this is called the "principle of least astonishment" or "rule of least surprise"[3]). This in turn implies conformity of interface and functionality, which is enabled by library services functioning as a single entity at the network level. (This is not to say that library discovery should be monolithic -- providing single-purposes services can be an important aspect of discovery -- but that data and service silos should be broken down and combined into a single technology, with a single interface, to reduce duplication and be more intuitive to users). The idea of making our information available at the network level which underpins these suggestions signals a major shift in the thinking around library discovery: rather than providing access to owned or licensed material for "our" users, we are providing our information to the open web for "any" users. Dempsey calls this inside-out discovery (as opposed to more traditional outside-in discovery), and we will return to this idea later.

Much of Dempsey's article is concerned with how these suggestions might look in practice. I took them as starting points for an investigation into discovery options, and made a final recommendation for a discovery system that would allow us to begin to focus on the network level. In addition to the Dempsey article, the 2013 discovery report adopted some of the principles of a 2009 study by the

[3] https://en.wikipedia.org/wiki/Principle_of_least_astonishment

University of Minnesota as criteria for discovery system evaluation. For example, the Minnesota study identifies five trends, drawn from user studies and statistical usage reports of existing discovery systems (Web Services Steering Committee 2009):

- users are discovering relevant resources outside traditional library systems
- users expect discovery and delivery to coincide
- usage of mobile devices is expanding
- discovery increasingly happens through recommendation
- users are searching for more than just books and journals

In the discovery report, I tried to think about library discovery through the lens of Dempsey's network-level requirements, the trends identified by the University of Minnesota report, and the technical capacity and skills available at UAL. The following discovery systems were evaluated: Ebsco Discovery Service (EDS), Ex Libris Primo, Serials Solutions Summon, OCLC WorldCat Local, and the Blacklight open-source system.[4] EDS, Primo, Summon, and WorldCat were chosen because they were and continue to be used extensively by academic libraries (as opposed to BiblioCommons or Aquabrowser, for example) and they are not tied to a particular ILS vendor (such as Innovative Interfaces' Encore).[5] One of the major differences between academic discovery systems (e.g EDS) and those used in public libraries (e.g BiblioCommons), is that the former focus on resource discovery, and always include a knowledge base of journal articles or other bibliographic material. Public library discovery systems tend to focus on "social discovery," allowing users to see other users' comments, what others have read, etc. Blacklight is primarily used by academic libraries, but does not include a knowledge base. The need to integrate a knowledge base is one of the issues we had to address. The final factors for evaluation were:

[4] Blacklight (http://projectblacklight.org) was initially developed at University of Virginia in 2009. More information can be found in Sadler 2009.
[5] These discovery systems were also chosen for evaluation in the *Library Technology Reports* special issue on web-scale discovery (Vaughn 2011)

- Open-source or proprietary
- Local or cloud/hosted
- Maintainability (infrastructure)
- Customizability (user-interface)
- Customizability (indexing)
- Supported metadata schemas
- Active development/support
- Modern underlying technologies
- In-house skills/knowledge
- Cost

These factors had no intrinsic value, but in aggregate were weighed against in-house skill and capacity, budget constraints, and risk (e.g. with an open-source project under active development, we were likely to be able to count on community support).I included Blacklight for a number of reasons. Besides being a proponent of open-source as opposed to vendor systems for library technology, I felt that under the current budgetary circumstances, we would be remiss not to evaluate an application that was free (in dollar terms) but which could capitalize on in-house skill and knowledge. In the end, the discovery report recommended continuing to use EDS while a Blacklight implementation was developed, and eventually to include EDS as one pane of a "bento box" interface design.[6] To accomplish the integration of EDS results, we are using a Blacklight plugin developed by Ebsco, which allows Blacklight to search EDS and present results using the EDS API.

Blacklight is an open-source discovery system that uses Apache Solr[7] to index records, and a Ruby on Rails[8] web application for the user interface. It is designed primarily for MARC bibliographic records, but can be extended to include other kinds of record and, because Solr is schema-agnostic, it can

[6] For bento-box designs, see Lown, et al, 2013.
[7] http://lucene.apache.org/solr/
[8] http://rubyonrails.org/

index records that use any metadata schema (albeit without retaining the structure of those records). While moving to open source software provides a distinct set of challenges, not all of which are technological, the adoption of a new (to us) web framework (Rails) was not a primary concern, as we already had some experience with Ruby, the language that the Rails framework uses, and we were looking at Rails as a possible web framework for other projects. Until this time, our programmers had written code in Java, Perl, PHP, and Cold Fusion, and it was deemed advantageous to begin to move towards streamlining our language and infrastructure stack.

The decision to move more fully in an open-source direction, and to unify our programming language and web-framework stack, was made as part of a formalization of our IT governance processes. Faced with a reduction in staffing and a need to modernize processes and workflows, UAL's systems department was at the time adopting the Information Technology Infrastructure Library (ITIL). ITIL is a management framework which seeks to define

> the organizational structure and skill requirements of an information technology organisation and a set of standard operational management procedures and practices to allow the organisation to manage an IT operation and associated infrastructure.[9]

Reworking our internal processes and procedures allowed us to think about modernizing our entire software development, configuration, and deployment workflow. Adherence to strict project management principles was also new to the department. A suite of ITIL, project management, and Agile-software-development methodologies were adopted for the Blacklight implementation project. Some of these methodologies necessitated the adoption of new tools and technologies (e.g. Ansible, an automated configuration tool, similar to Puppet and Chef), but also required changes in organizational culture, in the areas of team structure, work distribution, collaboration, documentation, and others. In the end, the development consisted of myself as technical lead and co-developer, one of our web application

[9] http://www.itilbrary.org

developers as the other co-developer, and a system administrator. The team looked at both Scrum and DevOps models of team formation and work[10], but as the team was so small, and none of us were assigned 100% to the discovery implementation, we decided to adopt as many Agile principles and methods as we could without going "full Scrum" or "full DevOps."

One of the most challenging aspects of organizational culture that we recognized fairly early on is the tension between the traditional, service, model of library work, and a more project-based model. Typically, library units provide services (cataloguing, reference, interlibrary loan), and library workers work on a queue, either of bibliographic material, or users at the reference desk, or interlibrary loan requests. Project work also happens, but the core work of the unit is its service function, and takes priority. This model breaks down when faced with work that is fundamentally project-based, and which requires that staff be assigned 100% to the project at hand, something the traditional model can't ensure. In short, having software development project teams composed of staff members who also provide services, or thinking of multiple software development projects as services that can and should be offered simultaneously, has a negative effect on the ability of a software development team to focus and concentrate on the project at hand. The Blacklight team were supporting and developing other software projects and IT services at the same time as the implementation project. This was a challenge that was raised within the systems department, but was never fully addressed. As a result, work on the Blacklight implementation tended to move forward in fits and starts, and it was difficult to get uninterrupted staff time for working on the project, even though discovery was identified as a strategic priority for UAL.

There were, of course, technical challenges, in addition to organizational ones. We planned to adopt many new tools (for example, GitHub,[11] Ansible,[12] Vagrant,[13] and Jenkins[14]) in addition to Ruby on

---

[10] "Agile" is a broad term for ways of organizing and working on software projects, primarily associated with the open-source world. Scrum is a flavour of Agile and DevOps is another way of thinking about the division of labour within software projects. For Agile: http://agilemanifesto.org/; for Scrum: https://www.scrum.org; for DevOps: http://devops.com/.

[11] http://github.com

[12] http://www.ansible.com

[13] http://www.vagrantup.com

[14] http://jenkins-ci.org

Rails and some useful libraries (e.g. Opinionated Metadata[15] and Solrizer[16] for mapping bibliographic metadata in XML to Solr indexes). There were metadata questions to be answered, and metadata remediation to be undertaken. Metadata in various formats had to be extracted from native systems and mapped to particular index fields, and metadata without an explicit format had to be analyzed and then mapped.

Perhaps the most challenging, but rewarding, aspect of the project, however, was the adoption of a new focus on student requirements and student experience. In 2013, UAL espoused a set of strategic priorities that emphasized student experience (along with research data management and preservation), and allowed us to focus on student feedback and student needs or preferences in designing the system. The new discovery system would specifically not be a staff tool, and would only secondarily attempt to meet the needs of researchers and faculty (other discovery strategies, such as better exposure of library material on the open web, were considered as ways to improve the researcher and faculty experience).

The explicit focus on student requirements allowed us to justify coming down on the side of the student when conflicting requirements arose, say, between student need and staff need. We were able not only to avoid "feature creep" (e.g. staff-only features were not considered hard requirements), but we were able to defer some design decisions until we had done sufficient usability testing. This marked a major change in the thinking around software design and user experience at UAL, and was only possible because of a change in strategic focus and the use of open-source software. In a proprietary, vendor-driven software ecosystem, features are already built into a licensed system; in an agile, iterative, open-source world, features can be added and modified when enough information is gathered. Nothing is set in stone.

Working on implementing a discovery system from the ground up not only required dealing with technical challenges (e.g. overcoming data siloing, meeting user experience needs), but also brought us face to face with the larger questions of discovery and discoverability. Is the library still the best starting

[15] https://github.com/projecthydra/om
[16] https://github.com/projecthydra/solrizer

point for research? What is the role of Google/Google Scholar in the research workflow? Are there ways that we can better integrate our services into users' workflow, which includes Google, but is also broader than Google? Some of these questions were raised by Dempsey's "Thirteen Ways…" but also in discussion with librarians at other institutions and other service areas. Data supported some propositions, contradicted others, and was insufficient to draw conclusions for the rest. Clearly, in addition to technical challenges and responses to broader questions, analytics needed to play a major role in our new system. The combination of extensive analytics (the UAL systems department is a heavy user of Google Analytics) and in-depth usability testing should give us not only a sold discovery system in the short term, but also point the way for future development and strategies.

Autumn 2013 to Summer 204 were taken up with project planning and infrastructure work (server reorganization, hardware purchases, planning, and provisioning development environments using Ansible and Vagrant). The core development team consisted of only three staff members, none of whom had the discovery project as their sole focus, even for defined periods of time. The core team drew upon knowledge and work being done in other areas, primarily the digital projects department, who were at that point planning to implement Hydra as a digital asset management system.[17] We were able to share some infrastructure and continuous integration work (e.g. Ansible, Vagrant, Jenkins CI) as well as to share knowledge around Ruby and Rails best practices.

Between Summer 2014 and Spring 2015, active development of the Blacklight systems was underway. We wanted to start gathering user feedback early, but we also decided to present users with a working version of the software rather than gather feedback in a vacuum. This runs counter to some requirement-gathering methodologies, but the fact that we knew we were developing a discovery system allowed us to come up with a beta version before moving to user assessment. Prior to the launch of the beta-site, we conducted a spot check on the design and basic functionality, from which we got some good initial pass/fail feedback. The next step was to speak to the newly-instituted Student Library Advisory Council, both to unveil the new website and discovery system and to gather some feedback, but also to

[17] Hydra uses Blacklight as its index and search interface: http://projecthydra.org

get their help in getting together student constituencies for full-scale usability testing and focus groups after the beta launch.

To sum up, the 2013-2015 discovery implementation project built on changes within the organizational culture of UAL, both in terms of strategic priorities and technical responses to financial challenges. A clear mandate for a student-centered system, the adoption of a full open-source stack, the preparedness on the part of IT librarians and staff to adopt new technologies, process, and procedures, all contributed to the implementation of a discovery system that would satisfy fiscal, experiential, and information-management requirements in a large and complex organization. The beta version of the new discovery system was launched in May 2015, and the initial feedback was very positive. Structured usability testing and assessment with undergraduate students, graduate students, and faculty will take place over the summer.

As part of the initial launch, we have tried to communicate the flexibility of the software, in terms of both design and functionality, as emphatically as possible. This is done so that students, faculty, and staff do not feel constrained by the initial state of the application. A suite of acceptance tests, covering workflow and relevance ranking, have been developed so that we can keep track of decisions, and have a way of regression-testing the application through further iterations. Transparency was also a priority for us; the application code is freely available on GitHub, and our backlog of outstanding development issues is also hosted there.[18]

The process of choosing and implementing an open-source discovery system has certainly been challenging, but it has always been rewarding. Along with our Hydra implementation, Blacklight puts the library in a good position for open and sustainable development of digital projects that serve the needs of our core constituents. The challenges have been cultural as well as technical; IT, public service, metadata, and cataloguing staff have had to think differently about how we do the work we do. In an organization as large as UAL, such changes can be difficult, and the learning and acclimatization will take time. But the benefits in terms of increased stability and flexibility, and higher levels of trust and effectiveness can be

[18] http://github.com/ualbertalib/discovery

huge. Rewards come in the form of risk- and skill-sharing, as we participate in the wider world of open-source development and implementation. Letting go of the idea that our workflows, staff, or students are somehow unique, allows us to build on standard methods and process that have been developed and tested within various communities, and which are themselves flexible, sustainable, and open to change. Longer-term rewards are expected in the ability to add collections to the discovery interface, to build scoped-interfaces for particular groups or projects, and to fluidly modify the indexing and the interface as new requirements come to light. By implementing two systems from the same ecosystem (Blacklight and Hydra) we are able to deploy time and expertise more efficiently which, in an era of budget uncertainty and belt-tightening, is an important consideration.

References

- Lorcan Dempsey, "Thirteen Ways of Looking at Libraries, Discovery, and the Catalog: Scale, Workflow, Attention," *Educause Review*, December 10, 2012. http://www.educause.edu/ero/article/thirteen-ways-looking-libraries-discovery-and-catalog-scale-workflow-attention.
- Sam Popowich, Discovery Report, 2013.http://hdl.handle.net/10402/era.41274.
- University of Minnesota Web Services Steering Committee, Discoverability Phase 1 Final Report, 2009. http://purl.umn.edu/48258.
- Elizabeth (Bess) Sadler, "Project Blacklight: a next generation library catalog at a first generation university," *Library Hi Tech,* 27(1): pp.57 - 67.
- Jason Vaughn, *Library Technology Reports: Web Scale Discovery Services*, 47(1) (2011).
- Cory Lown, Tito Sierra, and Josh Boyer, "How Users Search the Library from a Single Search Box", *College and Research Libraries* 74(3) (2013): 227-241. http://crl.acrl.org/content/74/3/227.full.pdf+html.