

ORGCORD

The ORGCORD cheat sheet

Some books collect their rules in one place at the end. Rob Fitzpatrick's *The Mom Test* does this well, and it makes the whole book easy to put into practice. This is ORGCORD's version: the principles and recommendations from this writing, in one list. The number after each one is the chapter it comes from.

The problem

- Get organized, whatever your style. The right level of detail depends on the work and the person. (2)
- Keep what you're not working on outside your head, in a shared system. (2)
- Agree on the conclusion, not the thinking. Then stay in sync as the plan changes. (3)
- Split a large group into teams, and have the team leads keep in sync across teams. (4)
- Put the tool at the center, use meetings, chat and email around it, and get everyone into it. A system that is only partly shared carries the costs of both approaches. (5)
- Save meetings for what works better in real time: debating an open question, deciding on the spot and building relationships. (5)

The framework

- Check the four prerequisites: KNOW, WANT, CAN, TRUST. (6)
- Keep five things in sync: goal, ownership, priority, status, dependency. (6)
- Make every task trace up to the goal. (7)
- Give every piece of work one owner, with the context to do it and the authority to decide. (8)
- Watch how much one person owns: one reliable person who keeps taking on more becomes the bottleneck. In a small company, leadership acts as the team leads. (8)
- Rank your priorities. If everything is high, nothing is, so when something goes to the top, decide what moves down. (9)
- Work on fewer things, so each one gets more energy. (9)
- Give every task and project a clear finish line in plain language, and avoid vague status words like "almost done" and "working on it". (10)
- Keep dependencies in one central place in the system, and give every external participant an internal owner. (11)
- Use the tool two ways: doing the work in it, and stepping back to see timelines, milestones and how the work went. (12)

Principles

- Make updating the tool part of doing the work: mark a task in progress when you start and done when you finish. That is the status update. (13)
- Connect your AI assistant to the tool, so updating it takes one prompt, and review what it writes. (13)
- Run a short cleanup every week for tasks that are stale, unowned or out of date. (13)
- Use the fewest, clearest units that represent the work. Don't make four projects when one will do, or thirty tasks when five will. (14)
- For fast work, use one task that names the action, mark it in progress at the start, and leave a short note at the end on what got done and what changed. (14)

- Check your planning with two tests. Internal: are you working on what moves the needle most? External: can the people around you keep up with what you're working on? Frustration with upkeep is a sign of over-planning. (14)
- Avoid redundancy: put dates on milestones, and give a task its own date only when it has to happen on a specific day or before the rest of the milestone can move. (14)
- Set the edges and leave the middle free (flow in a box). (15)
- For a creative sprint, create one task, mark it in progress, comment once a day or at the end, and mark it done. For project-level tracking, make it a small project with one or two tasks. (15)
- Represent the work as it really is right now. An idea for another department starts as a task to explore it and come up with a plan. (15)
- Explain why the work is visible. (16)
- Open most work to everyone. Restrict finance, customer data and leadership's own planning to the people it concerns. (16)
- Never turn one number into a person's target, and never tell someone you're tracking the number of tasks they complete. (17)
- Use numbers to compare a team or person against its own past, plan capacity, find where work piles up and help people see their own progress. AI can summarize the patterns. (17)
- If it's not in the tool, it's not happening, for everyone, leaders included. (18)
- If someone hands you work outside the tool, get it into the tool before you act on it. (18)
- As a leader, read the initiatives, project timelines and project updates before meeting with the team. (18)
- If an assistant keeps the tool updated for you, keep that work small and recognized, still use the tool yourself, and show them the value their work creates. (18)

Running it

- Make initiatives outcomes leadership decides on. Keep them few; it's fine to start with one. (19)
- Find initiatives by starting from the goal, from the constraint, or from the projects you already have. (19)
- Test each candidate initiative. Would leadership decide on its own whether to continue it? Does it have one owner, a measure of success and a target? Is it too big for a single project? (19)
- Give every project a finish line. No bins: keep recurring work off projects, as standalone, labeled recurring tasks. (19)
- Keep projects lightweight: a clear name, a couple of milestones that are outcomes, a few tasks under each, and one lead. When a project closes, decide on everything still open in it. (19)
- Make each task completable by one person. Once it needs several people, it's several tasks or a project. (19)
- Use estimates for the size of the work (a large estimate is a cue to break it down) and labels for the type of work. (19)
- Keep simple task descriptions to a sentence or two. For important or cross-team work, cover why it matters, the context, what done looks like, timing and blockers. (19)
- Use a checklist for steps one person works through. Make a separate task when a piece needs its own owner, status or dependency. (19)
- Name every level with a verb: initiatives with the outcome and a target, projects with the thing they deliver, tasks with a specific action. Write names the way a person would say them out loud, without codenames or em dashes. (20)
- Start with as few teams as you need, each with one lead. (21)
- Process triage to zero every day. (22)

- When you process triage, give each item a title, a description, a priority and an owner, then send it to the current plan, the backlog, or close it. (22)
- Use triage for messy work, for requests and for recurring tasks. Send a request to a person's triage when you know who should do it, and to the team's triage when you don't. (22)
- Keep the backlog small. Important things resurface. (23)
- Treat moving a task from the backlog to Todo as a commitment. (23)
- Start with one-week cycles, and plan to your real capacity. (23)
- At the end of each cycle, look at what was planned against what got done, what was added mid-cycle and where work got stuck. (23)
- Plan each week at the start of the cycle, post updates during it, and hand off a report at the end. AI can draft the plan and the report. (23)
- Mark a task In Progress only while someone is working on it. When a task sits untouched for weeks, close it, reassign it or move it back to the backlog. My rule of thumb: stale for 30 days and not urgent, delete it. (23)
- Put decisions, questions and things that didn't work in a comment on the task. (23)
- Link a dependency when a task is actually blocked or about to be, using blocks, blocked by and related. (24)
- When you're stuck, don't go silent. Comment with what's blocking you, what you've tried and who or what you need, and tag the person who can help, or whoever owns the outside relationship. (24)
- Use meetings for decisions, and build them around the tool. A leadership meeting can walk through the initiatives and the projects under each one. (25)
- Post short project updates on a regular cadence: on track, at risk or off track, what changed, what's blocked, what decision is needed and what happens next. Initiative owners post updates too, and a project with no recent update is worth checking on. (25)
- Keep working documents (plans, agendas, decision notes) in the tool, linked to the work. Permanent documentation can live in a knowledge base. (25)

When it breaks

- Watch for theater, invisible work and overload. (26)
- Before asking someone else to decide, say what you would do. (27)
- Give each role clear autonomy: a plain understanding of which decisions are theirs to make. (27)
- Review everything AI writes before it goes into the tool. If it writes poorly, fix it or change how you use it. (28)
- Don't spam tasks, and don't let AI spam them into your team. Watch the defaults, like AI marking everything high priority. (28)

Knowing it works

- Measure system health, flow and outcomes. Only outcomes ultimately count. (29)
- For system health, track goal traceability, tasks with no owner, stale tasks, blocked tasks and how work spreads across priorities. (29)
- For flow, track lead time, where work clusters by status, planned versus unplanned work, and throughput for the same team over time. (29)
- For outcomes, track each initiative's result against its success criteria, and the trajectory over time. (29)
- Compare people against their own past, not against each other. (29)
- Ask team leads what the tool can't see. (29)
- To find the constraint, remember it's always against something, zoom out to the goal, and work backward to the driver that moves it most. (30)
- Label work that builds people, systems or decisions, and give it priority. (appendix on leverage)