

The Theory of Constraints

The output of a system is determined by ONE constraint

A chain is as strong as its weakest link. Therefore, if you wish to improve the system, find and improve the constraint (or “bottleneck”)

The 5 Focusing Steps

0. Make the goal of the system explicit

This is the most difficult step:

- Ask people about the goal of their organisation. You’ll most likely either get no answer or different (even conflicting) answers.
- Does everyone agree on **one** goal (or the priority between goals)?
- How do you measure that you’re getting closer to the goal?

Look at the **whole system**. Where does it start? Where does it end?

1. Find the constraint

There’s always one! Symptoms to look for:

- Resource is heavily stressed
- Work in progress piles up in front of the resource
- Resources downstream from constraint are regularly idle

2. Exploit the constraint

The output of the system == the output of the bottleneck resource. Therefore, get as much **useful** work out of the bottleneck as possible:

- Remove any non-productive work, avoid distractions and task-switching
- Ensure that there is always something to work upon, so that the bottleneck is never idle (create a buffer of work before the bottleneck)
- But don’t overload the bottleneck with too much work in progress: let the bottleneck “pull” work from the buffer
- Carefully prioritize what the bottleneck works on

Always exploit first: the bottleneck is already paid for; get the most out of it.

3. Subordinate everything else to the constraint

Those resources that are not bottlenecks have (by definition) some extra capacity. Use that capacity to:

- Help the bottleneck where possible
- Take away some work from the bottleneck
- Everyone else works at the rhythm indicated by the bottleneck
- Ensure that the bottleneck only receives the highest quality input
- Ensure that all outputs of the bottleneck are treated carefully, so that they’re not wasted

If there are no more exploits, subordinate the other resources. The other resources are already paid for. This might mean slowing other resources down, but the goal is not to keep people busy; the goal is to generate useful output.

4. Elevate the constraint

Invest resources and time to improve the performance of the bottleneck, so as to increase their output (and thus the output of the system) by:

- Training, books, conferences, exercising
- Coaching
- Better tools
- More people to do the work (but take into account learning time)

This improvement requires investment and time; so only do this AFTER exploit and subordinate (which are “free”) offer no more improvement.

5. Again. Do not let inertia become the constraint.

There’s always a bottleneck. There’s always a way to improve further. Be sure to go back to step 0 occasionally (is the goal still relevant?). Always go back to step 1 (Identify bottleneck), because if you improve in one area, another might become the bottleneck. **“If you fix your worst problem, your number two problem gets a promotion.”** But try to keep the bottleneck under your control!

6. Change the system

Sometimes, there’s no way to improve the existing system further. You might need to examine the way the system is structured and change it. But that’s a lot more difficult, you will encounter:

- “Hidden rules” that are not explicit, but still followed: *“That’s the way we’ve always done things around here!”* Why did we decide to do it that way?
- *“You can’t touch that! You’ll break everything!”* How/where will it break?
- *“You don’t understand!”* Can you explain it to me?

See also:

“The Goal”, “It’s Not Luck”, “Necessary but not sufficient” – Eliyahu Goldratt

“Thinking for a Change” – Lisa Scheinkopf

“Agile Management for Software Engineering” – David Anderson

“Management Dynamics - Constraints Accounting” – John & Pamela Caspari

“Practical Lean Accounting” – Brian Maskell & Bruce Baggaley

“Critical Chain Project Management” – Lawrence P. Leach

“Project Management in the Fast Lane” – Robert C. Newbold

“Deming and Goldratt” – Domenico Lepore and Oded Cohen

“The Toyota Way” – Jeffrey K. Liker

“Implementing Lean Software Development” – Mary & Tom Poppendieck

See also books on “Lean” and “The Toyota Way” for specific techniques to exploit, subordinate and elevate. More books on Lean, Theory of Constraints, Systems Thinking and Agile at <http://wiki.systemsthinking.net/Systemsthinking/BookList.html>

Pascal Van Cauwenberghe
pascal@navima.be
Nayima
<http://www.navima.be>

Portia Tung
portia@portiatung.org
Exoftware
<http://www.portiatung.org>

