

Level 3: Advanced

All Librarians

Module 14

Automating repetitive tasks

An introduction to library workflow automation for nonprogrammers. You'll learn to identify what's worth automating, build your first real automations using visual tools, and see concrete time savings in your actual workflow.

By Yulia Brusova

15 min read · Reviewed October 2026

AI for Academic Libraries · ai-for-academic-libraries.vercel.app

© 2026 · Licensed under CC BY-NC-SA 4.0

WHAT YOU'LL BE ABLE TO DO

- 1 Identify three to five tasks in your current workflow that are worth automating
- 2 Build a basic automation connecting two library tools using Zapier or Make
- 3 Apply AI to a batch processing task that previously required manual effort
- 4 Calculate the time ROI of an automation you've built
- 5 Understand when automation is appropriate and when human judgment is required

The most time-consuming library tasks tend to be recurring ones. Weekly usage statistics are pulled and pasted into a spreadsheet. Subject guide footers are updated by hand each semester with new library hours. Patron questions are copied from a chat system into a tracking document week after week. None of these tasks requires a librarian's professional judgment; they require a librarian's time. Automation returns that time to higher-value work. The tools that make automation accessible to library professionals without programming backgrounds have also matured considerably in recent years.

How to decide what is worth automating

Not every repetitive task is worth automating, and automating indiscriminately creates its own problems. A disciplined decision framework, applied before any tool is opened, prevents wasted setup time and produces automations that stay reliable. The framework has four components, and each must be evaluated honestly before proceeding.

Frequency is the first consideration. A task that recurs daily or several times a week justifies the time needed to build and test an automation; a monthly task rarely does. The calculation is simple but often overlooked. Library professionals sometimes spend two hours automating a task they perform four times a year, which produces no meaningful time savings over any reasonable period.

Consistency is the second consideration, and it is closely tied to the question of judgment. Automation handles consistent, predictable processes well. If a task follows the same numbered steps every time, with no judgment calls between them, it is an excellent candidate. It translates directly into a workflow tool's trigger-action structure without the practitioner having to account for exceptions. Tasks that require evaluating context at each step, deciding what to do based on information that changes from one instance to the next,

are not suitable for full automation. They may benefit from AI assistance at specific points, but a human must remain accountable for each decision.

Volume is the third consideration. Processing a batch of two hundred catalog records is worth automating; processing five is not. The efficiency gains of automation depend heavily on scale, and a one-time task, however tedious, rarely justifies the setup investment.

Error cost is the fourth and most consequential consideration. To evaluate it accurately, ask what happens when the automation makes a mistake, because any automated system running over time will make mistakes. For example, an automation that updates an internal tracking spreadsheet has low error cost. A human will review the output and can correct errors before they have further consequences. An automation that sends patron-facing communications has much higher error cost, because errors reach patrons before anyone can review them. Patron-facing automations therefore need human review steps built deliberately into the workflow design.

A complementary frame from the AI Fluency course identifies three prerequisites for any delegation decision, automation included (Dakan & Feller, Anthropic Academy, 2025). The four-factor framework above assumes them but does not name them. The first is problem awareness. Before opening Zapier, Make, or any other tool, the practitioner must be able to state precisely what success looks like. A librarian who cannot describe what the finished automation should produce is not ready to build it; the tool will fill that gap with assumptions that may not match the actual need. The second is platform awareness: knowing which automation tool suits which type of workflow. Zapier offers branching through Filters and Paths on its paid plans, while Make builds conditional logic into its visual design from the start, a different architecture suited to a different class of problem. Choosing the wrong tool often produces an automation that technically runs but needs more maintenance than it saves. The third is task delegation: deciding which steps of a workflow the automation should handle and which must stay with the practitioner. That division is rarely obvious in advance. It requires knowing the workflow well enough to see exactly where judgment enters the process and where it does not.

Zapier for library workflows: starting simply

Zapier is the most accessible automation tool for library professionals without programming backgrounds. It connects applications with a trigger-action structure: when a specified event occurs in one application, a specified action runs in another. For example, when a patron submits a research consultation request through a library booking form, Zapier can log the submission to a shared tracking spreadsheet and notify the reference team. This replaces

manual copying and pasting and ensures requests are not missed when staff are out of the office.

To build a first automation, the practitioner selects a trigger, an event in one application that starts the workflow, and an action, what happens in a second application as a result. Zapier's interface guides the practitioner through this selection with no code required. Zapier also maintains a library of prebuilt integrations with most applications common in library settings, including Google Workspace, Microsoft 365, [LibCal](#), and standard email providers. Many library workflows can therefore be automated without any custom configuration.

Several workflow patterns suit Zapier's structure particularly well in library contexts. The form-to-spreadsheet-to-notification pattern handles research consultation intake automatically. A patron submits a LibCal or Google Forms request, and Zapier logs it to a shared spreadsheet and sends a [Slack](#) message or email to the relevant staff member. This pattern typically takes fifteen minutes to configure and needs no ongoing maintenance once running. The email-to-task pattern serves a different need. When email containing specified keywords arrives in a specific library inbox, Zapier creates a matching task in a project management tool such as [Asana](#), [Trello](#), or [Notion](#). A new-resource-to-announcement pattern lets a library queue social media posts or newsletter items automatically when new resources are added to a tracking list, provided the data source supports a trigger event.

The free Zapier tier, which allows unlimited two-step Zaps (one trigger and one action) and 100 tasks per month, is sufficient for most light library use. Start with one automation, test it for several weeks, and evaluate its reliability before building more. This incremental approach builds confidence with the tool before you build anything with higher error costs.

Make for more complex library workflows

Make, formerly known as Integromat, handles more complex automation scenarios than Zapier. Its visual interface represents a workflow as a flowchart. That makes it easier to see what happens at each step and to find where errors occur when a workflow misbehaves. To use Make effectively, map out the full workflow before opening the tool. The visual design process is much faster when the steps are already set down on paper or in a simple diagram.

Make is the right choice when a workflow has more than two steps, requires conditional logic, or processes batches of data rather than responding to individual events. Consider an automated notification workflow for new database trials. It begins when a new row is added to a Google Sheet tracking active trials, then passes through a filter that checks whether the trial status column reads "Active." An AI integration drafts an announcement in the library's

established voice from the database name and description columns. The draft goes into a Google Doc queue for staff review before any patron-facing communication is sent. This workflow turns a thirty-minute task into a five-minute review without removing human judgment from the final approval step.

Conditional logic is Make's most important practical advantage over Zapier, particularly for libraries on free plans, since Zapier reserves Filters and Paths for its paid tiers. For example, a single automation might route a patron request by the patron's institutional role. Faculty requests go to one staff member, student requests to another, and requests from specific departments are flagged for a subject librarian. This routing requires branching, which Make supports through its filter and router modules. Make also handles data transformation. The practitioner can extract specific fields from a spreadsheet, reformat dates, or combine text before passing data to the next step.

Make's free tier allows 1,000 credits per month (the unit that replaced operations in 2025) and two active scenarios, with scheduled runs no more often than every fifteen minutes. Most library automations built in Make for internal workflows will not approach this limit under ordinary conditions. That makes the free tier a viable starting point before any paid plan is considered.

Batch processing with AI: high-value library applications

Some of the highest-value library automation needs neither Zapier nor Make. It needs a systematic approach to prompting AI with batches of content that would otherwise be processed one item at a time. For example, a library that receives ten similar patron reference questions each week can build a response template for each question type. AI then customizes the template for each patron's wording and context. Twenty minutes of response drafting becomes approximately two minutes of prompt submission and review.

Batch metadata work is another application with a significant time return. To normalize inconsistent catalog records, export a CSV of the affected records and paste batches of thirty to fifty records into [Claude](#) or [ChatGPT](#) with a clear instruction: for each record in this list, suggest a normalized publisher name, and flag any records that appear to be duplicates. The structured output can be reviewed in a fraction of the time needed to check each record manually. The AI output is not applied directly to the catalog. A staff member reviews and corrects it before any changes are made, which keeps a human accountable for the final state of the records.

Document summarization is a third high-value batch application. Monthly vendor reports, annual assessment survey results, and faculty feedback documents all follow consistent internal structures, which suits them to AI summarization. A prompt requesting a two-paragraph executive summary and a list of action items from a ten-page vendor report typically produces accurate, usable output in under thirty seconds. This is especially valuable for library professionals who receive reports from multiple vendors or departments and must synthesize them before a budget or planning meeting.

The time arithmetic of batch processing is concrete and worth calculating explicitly. For example, a monthly chat reference statistics summary that takes forty-five minutes by hand typically takes under ten minutes with an AI-assisted workflow. The statistics go to Claude with a request for a summary paragraph and a comparison to the previous month, and the output is reviewed and lightly edited before use. A saving of approximately thirty-five minutes a month compounds across all recurring documentation tasks into a meaningful return over a full academic year.

When automation is inappropriate and what it cannot replace

Automation is powerful when tasks are consistent, low-stakes, and well-defined. It creates serious problems when applied to tasks that do not meet those conditions. The professional judgment needed to identify which tasks suit automation is itself a core competency, one that no automation tool can supply or replace.

Certain categories of library work must not be automated. Communications that require empathy or individual judgment, such as responding to a patron in distress or handling a sensitive reference question, need human presence at every step. These situations call for the practitioner's professional assessment of what the patron needs. That assessment cannot be reduced to a trigger-action rule without meaningful risk of harm. Similarly, decisions with institutional consequences, including collection weeding and access policy changes, require human accountability and must not be delegated to an automated system running without oversight.

Patron-facing communications warrant particular caution, however well an automation has performed in testing. An automation that emails patrons directly, without a human review step, removes the library's ability to catch errors before they reach the patron. Errors range from minor (a date formatted incorrectly) to significant (a message sent to the wrong patron or containing inaccurate policy information). To use automation responsibly in patron-facing contexts, build a human review step into the workflow design as a structural requirement. Do not add it as an afterthought when problems arise.

The most important limitation of automation is not technical but conceptual. Automation handles the standard case; the practitioner handles everything else. For example, an intake form automation can route a consultation request to the right staff member under normal circumstances. When the patron's need does not fit a standard category, or the request is unusual, urgent, or needs an exception to procedure, a human must assess and respond. These exceptions are not failures of the automation. They are exactly the situations that distinguish professional library work from rule-following. Professional judgment is not what remains after automation takes the routine tasks. It is the core of library practice, and automation is one of several tools library professionals use to protect time for that work.

Dakan and Feller describe the professional responsibility at the point of sharing AI-assisted work as deployment diligence: the practitioner, not the tool, is accountable for what is produced and distributed (Dakan & Feller, Anthropic Academy, 2025). That accountability applies without qualification to automated workflows. An automation may generate a patron-facing email, a statistical summary sent to administrators, or a catalog record deposited without manual review. In each case, the expected standard of accuracy is the same as for work done entirely by hand. The automation does not excuse an error; it is the means the practitioner chose to produce the work. This framing makes a stronger case for human review steps than any list of risks. Review steps exist not because automations are unreliable, but because the practitioner remains professionally responsible for every output that leaves the library under its name.

PRACTITIONER NOTE

The automation that changes library workflows most reliably is rarely the most sophisticated one. A simple Zap connecting a LibCal appointment form to a shared tracking spreadsheet, with a Slack notification to the team, takes approximately five minutes to build. It prevents missed appointments when staff members are out of the office. The decision about which tasks to automate matters far more than the technical complexity of the resulting automations.

KEY TAKEAWAYS

- A task worth automating is frequent, consistent in its steps, high in volume, and low in error cost if the automation produces a mistake.
- Zapier connects two applications through a trigger-action structure and is the most accessible starting point for library professionals without programming backgrounds.

- Make handles workflows requiring conditional logic, multiple sequential steps, and batch data processing, and is the appropriate tool when a Zapier Zap is not sufficient.
- Batch AI processing, submitting groups of similar documents, records, or patron questions to an AI tool with a structured prompt, produces significant time savings without any automation platform.
- Patron-facing communications and decisions with meaningful institutional consequences must not be fully automated; human review steps are non-negotiable at those points.
- Professional judgment is not what remains after automation handles routine tasks; it is the core of library practice, and automation exists to protect the time available for that work.

References

APA 7TH EDITION

1. Dakan, R., & Feller, J. (2025). *AI fluency: Framework & foundations* [Online course]. Anthropic Academy. <https://anthropic.skilljar.com/ai-fluency-framework-foundations>
2. Make. (n.d.). *Pricing*. Retrieved October 11, 2026. <https://www.make.com/en/pricing>
Free plan: 1,000 credits a month, two active scenarios, 15-minute minimum interval; routers and filters included.
3. Zapier. (n.d.). *Plans and pricing*. Retrieved October 11, 2026. <https://zapier.com/pricing>
Free plan: 100 tasks a month and two-step Zaps; Filters and Paths start on paid plans.

ACRL AI Competencies covered

Use & Application

Sub-competencies: 4.1, 4.4 · ACRL AI Competencies (2025)