

High Output Agent Management - V1

i. Preview/Intro

The way we work is changing faster than we realize.

We're learning to manage intelligence instead of just people. And the old playbooks don't quite fit.

But the principles that worked for managing people? They still apply. You just need to apply them differently.

Andy Grove wrote about this decades ago. He understood that a manager's job is leverage, not labor. That insight is exactly what you need now when agents become part of your team.

The core idea: build the system well, invest in clear instructions, and set each agent up to win.

ii. The System

An agent is made of five elements: role, instructions, context, memory, and tools.

The real asset isn't the model itself. It's the system you build on top of it.

Every model eventually gets replaced. Your system compounds.

What makes a system durable is consistency. Same role, same instructions, same context. Swap the model, the system works.

This is why you should think of agents as team members, not just tools. Then you can build teams of them, not just individual agents.

A team of agents compounds in ways a single agent never can.

iii. The Problem - Cost Equation

There's an equation that determines whether your agent work is efficient or wasteful. It's about time, tokens, and tool use.

Every agent task burns time, tokens, and tool use. The goal is understanding those costs and minimizing them while still getting the output right the first time.

One accurate attempt costs you far less than three mediocre ones.

That doesn't mean you never iterate. It means you iterate with direction, not randomly.

The danger is iteration without direction. You dig yourself into a hole you have to undig.

So before you brief an agent, understand there's a cost equation at play. Mastering this equation is how you work efficiently.

Efficient token usage is the lever. Clear instructions and tight context keep your agent focused, not wandering.

iv. The Solution - Planning & Context

The solution to the cost equation is simple. Invest time upfront in planning and context so the agent gets it right in the first execution.

This is where you and the agent work together. You don't write perfect instructions in isolation.

You brief the agent, it questions back, you refine, it surfaces edge cases, you adjust. That back and forth is how you build real instructions.

This is exactly like onboarding a team member. You spend time aligning on constraints, decisions, and the judgment calls they'll face. That alignment is what prevents disaster.

Perfect instructions don't exist. You will tinker over time. The goal is clarity now. Clear enough that the agent understands your intent and won't wander.

Once you ship the system, you learn from what actually happens. That feedback refines everything. The craft is continuous.

v. Creative Selection

This is where you show your taste. Your judgment. The vision only you have for what matters and what doesn't.

An agent without taste produces slop. Even the smartest model will churn out generic work if you don't steer it with creative judgment.

You know what's necessary and what isn't. You know when something has soul and when it's empty. That knowledge is your leverage.

This is Grove's leverage principle, but applied to taste instead of labor. You multiply your impact through judgment, not hours.

vi. Revisions + Learning

Your systems get smarter through feedback. You measure what works, what doesn't, and you iterate.

The magic is that you can document what worked and what didn't. Build that back into your context for next time. Your agent systems improve by learning from themselves.

Your agent's context becomes a living document. Every iteration, every insight, feeds back into it. The system learns.

Your instructions tighten with each cycle. Your context gets sharper. The system gets faster at finding the answer.

This compounding effect is what separates a good system from a great one. You're not starting from scratch every time.

vii. Conclusion + Actions

For decades, knowledge workers were valued for three things: remembering how things work at your company, knowing what to do, and using the right tools. Agents can do all three. Now your job is different. You architect the system. You bring taste. You make the opinionated calls on what matters.

That shift is the whole game. You become a manager, an architect, a creative director. All at once.

Your job is to architect the system. Feed it clarity. Let it learn. The compounding starts immediately.