Hello everyone, and welcome. I’m excited to be talking with you today as part of this online conference on innovating technical services workflows. My name is Rebecca French, and I’m the Metadata Analyst Librarian at James Madison University in Virginia.

In this presentation, Small Steps Make Giant Leaps, I’m going to be sharing a number of steps that we took (or are currently taking) to improve and automate a particular workflow at our library, namely the process for making our Special Collections finding aids available through a number of discovery platforms.
I know that those listening today probably represent a variety of types of technical services work, so I’m first going to make sure we have a shared understanding of some of the terms and concepts I’ll be using throughout this presentation. I’ll also provide some background and context about our organizational structure and responsibilities, and the systems we use. Then I’ll walk through each of the steps seen here on the slide and explain what we did and how that changed our workflow. I’ll also share the benefits we’ve observed, both from the workflow improvements as well as our gradual step-by-step approach. There should be time left at the end for questions, so feel free to type those into the chat box as you think of them and I’ll address them at the end.
I’ll start with a few definitions. As someone with a background in cataloging but not archives, some of these concepts were new to me when I first started working on this project.

Within Special Collections, when we talk about a collection we’re referring to “a group of materials with some unifying characteristic.” Often that unifying characteristic is that they belonged to a particular person or organization.

A finding aid is a guide to the contents of such a collection. Some of the key pieces of information included in a finding aid are historical or biographical information, a description of the collection’s arrangement (meaning how it’s organized), and a container list, which is an inventory of what’s in the collection. The machine-readable format used for finding aids is called Encoded Archival Description, or EAD, which is an XML standard.
To illustrate, here’s a section of a finding aid showing the description of one series and part of the container list for the series. The container list is an inventory and description of the materials that are in various folders. On the left side of the screen is the EAD XML document, and on the right is the same part of the finding aid as we display it on our website. [pause]
As I mentioned, at JMU we have a number of different discovery platforms where we make our finding aids accessible.
The first is on our website, where we have the complete finding aids for researchers to reference. The finding aid shown here is for the Wampler Business Records collection.
We also create a bibliographic record for each of our collections, and those are available through OCLC/WorldCat and in our local catalog, which is currently the Sierra ILS from Innovative.
Finally, we also contribute the full finding aids to Virginia Heritage, which is a statewide database that contains finding aids from many Virginia libraries.
To set the scene for our workflow improvement project, here’s an overview of who is involved in this work and how we divide up the responsibilities for different tasks. The workflow I’m discussing today involves two departments in JMU Libraries, Special Collections and Metadata Strategies.

In Special Collections, most of the arrangement and description of manuscript collections is done by two staff members. After a collection has been described, they put the finding aid on our Libraries website and submit it to Virginia Heritage, the state-wide finding aid database.

In Metadata Strategies, our cataloger of Special Collections materials creates bibliographic records for manuscript collections, based off of the finding aids created in Special Collections. She also assigns Library of Congress Subject Headings to these materials and adds the records to OCLC and to our local catalog. We also have a metadata librarian (that’s me). I’m not involved in creating the finding aids or other records. My role here was to figure out how we could improve this workflow, and I developed scripts and other tools that we used to automate parts of the process.

One part of the organization you don’t see listed here is our Technology department. We were able to do this work using the programming expertise we had within Metadata Strategies, and only consulted with our Technology folks once just to make sure that what we were developing met security requirements. In a different library, this same project might have needed to involve Technology more directly.
Here's a diagram showing our original workflow before we started making changes. The darker colored boxes represent systems or discovery platforms, for example ArchivesSpace. The lighter blue/green boxes are files, like the EAD XML file.

Our process started off on the left-hand side when a collection was described in ArchivesSpace. The Archivist then exported the EAD from ArchivesSpace. She copied data from that EAD into Adobe Contribute, which is HTML editing software, in order to publish the finding aid on the JMU Libraries website.

At this point, work moved to Metadata Strategies. The cataloger copied information from the web finding aid into a bibliographic record in OCLC and assigned Library of Congress Subject Headings. She added the completed record to the OCLC database and then imported that record into our local Sierra catalog.

Finally, the Archivist copied the subject headings from Sierra back into ArchivesSpace and into Contribute to update the website. With subjects added, she then exported an updated EAD from ArchivesSpace, made a number of additional modifications to the EAD to prepare it for Virginia Heritage, and uploaded it to that platform.
As I've described this workflow, you might have noticed that there's quite a bit of "copy and paste" happening. On this slide, I've put a star by the steps that involved a significant amount of manual work like hand-encoding a finding aid or copying and pasting data from one place to another. Modifying an EAD for Virginia Heritage was one such step, as were copying data into the website and catalog records. After subject headings were assigned, they also had to be manually copied back into the other platforms.
These points were problematic for a few reasons. First, it was incredibly time consuming to make all the modifications needed for preparing an EAD for Virginia Heritage – anywhere from half an hour to a couple of days worth of work, depending on the length of the finding aid. Copying and pasting data from one system or record to another is also not the most efficient, and ...
... doing this work by hand also creates opportunities for errors.
It made it difficult to make updates because changes would have to be made in multiple places in order to keep all the different platforms in sync with each other.
It’s also extremely boring, and my colleagues could instead be using their time to do work that they are uniquely skilled at, rather than copying and pasting.
So our first step in improving this workflow was to tackle the areas that were the biggest pain points. We hoped this would give us the greatest return on the time we would be investing in setting up a more automated solution.

For this step, we used XSLT, which is a language for transforming XML documents into other XML documents. I wrote XSLT transformation scripts that took the EAD XML file created from ArchivesSpace and turned them into the files we needed for the other systems. One XSLT made most of the modifications that were required for Virginia Heritage. Another XSLT generated an HTML file for the website. And a third XSLT created a MARCXML record that could then be further edited by our cataloger.

Special Collections staff used the Oxygen XML Editor program to run these XSLT scripts.
Here’s what our workflow looked like with the addition of the XSLTs. Special Collections staff would describe the collection in ArchivesSpace as before and export the EAD. Then using Oxygen, they would run the XSLT transformation scripts to produce HTML and MARCXML files. Our cataloger would make some additional changes to the MARC record, including adding subject headings and authorized access points, before uploading the record to OCLC and our ILS. Our archivist would then add the subject headings back into ArchivesSpace and use the XSLT scripts to generate an EAD for Virginia Heritage and an updated HTML file for the website.

So not only did we save a lot of time by eliminating much of the tedious copying and pasting, we’ve also made it easier to get the subject headings everywhere they need to go. We only needed to add the subjects back into ArchivesSpace because we would use the XSLT scripts to update the files instead of manually adding the subjects to those other places.

We approached this first step in improving our workflow with the mindset that it was very much an experiment, ...
... and I’m glad to say that it was a very successful one! Our archivist Tiffany said it was “a DRASTIC improvement over [her] previous workflow.” Based on that initial success, I started thinking about how we could further improve the process.
And that led to step 2, which was the creation of a custom Python app called Spaceport. The app uses the ArchivesSpace API to get the EAD from ArchivesSpace, which replaces the need to manually export the EAD from the ArchivesSpace staff interface. Then it runs the XSLTs to generate the desired files, replacing the need to run these individually in Oxygen. There were a few changes we needed to make for the Virginia Heritage EADs that weren’t possible to do using XSLTs, and I was able to use this app to also make those changes with Python, so we no longer needed to do any work by hand to produce the Virginia Heritage files.

I created a simple graphical user interface, which you can see on the slide here, to allow the user to easily select the files they want to create and to run the app. It is packaged into an executable, meaning it’s easily run from a shortcut, no command line required. We host the app on a shared network drive, which allows me to easily make updates and have them automatically pushed out to all users without any action needed on their part to receive the updates.

The code for Spaceport is freely available on my GitHub profile.
You can see from this diagram that the Spaceport app takes care of the file-handling parts of the workflow. It gets the EAD files from ArchivesSpace and runs the XSLTs that transform the EAD into the different files needed for our other platforms. Our archivist runs Spaceport after describing a new collection and then uploads the HTML files generated by the app to the website. The MARC file is sent to Metadata Strategies, where our cataloger adds subject headings and other authorized access points before adding the record to OCLC and our ILS. The archivist inserts the subject headings into ArchivesSpace and runs Spaceport again to generate an updated HTML file and the EAD file for Virginia Heritage. The Spaceport app can handle any number of collections, so we’re able to process a single collection or a large group at one time.
Our third step in improving this workflow involved making additional modifications to Spaceport. This step was separated from the initial development of the app because we wanted to get the basic functionality in place and tested first before spending additional time on these other improvements. As you can see from the image on the slide, one of the changes was adding status reporting to the interface to show progress as the app runs. Other changes were made to include error messages in this same interface. Users can copy this text if they encounter an error and send it to me, which makes it easier for me to troubleshoot. None of these changes altered the workflow; they simply improved the stability and functionality of the app.

This is our current production workflow. We’ve been using Spaceport for about a year and a half, with this updated version of the app in use for about eight months.
Before I move on to talk about our current development work and future plans, I want to share the benefits that we’ve experienced so far. Because we’re now able to easily make changes in ArchivesSpace and export them to other destinations, we truly have one system of record, rather than having finding aids in four different places that are all out of sync with each other where none is truly the “master.” Automating parts of this workflow has improved the accuracy of our metadata by decreasing the chances of typos and other mistakes.

It has also greatly sped up preparing files for our various discovery platforms. It now only takes a few seconds per collection to generate files, which is a 99% decrease compared to the time it used to take to hand-encode an EAD for Virginia Heritage. We’re able to generate web pages for all our collections in about 20 minutes with Spaceport. Because the app has greatly reduced the amount of time that needs to be spent on tedious formatting, Special Collections has been able to dedicate more time to standardizing description and making updates to legacy finding aids.

Finally, having this workflow with Spaceport in place has helped us complete a number of other projects that we might not have attempted or that would have been much more difficult without it.
The first one of these projects was the migration of our Libraries website content management system to WordPress. Because we could easily generate web pages for all of our finding aids using Spaceport, this prevented us from having to recreate each finding aid page by hand in the new CMS.
More recently, we decided to update our finding aid web pages to include much more information from the EAD and improve the look of the pages. This involved updating the XSLT transformation script that produced the HTML pages. The new XSLT is more flexible to accommodate a variety of collection hierarchical structures, which will mean less of a need for updates to the script in the future. Once we had the new XSLT, we could quickly update all the web pages with our improved workflow.
We also recently implemented Aeon, a request system for Special Collections materials. The system uses OpenURLs to auto-populate online request forms, which are linked from each finding aid web page. Again, we were able to use Spaceport to easily add the request links and generate all the updated pages.
Addressing this long-standing pain point has improved morale and generated lots of enthusiasm. Our archivist Tiffany says she “can’t even imagine [her] life without Spaceport.” Working together on these projects has also strengthened relationships between our two departments.
We have another improvement to this workflow that is currently under development. Remember that after our cataloger assigned subject headings, they needed to be copied back into the other platforms? We are automating the process of keeping our subject headings in sync between systems. This is building off of work done by Ruth Tillman, which she shared in a presentation at the 2019 ArchivesSpace Online Forum.

Using the APIs for our ILS and ArchivesSpace, a Python script will find manuscript collection bib records that have been updated in the catalog, parse the LC subject headings in those records, and then add any new subject headings to ArchivesSpace and link them to the corresponding resource record. Unlike the other workflow automation steps presented so far, this process would be entirely automated. We would set the script to run daily or weekly and it wouldn’t need any human intervention other than monitoring the results. Automating the addition of subjects to ArchivesSpace will make it easier to have more complete subject records, with correctly categorized subject terms within the larger subject string. We could also potentially look up URIs through the LC Linked Data Service and include those in our subject records.
Here’s what this would look like as part of our workflow. We would use the Spaceport app as before to create the web page and then upload it to the Libraries website. We’d also generate the catalog record and the cataloger would enhance it and add it to OCLC and our ILS. The subject sync script would run on a regular schedule, find updated records in the ILS, and add those subjects to the corresponding records in ArchivesSpace. Our archivist would then generate the EAD for Virginia Heritage and a new HTML file with the subjects to update the website.

At this point, we would have automated all of the places in the original workflow that required a significant amount of manual work: creating or modifying files for Virginia Heritage, the website, and OCLC, and updating all these platforms after subject headings have been assigned.
A future step we could consider to further streamline this workflow would be automating the process of uploading records to the various discovery platforms. Because the process of uploading records doesn’t involve a significant amount of work, in either time or difficulty, compared to the other parts of the workflow, this is the lowest priority for making improvements. But if we wanted to, we could experiment with using Python scripting and APIs to automate uploads to OCLC, our ILS, or the website. This might look like a script that detects new records in ArchivesSpace, or new files generated with Spaceport, and processes and uploads them.
Looking back on the work we’ve done so far, there have been a number of benefits to taking this incremental approach. As I’ve tried to make clear, this has not been a project that we planned out in great detail from the very beginning. Rather, each step has organically led to the next one. This gave us opportunities to pursue new ideas that wouldn’t have been apparent at the start, as well as to gradually build skills to work toward our evolving vision. By starting with the one change that would give the biggest bang for the buck first, we were able to generate excitement for the work we were doing and thus get more buy-in from stakeholders. This proved to be beneficial as we worked closely together throughout the project, meeting frequently to check in and provide feedback. It’s also been easier to gradually adjust to smaller changes in the workflow, compared to overhauling everything all at once, and it’s been easier to test these changes in small chunks. Finally, a step-by-step approach has enabled us to fit this work in around other priorities and projects instead of putting it off until we had time to do everything all at once.

Much of what we did in terms of working in small chunks and collaborating closely and frequently with stakeholders has a lot in common with agile project management methodologies, such as Scrum, so that’s something you could look into if you’re interested in a more formal, structured process with many of the same characteristics as what we did.
In closing, I’d like to share one more quote, this time from our Head of Special Collections. Kate said, “Archival description is fluid, and as archivists, we should be open to revisions and corrections to our description. Because it was so difficult and unwieldy to make changes across platforms, much of our description has languished for years because of the difficulty in applying changes. Spaceport has changed all of that. We are now much more able to uphold our ethical obligations to the historical record.”

I like this quote because it speaks to the impact these types of projects can have beyond workflow efficiencies. It’s not just about automating more processes, or getting more done faster. We’re improving our workflows to enable our colleagues to perform their jobs better and give them the time to do work that is meaningful for them and for the communities we serve.
I’d like to take a moment to thank all the creators of the images used in these slides, as well as my JMU Libraries colleagues who provided feedback on a draft of this presentation.
And now I’m happy to take questions. Thank you!