

TRAINING MODULE

Bottleneck Elimination

The Theory of Constraints, from first principles

Based on Eliyahu M. Goldratt — The Goal (1984), and the Theory of Constraints literature

What You Will Be Able to Do After This Module

- 1 Define “constraint” correctly, and explain why a plant can have only ONE at a time.
- 2 Calculate which station is the constraint from real cycle-time data, by hand.
- 3 Apply Goldratt’s Five Focusing Steps in the correct order, without skipping any.
- 4 Explain Drum-Buffer-Rope well enough to defend it to a plant manager.
- 5 Recognize the two most common mistakes junior IEs make with bottlenecks — and avoid them.

Takt Time vs. Cycle Time

Every idea in this module rests on comparing these two numbers. Confuse them, and everything downstream breaks.

TAKT TIME

The RATE the customer demands — not a measurement of your line at all.

Takt = Available Working Time ÷ Customer Demand

Example: 27,000 sec shift ÷ 600 pcs/shift required = 45 sec Takt.

Set by the MARKET. You do not “improve” Takt — you meet it.

CYCLE TIME (CT)

The time YOUR station actually takes to complete one cycle of its work.

Measured with a stopwatch / MTM — station by station, real data

Example: Station 3 actually takes 52 sec to finish its work.

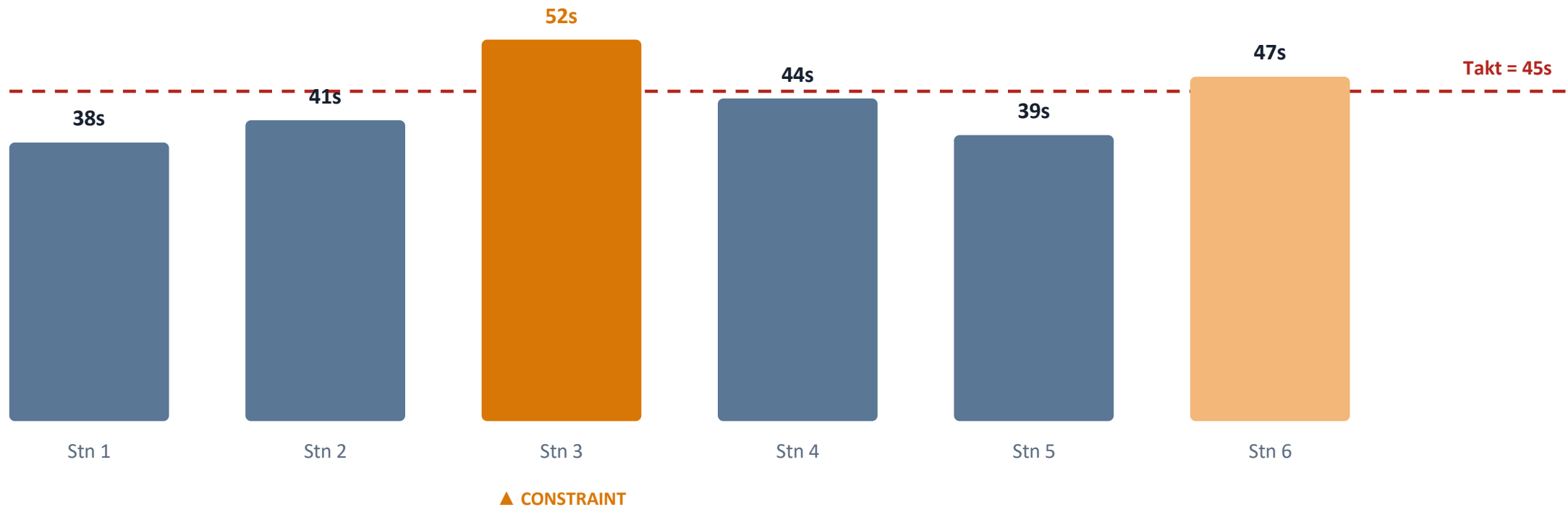
Set by the PROCESS. This is exactly what you're allowed to improve.

Rule to memorize: If CT > Takt at any station, that station cannot keep up — the whole line loses output at its rate.

THE CORE IDEA

A Line Only Moves as Fast as Its Slowest Station

Picture six stations in a row. Five of them are fast. One is slow. Watch what happens to the whole line's output:



Whatever Station 3 can push out — one unit every 52 seconds — is ALL the line can ever produce, no matter how fast the other five stations run.

Theory of Constraints

*“An hour lost at a bottleneck is an hour lost for the entire system.
An hour saved at a non-bottleneck is a mirage.”*

Goldratt’s revolutionary claim, told as a novel about a plant manager: a system's output is governed by its ONE weakest link — not by the average performance of every station. Improving a non-constraint station feels productive but changes nothing at the plant’s exit gate.

This is the single most important mental shift for a junior IE moving from “improve everything a little” to “improve the one thing that matters.”

THE GOAL

Eliyahu M. Goldratt
1984

Goldratt's Five Focusing Steps

Always in this order. Skipping a step — especially jumping straight to Elevate — is the #1 mistake junior IEs make.



The rest of this module walks through each step with the SAME worked example — Station 3 at 52 seconds against a 45-second Takt — so you see one problem solved end to end.

1

STEP 1

Identify

Rank every station — Man, Man-Machine, AND Auto — by CT vs Takt. Name the ONE constraint. Not a list. One.

A machine is just as eligible as an operator — in The Goal, the constraint is a machine (the NCX-10).

WORKED EXAMPLE — 6 STATIONS, TAKT = 45s

Station	CT (s)	vs Takt (45s)	Verdict
Stn 1	38	-7	OK
Stn 2	41	-4	OK
Stn 3	52	+7	▲ HIGHEST — CONSTRAINT
Stn 4	44	-1	OK
Stn 5	39	-6	OK
Stn 6	47	+2	Over Takt — watch this one

WHY STATION 3, NOT STATION 6?

TOC is explicit: name ONE constraint, not a shortlist. Station 3 is the single highest CT, so it is THE constraint. Station 6 stays on your radar — you'll come back to it in Step 5.

COMMON JUNIOR-IE MISTAKE

Treating Stations 3 AND 6 as “two bottlenecks” and splitting effort between them. TOC's whole point is focus — diluting it across a list defeats the method.

2

STEP 2

Exploit

Get everything possible out of Station 3 alone — without spending a rupee. This step is free. Always do it fully before Step 4.

Goldratt's dice-game chapter: variation at the constraint directly steals capacity — not just averages.

THREE FREE LEVERS — IN ORDER OF WHERE TO LOOK FIRST

ECRS — Eliminate

Any element that adds zero value — searching, re-grip, walking for a part that could be at hand. Removing it costs nothing and never comes back.

– 4s

High C2C Variation

A station that averages 52s but swings 45–60s is losing capacity to instability, not slowness. Standardize the method first — before touching layout or tooling.

– 2s

Non-Value-Add (NVA) Time

Waiting, inspection-by-habit, double-handling that the process doesn't actually require. Different from Eliminate: NVA often needs a small method change, not just deletion.

– 1s

RUNNING TOTAL: 52s – 7s of Exploit savings = 45s

This example resolves fully at Exploit — 45s meets the 45s Takt exactly. No spending needed. That won't always happen; see Step 4 for when it doesn't.

3

STEP 3

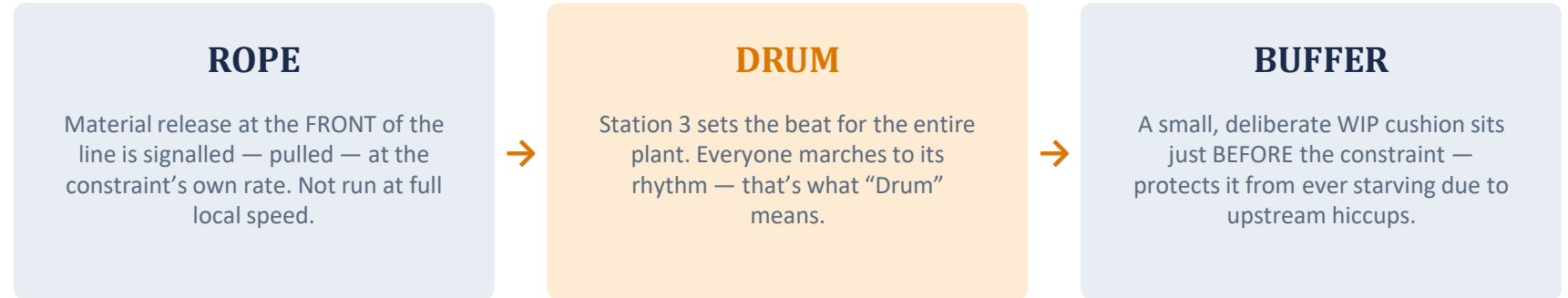
Subordinate

The hardest step to accept: every OTHER station must slow down to match the constraint. Deliberately. Not improved.

Running upstream stations at full speed just piles up WIP the constraint can't absorb — it doesn't help.

DRUM — BUFFER — ROPE

The mechanism that makes subordination work in practice:



Line order shown left to right: material release → the constraint → protective buffer directly ahead of it.

SUBORDINATE CHECK ON STATION 2 (JUST UPSTREAM OF THE CONSTRAINT)

Station 2 runs at 41s, the constraint now runs at 45s → Station 2 has 4s of protective capacity per cycle. That's enough slack to keep the constraint fed. Verdict: SUFFICIENT — no action needed here.

COMMON JUNIOR-IE MISTAKE

"Speeding up" non-constraint stations because it feels like progress. It only grows WIP inventory in front of the constraint — output at the plant's exit gate does not change at all.

4

STEP 4

Elevate

Only reached if Exploit truly wasn't enough. This is where you spend money — and only here.

In our worked example, Exploit already closed the gap — so this station needs no Elevate spend. Shown here as the general method.

IF EXPLOIT ISN'T ENOUGH, IN THIS ORDER

1

Extra manpower / an added shift on this station only

LOW COST

2

Parallel fixture — duplicate the constraint step, split the load

MED COST

3

Automation / machine upgrade at the constraint station

HIGH COST

ALWAYS CHECK FIRST — A FREE ELEVATE MAY ALREADY EXIST

If a Line Balance study exists for this line, check it before spending anything — a station elsewhere may already have accepted moves that free up capacity at the constraint for free.

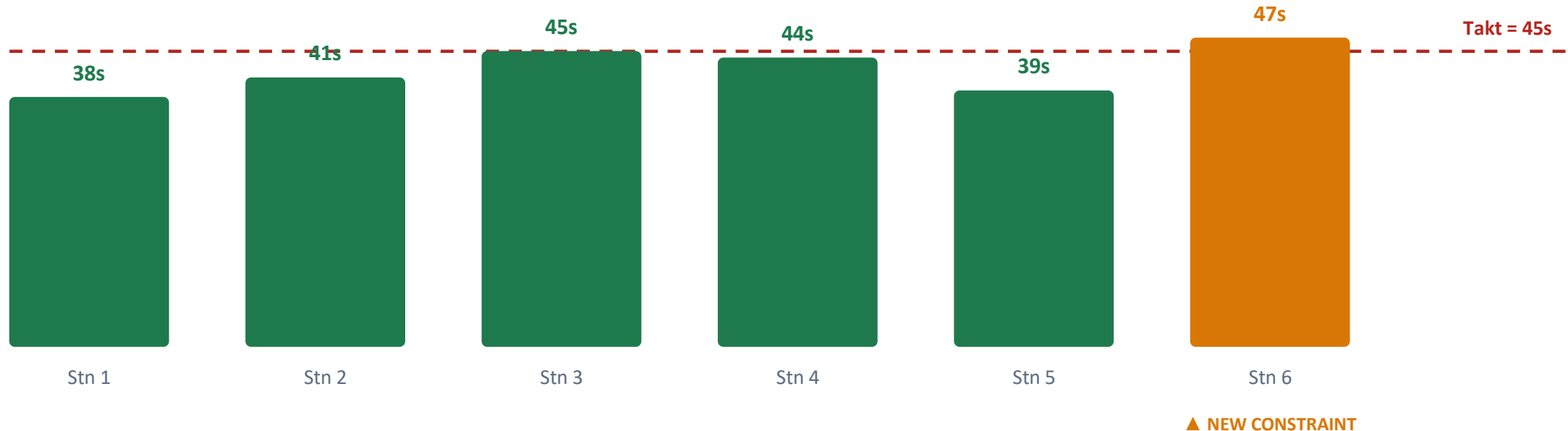
COMMON JUNIOR-IE MISTAKE

Proposing Elevate (a capex request) in Week 1, before Exploit has even been tried. It almost always gets rejected — and rightly so.

STEP 5

Repeat — The Constraint Moves, the Work Never Stops

Fix Station 3 down to 45s (Takt), and the plant's constraint doesn't disappear — it moves to whichever station is now the new highest CT.



Station 6, flagged back in Step 1 as “over Takt — watch this one,” is now THE constraint at 47s. Restart at Step 1 — with Station 6. This is continuous improvement, not a one-time project.

Five Mistakes That Give Junior IEs Away



Naming multiple “bottlenecks” instead of the one true constraint — dilutes every fix that follows.



Speeding up non-constraint stations because it feels productive — it only grows WIP, output stays flat.



Requesting Elevate spend before Exploit has been fully tried and shown insufficient.



Ignoring machine-paced (Auto / Man-Machine) stations as candidates — a machine is as eligible as an operator.



Treating this as a one-time project instead of Step 5’s continuous loop — the constraint always moves somewhere.

What Bottleneck_Elimination.xlsx Does For You

Everything on the last five slides, done automatically from your Production Study — in seconds, on real station data, every time you re-run it.

1

Identify

Ranks every station automatically — Man, Man-Machine, Auto — and names the constraint, with the same downstream tie-break logic you just learned.

2

Exploit

Finds every ECRS-Eliminate element and high-C2C element at the constraint automatically, and opens them as an interactive Accept / Reject debate.

3

Subordinate

Checks the immediately upstream station's protective capacity for you — the exact 4-second check you just did by hand.

4

Elevate

Cross-checks Line_Balance_Optimizer.xlsx if it exists alongside — a free Elevate option may already be sitting there.

5

Repeat

Simulates the constraint fixed and previews the next one — so you walk in already knowing Step 5's answer.

KEY TAKEAWAYS

Recap — Say This Back In Your Own Words

- A plant has exactly ONE constraint at a time — not several.
- Fix it in this order: Identify → Exploit → Subordinate → Elevate → Repeat. Never skip ahead.
- Exploit is free. Elevate costs money — earn the right to spend by exhausting Exploit first.
- Subordinating other stations feels like giving something up. It is the method working correctly.

TALK IT THROUGH — 3 QUESTIONS FOR YOUR NEXT LINE WALK

1. On your own line right now, which station would you name as THE constraint — and can you defend it with CT vs Takt numbers, not a guess?
2. Pick one Exploit lever (Eliminate, C2C, NVA) and find one real example at that station today.
3. If you proposed an Elevate spend tomorrow, could you show the Exploit step was already fully tried? If not — what's missing?