A Practical Guide to Creative Automation

What to hand to machines in creative production, what to keep, and how to start without breaking anything

RELAY WORKS



Six checkpoints

Automate the plumbing, not the taste 4

What to automate, and what to keep human 7

Map the workflow before the tool 11

Pilot one automation in two weeks 15

Prompts and templates that hold the brand 18

Quality control and governance 21

A Practical Guide to Creative Automation

What to hand to machines in creative production, what to keep, and how to start without breaking anything.

Every creative team has a second job it never applied for. Between the ideas and the finished work sits a layer of moving, renaming, resizing, chasing, copying and reformatting that nobody trained for and nobody enjoys. It is where deadlines slip and where tired people make small, expensive mistakes. It is also, almost entirely, work that a machine can do.

This guide is for creative teams, content marketers and small agencies who want to automate that layer without automating the parts that make their work worth paying for. It is tool-neutral on purpose. We name categories of tools rather than products, because the products change every quarter and the method does not. Every example in these pages is illustrative: the studios are invented, and we quote no time savings, because yours will depend on your own work and we would rather you measured them.

Automate the plumbing, not the taste

Creative production is two kinds of work tangled together. One kind needs taste: choosing, judging, persuading. The other is plumbing. Automation belongs in the plumbing, and most teams start at the wrong end.

Take any week in a small studio apart and the tasks sort themselves into three piles. The sorting matters more than any tool you will pick later, because it decides what you are allowed to hand over.

Mechanical work.	The same input always gives the same correct output: resizing a banner, naming a file, exporting a PDF.
Judgement work.	The right answer depends on context, taste or stakes: choosing a concept, cutting a headline, deciding it is finished.
Relationship work.	The value lies in who does it: presenting to a client, delivering bad news, negotiating scope.

Automation succeeds reliably on the first pile. It can assist on the second, by drafting options, flagging problems and running checks, but it should never decide. It should be kept well away from the third, because a client who discovers they have been handled by a machine stops trusting the rest of your work too.

There is a simple test for any step you are tempted to automate. Ask: if this step went wrong silently, who would notice, and when? If the answer is "we would, within the hour", automate it freely. If the answer is "the client, next week", you may still automate it, but only with a human checkpoint in front of the client. If the answer is "nobody, ever", stop and ask why the step exists at all.

What has changed, and what has not

The tools have changed. A no-code automation tool such as Zapier or Make can connect a form, a project board, a file store and a chat channel without an engineer. Design tools have batch export and scripting. Generative models can draft copy, variations and summaries in seconds. Document tools can turn plain Markdown into a designed, branded PDF.

What has not changed is that a broken process, automated, becomes a broken process that runs faster and at night. If a brief arrives incomplete today and someone fixes it by phone, an automation will happily build a whole job folder around the incomplete brief and nobody will phone anyone.

> **IMPORTANT**
> Automation is not a strategy. It makes a decision you have already made happen without you. If the decision is not written down, there is nothing to automate yet.

The cost nobody budgets for

Every automation is a small piece of software, and software needs an owner. Connections break when an app renames a field. A colleague who built the workflow leaves, and nobody knows what it does. A trigger fires twice and sends a client two invoices. None of this is a reason to avoid automation. It is a reason to budget for ownership from the start: a named owner for each automation, a short written note of what it does, and a known way to switch it off. A team that cannot list its automations does not control them.

02

What to automate, and what to keep human

The best first candidates happen often, follow rules you can write down and fail cheaply. Six areas of creative production almost always qualify. Four kinds of work never do, not because machines are bad at them, but because their value is that a person did them.

Resizing and versioning

A campaign rarely ships as one asset. It ships as a master and a long tail of sizes, crops, languages and formats. Building the master is creative work. Producing the tail is not. Set up master templates with fixed safe areas and let your design tool's batch export, or a dedicated creative automation platform, generate the variants. The designer's job becomes reviewing the set, not producing it, and the review is faster because every variant was built from the same rules.

Briefs and intake

Most rework starts with a brief that was missing something. Replace the email brief with a form that will not submit until the essentials are there: objective, audience, the single message, mandatory elements, formats and sizes, deadline, approver. A no-code automation tool can then create the job card, the folder and the chat thread in one move. Nothing creative has been decided. The information simply arrives complete, every time, in the same place.

Asset naming and handoff

File names are a handoff protocol. Agree one pattern and let the system apply it: client code, campaign, format, size, version and date, in that order and always in lowercase, so a file reads like "nrt_spring-launch_story_1080x1920_v03_2026-03-14". Once names are predictable, routing becomes automatic: finals move to the client folder, superseded versions move to an archive, and nobody sends "final_FINAL_v2" again.

Approvals and routing

The decision to approve is judgement work. Everything around it is plumbing: sending the right version to the right person, reminding them, recording the answer and moving the file on. Automate the routing and the record, never the decision. A useful side effect is an approval log you can show a client when they ask who signed off what.

Reporting

Pulling numbers out of ad platforms and analytics into a weekly sheet is mechanical. Writing what the numbers mean is not. Automate the collection and the layout, then reserve a person's time for the three sentences at the top that say what changed, why it probably changed and what you will do about it. That paragraph is the part clients read.

Document production

Guides, proposals, reports and case studies are often written well and then lose a day to layout. Write them in plain Markdown, keep the brand in a locked theme, and let a Markdown to PDF publishing tool produce the designed document. The writer stays in the words; the brand stays consistent because nobody is re-creating it by hand each time.

What stays human

Some work should stay human not because machines are bad at it today, but because its value is that a person did it, stands behind it and can be asked why. Four kinds of work belong in that group.

Taste

A generative model can produce fifty headlines in the time it takes to read this sentence. Choosing the one that is right for this client, this moment and this audience is the job you are paid for. Treat generated options as raw material, never as output. The more options a machine can produce, the more valuable the person who can discard forty-nine of them becomes.

> Generate as many options as you like. Choose them yourself.

Final judgement

Someone must decide that a piece of work is finished and fit to publish, and that someone must be a person with a name. This is partly about quality and partly about accountability: when a claim turns out to be wrong, an image infringes someone's rights or a message lands badly, "the workflow published it" is not an answer any client will accept. Automate everything up to the final yes, and keep the yes human.

Client relationships

Presenting work, explaining a delay, pushing back on a bad idea, noticing that a client is unhappy before they say so: these are the moments that keep accounts. Automated status emails are fine; automated apologies are not. A useful rule is that any message where tone matters more than content is written by a person, even if a tool drafted it first.

The quiet fourth: deciding what not to make

The most valuable creative decision is often to make less: one strong asset instead of twelve weak sizes, one clear report instead of a weekly dashboard nobody reads. Automation makes volume cheap, and cheap volume tempts teams to produce more of everything. Keep a person responsible for asking whether each automated output should exist at all.

Do

- Automate the steps around a decision: preparing, routing, recording.
- Use generated drafts as starting material for a person to shape.
- Keep a named human on every piece of work that reaches a client.

Don’t

- Let an automation approve, publish or send on its own judgement.
- Send generated text to a client without a person reading it first.
- Automate a step just because it is possible.

Map the workflow before the tool

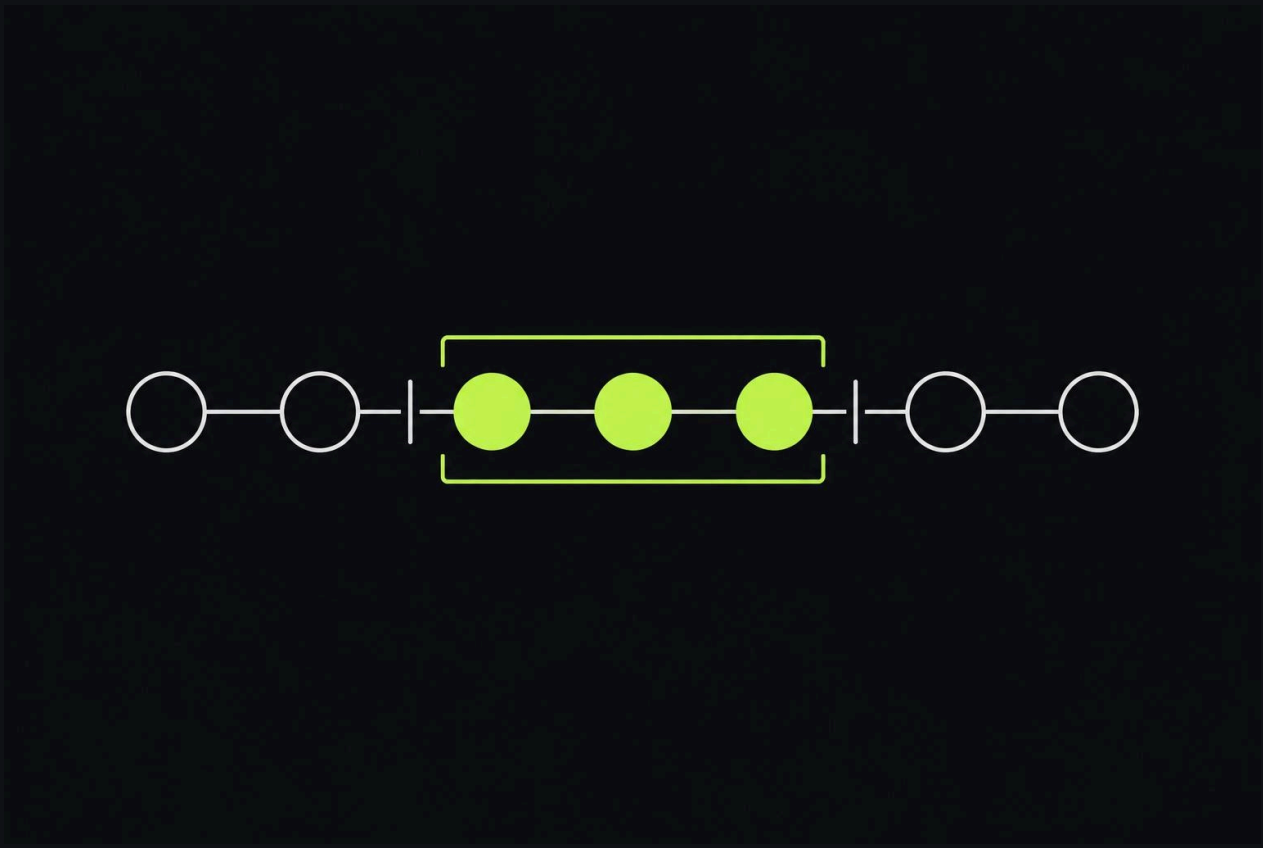
Most automation projects fail at the whiteboard, not in the software. If you cannot draw the process as it really runs today, workarounds included, you will automate the version people describe in meetings.

Mapping is a two hour job with the people who do the work, a wall and a stack of sticky notes. Do not invite only the managers. The person who renames the files knows things about the process that nobody else in the building does.

How to map one job

1	**Pick one real job.** Choose something your team delivered recently, not a hypothetical, so every step can be checked against what happened.
2	**Walk it backwards.** Start from the delivered file and ask "what happened just before this?" until you reach the first email.
3	**Write each step as a verb and an object.** "Export master banner", "rename files", "upload to client folder". Vague steps hide work.
4	**Mark every handoff.** Each time work changes hands or changes tools, draw a line. Handoffs are where things wait and get lost.
5	**Mark the waits and the rework.** Note where the job sat idle and where something was done twice. These are your real costs.
6	**Sort each step into a pile.** Mechanical, judgement or relationship. Only the mechanical steps are candidates.

When the wall is full, step back and look for clusters. Automation pays best where several mechanical steps sit in a row between two handoffs, because one automation can then remove a whole stretch of waiting rather than a single click. A lone mechanical step surrounded by judgement work is rarely worth automating on its own.



One job drawn as a chain of steps. The highlighted run of mechanical steps between two handoffs is where a single automation removes the most waiting.

> **TIP**
>
> Photograph the wall before you take the notes down. The map you made is the specification for whatever you build, and the first thing you will want when an automation misbehaves.

An illustrative map

Here is a shortened map of an invented studio's social campaign job, sorted with the three piles:

Step	Kind	Candidate for automation
Receive the brief by email	Mechanical	Yes: an intake form creates the job card and folder
Choose the key visual	Judgement	No: the art director decides
Produce every size from the master	Mechanical	Yes: batch export from master templates
Rename and upload the files	Mechanical	Yes: a naming rule and a folder sync
Check every version	Judgement	Assist: an automated checklist, then human sign-off
Present to the client	Relationship	No: a person presents and listens
Collect and record approval	Mechanical	Yes: a routed request with a reminder and a log

Read the right-hand column as a shortlist, not a plan. Three yeses in a row (the export, the naming and the upload) form the cluster this studio should look at first, because together they remove a handoff and a wait. The intake form is a strong second candidate. The checking step stays human, with a machine doing the tedious part of the inspection.

Pilot one automation in two weeks

A pilot is a small, reversible experiment with one owner, a written definition of success and a fixed date on which you decide. Two weeks is long enough to see real jobs pass through and short enough that nobody gets attached.

Choosing the pilot

Pick from your map one cluster that is frequent, low risk and visible. Frequent, so that two weeks contains enough real jobs to judge it. Low risk, so that a failure costs an hour, not a client. Visible, so that the team feels the difference and wants the next one. Asset naming and handoff is the classic first pilot for exactly these reasons. Approvals routing is a close second.

Before anyone opens an automation tool, write one paragraph that says what the automation will do, what it will not do, who owns it and what you will look at on the last day. If that paragraph takes more than ten minutes to agree, the pilot is too big.

The two weeks

Ten working days, one owner, one pilot:

- Day 1: Write the one paragraph definition, agree the success signals and the kill criteria, and name the owner.
- Day 2: Build the smallest version that works end to end, on test files only. Ugly is fine.
- Day 3: Run it on last month's real jobs, in a copy of the folders, and compare with what people did by hand.
- Day 4: Switch it on for live work, with the manual way still available beside it.
- Day 6: Hold a fifteen minute check-in. Collect every surprise, fix the two that matter, ignore the rest.
- Day 8: Write the one page note: what it does, where it lives, how to switch it off.
- Day 10: Decide. Keep it, change it or remove it, and write down why.

A worked example (illustrative)

Northgate Studio is an invented six person agency producing social and display campaigns for regional retailers. Its map showed that finished files were renamed by hand, uploaded to a shared drive and announced to the account lead in chat, three steps that each waited on a different person. Its pilot replaced them with one rule: any file saved to an "approved" folder is renamed to the agreed pattern, copied to the client's delivery folder and announced in the job's chat thread.

The team did not try to measure time saved. They tracked three observable things across the two weeks: how many deliveries needed a correction after sending, how often someone asked where the final version was, and whether anyone quietly went back to doing it by hand. The last signal turned out to be the most useful. A designer kept renaming files manually because the pattern did not cover animated formats, which no one had mentioned at the wall. The owner added one rule, and on day 10 the team kept the automation.

> **REMEMBER**
> The people who bypass an automation are giving you the best feedback you will get. Ask them why before you ask them to stop.

Whatever you decide on day 10, write it into the automation register: what the automation does, who owns it, where its note lives and how to switch it off. Then wait a fortnight before starting the next pilot. Running two at once makes it impossible to tell which one caused a surprise, and the second pilot always goes better when the team has lived with the first for a while. Your map already holds the shortlist; the order is yours to choose.

Prompts and templates that hold the brand

Generative tools are fluent in the average. Left alone, they write like everyone, which is what a brand exists to avoid. The fix is better inputs: a written voice, real examples, hard constraints and a fixed output format.

Treat a prompt the way you treat a brief. A designer handed a vague brief produces generic work and blames the brief; a model does the same without the complaint. Most disappointing output comes from a prompt that left out what a new freelancer would have needed to know.

The anatomy of a brand-safe prompt

Role and task.	One sentence that says who is writing and what for.
Context.	Audience, channel and the single objective, copied from the brief.
Voice rules.	Words you use and avoid, sentence length, how you address the reader.
Examples.	Two or three approved pieces in the same format, marked as examples.
Constraints.	Character limits, mandatory terms, and claims that must never be made.
Output format.	Exactly what to return, how many, in what order.

Here is a reusable template for social captions, with the variable parts in square brackets. Store it, name it and version it like any other asset:

> You are writing social captions for [brand], [one line description]. Audience: [who]. Objective: [one action]. Follow the voice rules below exactly. Never use these words: [list]. Match the tone of these approved captions: [examples]. Write five options, numbered, each under [n] characters, with no hashtags and no emoji. Do not invent product claims, prices, dates or statistics.

The last sentence earns its place in every template. Models fill gaps with plausible inventions, and an invented price or statistic in a published caption is a problem no amount of brand voice will fix. Telling the model what it may not invent, and then checking for it anyway, is cheaper than an apology.

Templates beyond prompts

The same principle applies to everything you produce repeatedly. A proposal, a monthly report or a case study should start from a template with fixed headings, fixed order and fixed design, so the variable part is only the content. Plain Markdown is a good container for this: it is easy for people and models to write, easy to compare between versions, and a publishing tool can apply the brand theme at the end, so nobody rebuilds the layout by hand.

> **NOTE**
> Keep one master per document type and delete the copies. A template that exists in four slightly different versions is not a template, it is four brands.

06

Quality control and governance

Automation moves errors faster and in bulk, and it blurs who made what. Quality control and governance are the counterweights: a few deliberate checkpoints, three short documents and one habit of writing things down.

Two kinds of check

Split the checking in two. Machines are good at checks with a right answer: is every required size present, does every file follow the naming pattern, is the logo the current version, is any banned word in the copy, does every link resolve. People are good at checks without one: does this feel like us, is the crop awkward, would this headline embarrass the client in a year.

A checklist for any automated output

1	**Completeness.** Every format, size and language the brief listed is present, and nothing extra slipped in.
2	**Correctness.** Names, prices, dates and legal lines match the source of truth, not a model's memory.
3	**Brand.** Colours, logo version, typefaces and voice rules match the current brand sheet.
4	**Rights.** Every image, font, clip and track has a known source and a licence that covers this use.
5	**Disclosure.** Generated content is labelled wherever your policy, the platform or the client requires it.
6	**Fit.** A person who knows the client has read it whole and would put their name to it.

The first four can be partly automated. The last two cannot, and should not be.

Sampling and escalation

For high-volume output, you cannot read everything, and pretending you do leads to skimming. Check every item for the first batches of a new automation, then move to a sample, and go back to checking everything whenever something upstream changes: a new template, a new model, a new client. Decide in advance what happens when a check fails. The simplest rule is also the best: stop the line, fix the cause, then rerun the batch. Patching one bad file and letting the rest through is how the same mistake ships a hundred times.

> **WARNING**
> Never let the same automation produce work and approve it. The checker must be independent of the maker, or the check is only a second copy of the same mistake.

Governance without the bureaucracy

Governance sounds like a big company word. For a small team it means three short documents and one habit: know what you may use, say what was generated where it matters, and keep client data where it belongs.

None of this is legal advice, and the rules are moving. Copyright treatment of generated work, platform labelling rules and data protection obligations differ by country and change often; regulation such as the EU AI Act, for example, includes transparency duties for some kinds of generated content. Ask a qualified adviser what applies to your work, then write the answer down in the plainest words you can.

Rights

Every asset in your work has a source and a licence, and automation makes it easier to lose track of both. Keep an asset log that records, for each image, font, clip and track, where it came from, under what terms and for which client. For generated material, record the tool and the account it was made with, and read that tool's terms for commercial use. Never prompt for the style of a named living artist, and never feed a client's confidential material into a tool that may learn from it.

Disclosure

Agree a disclosure policy with each client in writing before the work starts: which kinds of generated content they accept, where it must be labelled and who decides edge cases. Then follow the platforms' own rules on top. The commercial reason is as strong as the ethical one: a client who finds out afterwards feels deceived, even when the work was good.

Data

Decide which data may enter which tool, and write it on one page. Client confidential information, unreleased products and personal data belong only in tools with business terms and data controls you have checked. Keep an approved tools list, review it when a tool changes its terms, and make it easy for people to ask before they paste.

The register

Five questions, and where each answer should live:

Question	Where the answer lives	Owner
Which tools are approved for client work?	The approved tools list	Operations lead
Can this asset be used commercially?	The asset log, with source and licence	Producer on the job
Does this client accept generated content?	The contract and its disclosure note	Account lead
What data may enter which tool?	The one page data rules	Operations lead
Who can switch an automation off?	The automation register	The named owner

A register this short can live in a shared document and be reviewed once a quarter in half an hour. That is enough governance for most small teams, and far more than most have. It turns automation from a set of clever shortcuts that one person understands into infrastructure the whole team can trust, which is the only kind worth building.

Start with one workflow, not a strategy deck

1. Send us one workflow that eats your team's week
2. We map it with you in a single working session
3. You leave with one pilot, scoped and ready for two weeks

Book a workflow mapping session