

Hiring through The Goal.

What a 1984 factory novel knows about why your searches stall.

Most hiring advice is about people: how to interview better, how to write a job post, how to sell a candidate. This guide is about something less discussed and more decisive: the *system* your hiring runs through. And the sharpest book ever written about systems that produce things is not a hiring book at all. It is The Goal, Eliyahu Goldratt's 1984 novel about a plant manager named Alex Rogo who has three months to save a failing factory.

The plant Alex inherits looks busy. Every machine runs. Every worker has a queue of work. Efficiency reports are green. And nothing ships on time. If you have ever run a search where resumes kept arriving, interviews kept happening, everyone was working, and the role stayed open for four months anyway, you already know this factory. You work in it.

The Goal's ideas became the Theory of Constraints, and they translate to hiring almost line for line. This guide is the translation: where your pipeline's real bottleneck sits, why busy teams lose hires, and the five-step method for closing searches that used to stall.

What is your hiring process actually for?

Goldratt opens with a question that sounds too obvious to ask: what is the goal? The plant tracks dozens of measures. The book sweeps them all aside: the goal is to make money, and everything else is a means or a distraction. Ask the same of hiring. The goal is not to post jobs, collect applicants, screen resumes, or run interviews. The goal is a signed offer that becomes a productive hire. Now, and again the next time a seat opens.

Goldratt reduces the plant to three measures. Hiring reduces the same way:

Throughput

Signed, accepted offers. The only number that is the goal itself.

Inventory

Every candidate currently in process. The applicant pool, the shortlist, everyone "awaiting feedback" or "in approval." All of it is work you have paid for that has not yet produced a hire.

Operating expense

Everything you spend converting inventory into throughput: coordinator hours, manager attention, interviewer time, tooling.

Every hiring decision you make can be judged by one test: **does it increase signed offers while shrinking the pile of stuck candidates and the effort spent per hire?** A bigger applicant pool fails that test unless the pool converts. A faster screening tool fails that test unless screening was actually your slowest step. Which brings us to the central idea.

Your pipeline has a bottleneck, and it isn't sourcing

The Goal's core claim: every system has a constraint, and the constraint sets the output of the entire system. The plant can only ship as fast as its slowest critical machine, no matter how fast every other machine runs. An hour lost at the bottleneck is an hour lost for the whole plant. An hour saved anywhere else is a mirage.

Every hiring pipeline has a bottleneck too. Here is the uncomfortable part: for most small and mid-size teams, it is not where the effort goes. The effort goes to the front of the funnel: sourcing, posting, screening. But applicants are abundant and reading resumes scales. The constraint is almost always at the back, in the chain of human decisions: the manager who has not returned feedback, the comp approval sitting in an inbox, the offer letter that took a week to draft and another week to sign off. Those steps have the least capacity, the most dependencies, and the highest cost of delay.

Run Goldratt's test on your own last three searches. For each candidate who got deep into the process, add up the time they spent actually being *evaluated*: interviews, reviews, reference calls. Then add up the time they spent *waiting* for someone to decide something. The second number is usually five to twenty times the first. Your pipeline's speed is not set by how fast you work. It is set by how long work waits.

Candidates are inventory that spoils

Here the factory analogy breaks, and it breaks against you. In Alex's plant, a part sitting in front of a machine costs carrying cost and nothing more. It waits patiently. It will still be a part next month.

A candidate does not wait patiently. A candidate sitting in front of your bottleneck is interviewing somewhere else. Every day of queue time actively degrades the asset: interest cools, competing offers land, the story your team told about momentum stops being believable. When a candidate you spent six weeks and fifteen interviewer-hours on signs elsewhere during your approval process, your system did not lose a candidate. **It scrapped finished goods.** This is why constraint thinking matters more in hiring than in manufacturing, not less. Goldratt's plant could at least warehouse its mistakes. Yours walk.

Busy is not the same as hiring

The most quoted line in *The Goal*: "A plant in which everyone is working all the time is very inefficient." It sounds backwards until you see what full utilization produces: mountains of half-finished work piling up in front of the one machine that cannot keep up.

Hiring has an exact equivalent, and most teams mistake it for health. The job is posted everywhere. Two hundred applicants. Screens booked wall to wall. It feels like momentum. It is the busy factory: activity at every station, nothing shipping. Volume at a non-bottleneck step does not produce hires. It produces inventory, and the pile-up makes things worse in a specifically Goldrattian way: every marginal candidate who reaches the decision stage consumes a slice of the scarcest resource in the system, the decision-maker's attention, and gives back nothing.

The fix is not to work less. It is to stop measuring stations by how busy they are and start measuring the system by what comes out the end. "Resumes screened this week" is a utilization number. "Offers signed this quarter" is the goal.

Why searches age even when nothing is broken

One more idea from the book explains the most maddening pattern in hiring: the search where every individual step seemed fine and the role still took five months. Goldratt demonstrates it with a Boy Scout hike and a matchstick game: in any chain of dependent steps where each step's timing varies, delays accumulate and speedups do not. A fast week at one stage cannot be banked; the candidate still waits for the next stage. A slow week is never recovered; it pushes everything behind it. The chain drifts one direction only.

Hiring is exactly such a chain: source, screen, interview, feedback, decide, approve, offer, sign. Every step depends on the one before it, and every step's timing varies with travel, workload, and vacations. So the pipeline ages by default, with no villain and no broken step, unless something actively manages the hand-offs. That is why "we have a good process" and "our roles sit open for months" are both true in the same company. The process is fine. The chain is unmanaged. The countermeasures are the unglamorous ones: visible queues, named owners for every pending decision, reminders that fire without anyone having to remember. Boring on purpose. Flow is boring.

The five focusing steps, for a hiring team

The Goal's method is five steps, applied in order. Most teams do them in reverse: they buy capacity (step four) before they have identified the constraint (step one).

1 Identify the constraint

Find where candidates wait longest. Not where your team works hardest; where work sits. Pull your last few searches and look at queue time by stage: days awaiting feedback, days awaiting approval, days between verbal yes and signed letter. The longest queue is your constraint. It is probably a person's attention, not a tool.

2 Exploit the constraint

Before adding anything, squeeze what you have. Decision-maker attention must never be idle (no candidate waiting on a decision nobody has been asked to make) and never wasted (every decision arrives prepared: criteria, evidence, and panel input side by side, so it takes minutes, not a meeting). And check quality before the bottleneck: a candidate who fails a hard requirement should never consume an interview slot.

3 Subordinate everything else

Run every other step at the pace of the constraint, even if that means slowing down. Release candidates at the rate decisions can be made, not the rate applications arrive. And lock the brief before anything moves: comp, hard requirements, who approves what. Every re-litigated requirement is a stoppage at the bottleneck, which means a stoppage for the whole search.

4 Elevate the constraint

Only now add capacity: delegate screens, add a coordinator, bring in outside help for stages your team cannot staff. Most teams do this first and wonder why it does not help. Capacity at a non-bottleneck changes nothing; capacity at the bottleneck changes everything, but only after steps two and three stopped wasting the capacity already there.

5 Go back to step 1

When you fix the constraint, it moves. Speed up offers and the constraint becomes feedback; speed up feedback and it becomes scheduling. This is success; it just does not feel like it. The failure mode is inertia: keeping the rules built for the old constraint after it has moved. Re-ask the question every few searches.

The Herbie question

The Goal's most famous scene: Alex leads a Boy Scout hike that keeps stretching out because one slow scout, Herbie, sets the pace for the whole line no matter where he walks. The fix is not to yell at Herbie. It is to put Herbie at the front, lighten his pack, and organize the entire hike around his pace.

Every search has a Herbie, and it is usually the hiring manager: the one person whose judgment cannot be delegated and whose calendar cannot be expanded. Most processes treat the hiring manager as one stop among many, then wonder why everything queues behind them. The Goldratt move is the opposite: accept that this person sets the pace, put them at the front, and build the line around them. Lighten the pack (everything that is not judgment, someone or something else carries), then let their pace be the pace, visibly, so nobody floods the line with more than they can walk. **Stop asking your constraint to keep up with the process. Make the process keep up with the constraint.**

What to measure on Monday

Flow numbers, worth reporting: queue time at decision points (days each active candidate has been waiting on feedback, approval, or an offer: your constraint finder); verbal yes to signed letter (the most perishable interval in the pipeline: if it is more than a couple of days, this is where your searches are dying); and time to hire, stage by stage, so you can see which link stretched.

Volume numbers, fine to know, dangerous to manage by: applicants per posting, resumes screened, interviews held per week. All utilization measures. All the busy factory.

And one question for your next pipeline review, borrowed straight from the book. For each stalled candidate, not "what is happening with them?" but **"what are they waiting on, and who owns it?"** The first question produces status updates. The second produces flow.

Hireware is a hiring operations platform, and this guide is, candidly, its worldview: lock the brief before anything moves, keep the decision-maker's queue short and prepared, treat a slow offer as a system failure rather than bad luck, measure flow instead of activity. But you do not need our product to use any of this. You need your last three searches, an honest look at where the waiting happened, and a Monday.

The Goal, by Eliyahu M. Goldratt and Jeff Cox, is published by North River Press. Read it; it is a novel, it moves fast, and it will ruin status meetings for you in the best way.