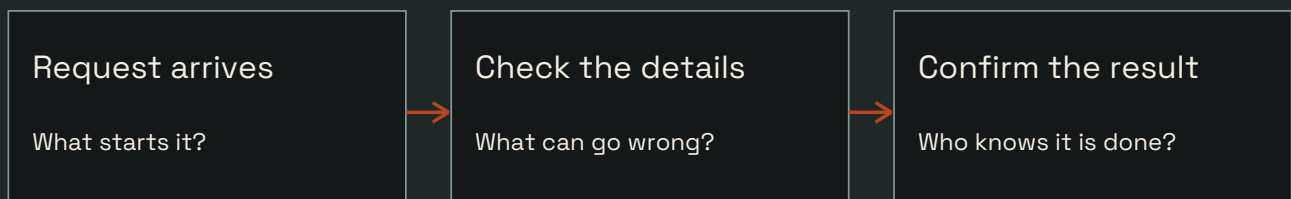


Map it before you automate it.

Find the repeatable work. Make the hand-over easier.

For the job at work, the community project or the routine at home that you keep meaning to sort out.

TRY ONE REAL EXAMPLE



Then add the wait, the missing detail and the person responsible.

I would start here before writing code or handing a job to someone new.
You need to see the work to make it easier.

Callum Janes

Founder and organiser, AutomateNI

Start with the job you have.

A first map needs an honest example, not specialist software.

1. Put a boundary around it

Name the result and the event that starts the work. “Request received to collection confirmed” is easier to map than “improve the whole organisation”.

2. Follow one recent run

Ask the person doing the job to walk through it. Write an action in each box: read the request, check the date, record the booking. Keep the current process separate from your proposed fix.

3. Make decisions visible

Use a question where the route splits. Label the answers and show where both paths go. Include the missing-information route, the refusal and any loop back for another check.

4. Add owners and hand-offs

Name who does each action and what they need before starting. Include the copy-and-paste between tools; that movement is part of the job too.

5. Measure work and waiting separately

Note active effort, time waiting and repeat visits to a step. Mark guesses as estimates. Follow several ordinary examples and at least one awkward one before trusting the numbers.

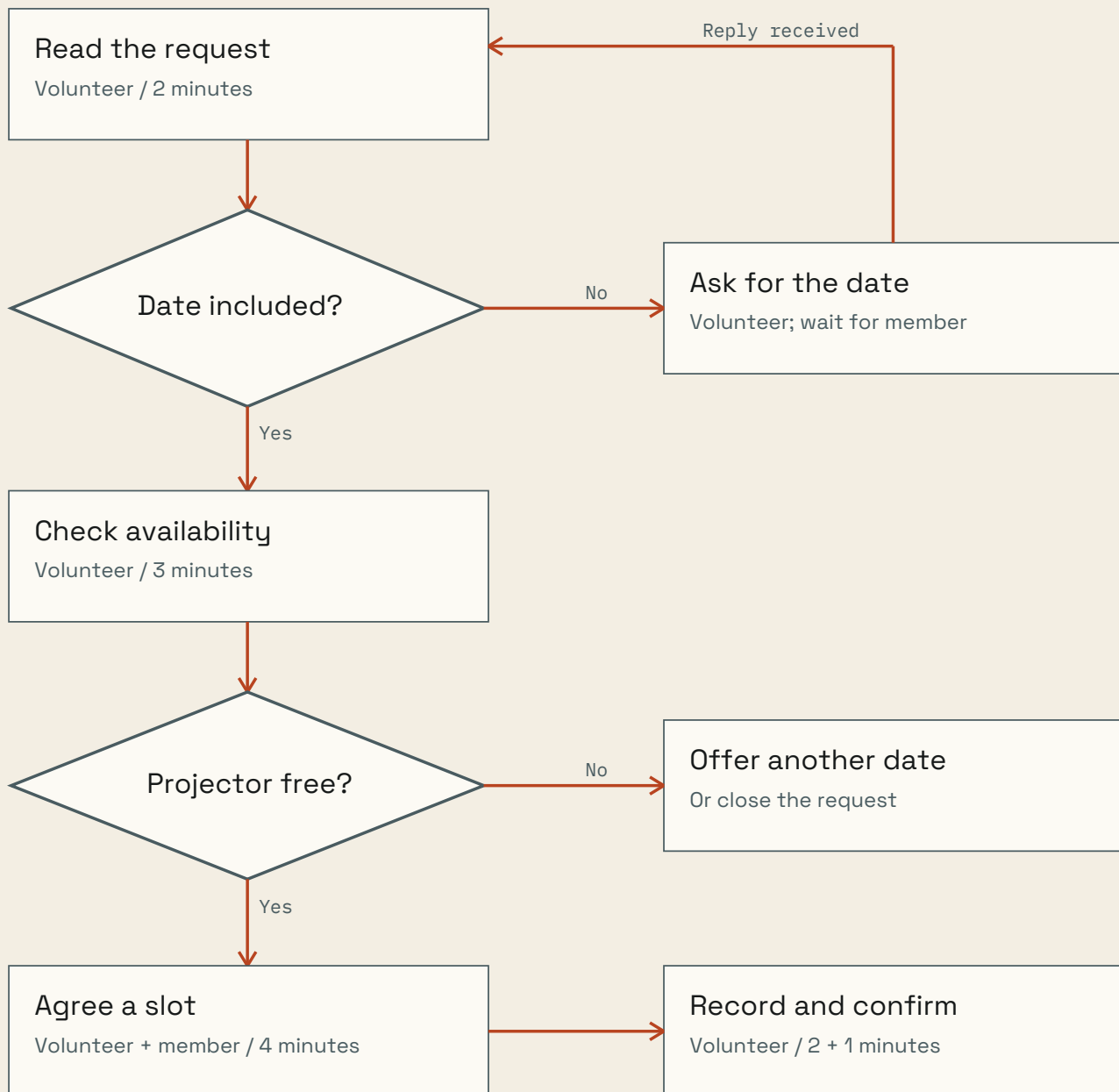
6. Walk it together

Ask someone else to follow the map using a real example. Correct the gaps while they are obvious. Add an owner, version and review date so the next change has somewhere to go.

Useful symbols: rectangle = action; diamond = decision; arrow = direction. Plain language matters more than perfect notation.

Can we borrow the projector?

Invented equipment-loan example. These are teaching details, not an AutomateNI service or measured result.



A changed date goes back through the availability check. An incomplete request has its own waiting time; do not add that branch to every complete request.

Twelve minutes of work. Eighty-seven minutes elapsed.

One complete request, with sequential steps. All figures below are invented.

Step	Work / min	Wait / min
Before reading	0	45
Read request	2	0
Check availability	3	0
Agree collection slot	4	30
Record loan	2	0
Send confirmation	1	0
Total for this route	12	75

Read the numbers before choosing a tool.

A faster confirmation removes at most one minute of active work in this example. It does not remove the time waiting to notice the request or waiting for a reply.

Investigate the cause of the wait.

Would a clearer request, an agreed checking time or a named owner help? Ask before adding a notification. More alerts can create another queue to manage.

Working time + waiting time = elapsed time only for this sequential example. Keep parallel work, alternate branches and repeated checks separate. Measure your own process before estimating savings.

Make one hand-off easier.

Your next improvement might be a better instruction, a smaller form or a script.

If you are delegating the work

Give the new person the map, a completed example and the instructions for the fiddly steps. Show what arrives, what “done” means and who can approve an exception. Let them try a normal case and an awkward one while you are available.

If you are automating a step

Choose a repeatable rule with readable input and a checkable result. Prepare a draft first. Keep judgement with the appropriate person, especially where a wrong result could affect safety, money or access.

When something fails

Show where the stopped request will be visible and who will pick it up. Keep the manual route available. A duplicate request, missing date or unavailable service should have an expected outcome before the workflow goes live.

The one change we will try

Name the step, the person affected and what should become easier.

How we will know it helped

Compare real runs. Include checking, exception handling and maintenance.

Put one process on paper.

Use this page to set the scope, then draw on the separate landscape canvas.

Process and boundary

Starts when... / Finished when...

People and the awkward route

Who owns it? Where does it wait, fail or return for another check?

What we will measure

Active work / waiting / repeated steps / result for the person using it

Owner: _____ Version / review date: _____

Continue on the AutomateNI website

The one-page mapping canvas

/downloads/workflow-mapping-canvas.pdf

The article and further reading

/resources/process-mapping-before-automation/

Save a plan in the Workflow Planner

/tools/workflow-planner/

Reference: ASQ, "Flowchart" (asq.org/quality-resources/flowchart). Common symbols and current-process review. Examples and worksheets: AutomateNI. Use the paths above on the community website.