# Mining the Process: Alt-Text Services in Small Shops

*(Partial* ***ROUGH DRAFT*** *of presentation – items to be further explained are in [brackets])*

**By KimBoo York, Assistive Technology Coordinator  
Student Disability Resource Center, Florida State University  
Updated May, 2014**

## Agenda:

1. Introduction
2. The EPIC Story of the FSU Alt-Text Team
3. Get it Out of Your Head
   1. Process Mining
   2. Workflow Mapping
4. Baby Steps Up the Mountain
5. Tools to Use
6. Go Big or Go Home (creating a SharePoint or similar online portal)

## Introduction

FSU is not technically a “small shop” when it comes to disability services. We have a director, an assistant director, an associate director, a testing center supervisor, an ASL interpreter, a full time program assistant and me, the assistive technology coordinator. I have a staff of two part time people who work almost exclusively on alt-text services. We are a “big shop” in terms of personnel and space, but given the size of our student population, we still get swamped sometimes.

I’ll talk more about the story behind the extraordinary alt-text team in a moment, but I want first to talk about why someone like me is talking to you about small-shop problems and solutions.

I’ve been where you are. When I was hired it was part time to work on embossing a specific book (more about that later) and my supervisor, Patti Lester, was swamped by alt-text requests. She had help and she already had a good team, but service requests had exploded over the past couple of years and the production processes she had in place at the time couldn’t – and spectacularly didn’t – scale.

As a person with project management experience, I was asked to step in and start problem solving. That’s what I do, and I enjoy it. What we’re doing here today is a very shortened version of what I did then in order to streamline our production processes. At the time we were a five person team – Patti and four part-timers – along with a rotating “staff” of three to five student volunteers. These days, we’re half that size and doing just as much. What I’m here to do is to show you how to tame the monster that is alt-text processing and production.

## The EPIC Story of the FSU Alt-Text Team

It’s not a secret that FSU faced a lawsuit not long ago regarding alt-text services. I won’t go into details on that, but it is the reason I was hired back in 2011. I came on part time to specifically recreate calculus graphs as tactile images. I did 1200 of them. By hand, using Illustrator. Then we had to emboss the book, going from LATex to Nemeth to Braille. I can say with complete confidence that not one of us knew what we were doing, although we learned pretty fast.

My point is that we had to do this while ALSO fulfilling other alt-text requests. I’m not proud to say that some texts did not make it out to students until late into that fall semester. A system that normally worked fairly well completely crashed and burned under the pressure of a single exceptional request.

Think about what you do for a moment: what happens if you, or your assistant (if you have one) is out for a week? People move on to other jobs, have kids, get sick. Or you could just get hit, as we did, with a single alt-text request that simply stops all the gears. What do you do? How do you provide services for your students in a timely manner when all heck has hit the deck?

Fortunately, we had an awesome team, made up of Patti, excellent student workers, two librarians-in-training, and an experienced alt-text provider. Patti knew it was time to move forward, and create a better system. I won’t sugar coat it: from top to bottom, it took a lot of extra time, talent and effort. We had to call in outside help at times, especially when I started to put together our Sharepoint portal. You might have to do the same. But even if all you have is yourself, you CAN improve alt-text production and fulfillment to the point where it runs smoothly and requires less work. We didn’t solve our problems because of the number of people on staff; we solved it by improving our workflow. Think about how much less stress (and work) you would have in your life if you could cut the amount of time you spend on alt-text services by half. That’s what we did. Here’s how.

## Get it Out of Your Head

You’re a small shop, you know everything. You know where everything is, you know how everything is done, you know everything there is to know. I’m not challenging that. But as long as all those processes and procedures are locked up in your head, you can’t improve them. You can’t edit them or compress them or quantify them. Knowing everything is convenient right up to the point it’s not, and then you have a crisis on your hands.

### Process Mining

First, write it down. I know, it’s a pain. You’re busy! Who has time for that? You do. You have time for that, because if you don’t, you will eventually get beat down by the demands you’re making on yourself. I don’t mean time card yourself to the second, I mean write down what steps you take. It sounds dumb to write “check email for new requests” but it’s a start. WRITE IT DOWN.

Here’s what we found out when we started writing down what we were doing: a lot of time was spent going back and forth between tasks. A lot of time was spent looking up emails and checking paperwork. A lot of time was sunk into doing things twice. Even more time was sunk into tracking down work we had already done. You’ll not only get a clear picture of what you are doing, but where you’re getting bogged down.

### Workflow Mapping

Now that you’ve written it out, organize it in the way you think it should work. Boil down the steps you go through and either create a flowchart or a mind map or go old school and put each step on an index card and lay them out. Look at what we started out with. [image] It’s kind of horrifying, honestly, but this formed the backbone of site design for Sharepoint when we got there. Something like this can take you a long way.

## Baby Steps Up the Mountain

So let’s look at our example here. [image] It seems simple, we all know how it is supposed to go:

1. Get request from student
2. Get receipt from student
3. Request permissions and file from publisher
4. Send file to student

How does that work for you? Good? I didn’t think so. It’s a cronky, wonky system that never goes easily. Files come in and need to be chopped, cropped, converted to Word or Braille or Daisy. Sometimes files don’t come in at all and you’re scanning a whole book. Sometimes the publisher went under and you don’t know who holds the rights. Sometimes the publisher is in Italy and couldn’t care less to deal with you.

If things worked the way they were supposed to, I’d probably be out of a job. As it is, what we learned fast is that we needed a way to track the lifespan of a request from start to finish. And by track, I don’t mean “do a search in your Outlook folders.” You know who you are. [in depth discussion of paper forms , including value of knowing time spent on tasks]

We also learned that we needed better organization for the books we had already processed. At the time that number was…I don’t know. We had no idea how many texts we already had on hand. Not a clue. We had them on CDs and DVDs and on flash drives and in shared folders and personal desktops. [in depth discussion of alt-text library set up/maintenance]

What’s left? A procedure manual. This should be a combination of “standard operating procedures” and a “house style manual.” You will not believe the amount of time you will save if volunteers aren’t doing things wrong out of the gate. You won’t believe how many fewer panic stricken emails you will get on your vacation if other staff know they can just look at your procedure manual to figure out how to, say, request a book from a publisher. [example: FSU Alt-text manual, praise Joan a lot]

So you’ve got your elements in place: forms for tracking requests lifespans and task tracking (and volunteer hours if needed); a useful alt-text library; and a manual. Yes, this took time. Yes, you can now turn off email notifications during the weekend.

## Tools to Use, How to Save Money

This is quick overview of what we do to make things go fast. Here’s the deal: you get what you pay for. Is a Tiger Printer expensive? Yes. Is a PIAF printer expensive? Yes. [math book: costs+work hours was higher than outsourcing.] When you know you are spending X number of hours and X amount of money to scan and process a book, getting your bean counter to buy you a better scanner just got easier. Showing that you spend 40% of your time on alt-text requests goes a long way towards justifying both equipment and personnel.

1. Free/cheap labor (volunteers, work study)
2. Sheet fed and flat bed scanners
3. Adobe Acrobat Pro
4. Abbyy FineReader
5. View Plus Max embosser
6. PIAF thermal printer

## Go Big or Go Home

Honestly if you are doing a large number of requests, you need to automate. [Sharepoint story, quick overview of elements needed in place to develop a system]. AIM currently has an alt-text component, so if your department is looking at a data management tool that is worth considering. Clockwork is in the beta stages of their alt-text module as well. Outside of that you can use a free CMS such as Drupal or Joomla or Sharepoint, although be warned you need to have someone well versed in CMS development to make that happen. Having mapped out your processes, workflows and documentation will help ease the process. If your organization doesn’t have support systems in place for you to develop your own alt-text system, consider trying a secure a grant – and knowing your numbers, knowing the stats of what is going into your alt-text services will help you do that. It all comes back around to resources, time, and energy: YOURS.

## Tell Me Your Woes

[open floor for questions/discussion]

Contact Information:  
:::KimBoo York, MLIS

Assistive Technology Coordinator

Student Disability Resource Center / Department of Dean of Students

Florida State University

108 Student Services Building

874 Traditions Way

Tallahassee, FL 32306-4167

850-644-5532

[kyork@fsu.edu](mailto:kyork@fsu.edu)