Agent best practices

From the SourceLace manual: How to build agents that work first time, stay safe and cost what you expect. Each rule below is a mistake we want you to avoid, with what to do instead. If you are new to agents, start with Build your first agent, then come back here before you turn your first one on.

SourceLace version 0.1.0 · Made November 22, 2026
The latest version is at https://sourcelace.ai/docs

Agent best practices

How to build agents that work first time, stay safe and cost what you expect. Each rule below is a mistake we want you to avoid, with what to do instead. If you are new to agents, start with Build your first agent, then come back here before you turn your first one on.

The short version: **start small, try it safely, widen slowly, and watch the first week.**

Start with one small job

- **Do:** give an agent one clear job that you can check by eye, such as "list my open deals with no activity for three weeks and draft a nudge for each".
- **Avoid:** one agent that does five things. It is hard to write, hard to check and hard to fix when one part goes wrong. Build five small agents instead.
- **Why:** a small job is easy to dry-run and easy to trust. You can always add a second job later.

Write instructions that leave no room to guess

- **Do:** say what to read, which records count, what to produce, how many at most and what **not** to do. Use the names your systems show, such as stage names and field names.
- **Do:** tell the agent what to do when it is unsure, for example "if you are not sure, list the record in the summary instead of changing it".
- **Avoid:** open words like "fix", "clean up" or "update everything".
- **Why:** an agent follows your words literally. A vague instruction gets a vague, sometimes costly, result. See Write instructions that work for before-and-after examples.

Give it the fewest systems and actions

- **Do:** list only the systems the job needs and only the actions it needs. Start with a summary, a document or drafts.
- **Do:** add record changes last, and list only the fields it should change.
- **Avoid:** adding "just in case" systems or actions. An agent can only do what you list, so a longer list only adds ways to go wrong.
- **Why:** every extra system is more to read (and more credits) and every extra action is more that can be done by mistake.

Keep people in the loop for changes

- **Do:** choose **I approve each change** for record changes and review the **Approvals** inbox. An agent that proposes a change is much safer than one that makes it.
- **Do:** allow a change without approval only when it is small, easy to undo and you have watched the agent get it right many times. Your admin decides what is allowed.
- **Avoid:** turning approvals off to save a click.
- **Why:** a wrong change that nobody approved can spread through a system before anyone sees it.

Always dry-run first

- **Do:** press **Dry run** and read the run page: the summary, the cards under **Changes**, the **Steps** and the **Credits**. Compare the numbers with a report you trust.
- **Do:** dry-run again after every edit. Two or three tries is normal.
- **Avoid:** turning an agent on after one glance.
- **Why:** a dry run changes nothing and drafts nothing, so mistakes cost you nothing but a few credits. See Try it with a dry run.

Try it in shadow before it goes live

- **Do:** for any agent that will run on its own, let it try in shadow for a few days. It runs on its real schedule but changes nothing, and you review what it would have done. Press **Go live** only when the reviews look good.
- **Do:** if you edit a live agent to add a new kind of change, a new field or a new system, review a few runs of the new version first.
- **Why:** real data is always messier than your test. Shadow shows you the surprises before they matter. See Try it in shadow.

Control cost and risk

Credits pay for the AI reading and thinking in each run. These are general rules, not prices. Your plan page shows what you have.

- **Read less.** Ask for a date range, one team or one region instead of everything. A question that reads thousands of rows costs far more than one that reads a hundred, and it is slower.
- **Run less often.** An agent that runs every hour uses about twenty-four times what a daily one uses. Choose the slowest schedule that still meets the need. Event agents are usually cheaper than short schedules, because they run only when something happens.
- **Cap every run.** Keep **Most credits per run**, **Most steps** and **Most records changed** on the **Limits and alerts** tab as low as the job allows. A run that hits a limit stops and tells you, which is better than a surprise bill. Raise a limit only when you know why.
- **Set a daily maximum.** Use the most runs per day as a safety net, especially for event agents that a busy day could set off many times.
- **Do the sums before you turn it on.** The run page shows the credits one run used. Multiply by runs per day and by days to see a month. The checklist in Build your first agent has this step too.
- **Remember shared agents.** A team agent runs once for each person who subscribes, so ten subscribers cost about ten times one person.
- **Watch the first week.** Look at the **Usage** page and the agent's runs every day for the first week, then weekly. A sudden jump almost always means a filter that is too wide or a trigger that fires too often.

Handle records and email safely

- **Do:** remember that emails are always drafts in your own mailbox. SourceLace never sends them. Read each draft before you send it.
- **Do:** name who each draft is for ("the opportunity owner's email") so drafts do not go to the wrong people.
- **Do:** use your organization's data protection rules. Protected values reach the AI as placeholders, so the agent can still do its job.
- **Avoid:** pasting passwords, keys or other secrets into instructions.
- **Why:** instructions are visible to other people who can open the agent.

Share only what you have proven

- **Do:** share an agent as a template, or with a group, only after it has worked well for a few weeks. Write a clear name and description so others know what it does and what it needs.
- **Do:** pick the narrowest sharing level that works: just you, a group, then the organization.
- **Why:** people who use your agent depend on it. See Make templates for your organization.

Keep it healthy after launch

- **Check the results** in the project you chose, and the **Needs your attention** list, at least weekly.
- **Look at the runs** that failed, were refused or stopped at a limit. Each one tells you what to fix.
- **Pause** an agent the moment you doubt it. Pausing is instant and you can turn it back on.
- **Review it** when your systems change: a renamed stage or field is the most common reason an agent that worked yesterday starts returning nothing.
- **Retire** agents nobody uses. They still count toward your plan's number of agents that are on.

Checklist: the ten rules in one place

- [ ] One small job with clear instructions.
- [ ] Only the systems and actions it needs.
- [ ] Record changes wait for approval.
- [ ] Dry-run finished and the numbers match a report you trust.
- [ ] Tried in shadow for a few days (for agents that run on their own).
- [ ] Narrow reads (dates, teams, regions) and the slowest schedule that works.
- [ ] Limits set low, with a daily maximum.
- [ ] Monthly credits worked out and fit your budget.
- [ ] You will watch the first week.
- [ ] Shared only after it has proven itself.