



You Know That Your Team is Too Slow

Did you know that it's your fault?



**RobbinsNextGen**



The “Five Minute Review” Lie

You tell yourself: *“I just need 5 minutes to check this before it goes out. Just to be safe.”*

You think you’re adding **quality** control.

Actually, you’re adding **“Latency.”**

The Physics of the Queue

In systems engineering, wait times do not grow **linearly**. They grow **exponentially**.

The Rule: As your calendar utilization passes 80%, the wait time for your team approaches infinity.

If you are "back-to-back" in your scheduling, you are a brick wall.

Touch Time vs Wait Time

Touch Time: The 10 minutes you spend reviewing the slide deck.

Wait Time: The 6 hours the deck sat in your inbox while you were in meetings.

The Result: You delayed the project by 6 hours to add 10 minutes of value.

That is a **-97% ROI** on speed.

You Are Not the Safety Net

We review work because we want to catch mistakes.

But in a fast-moving market, Speed is a feature.

A "perfect" decision made three days late is a wrong decision.

You are not the safety net. You are the anchor.

Engineer a Constraint & Accountability Based Flow

1. The WIP Limit (Work In Progress):

- Each team/person gets 1 active approval slot in your queue.
- They cannot submit Item B until you clear Item A.
- Result: They are forced to prioritize the most critical item.

2. The "Definition of Ready" Gate:

- No "quick looks." Requests must have:
(1) Data, (2) Recommendation, (3) Risk Analysis.
- If incomplete? Auto-moved to the back of the line.

3. The "Urgency Tax":

- If it cannot wait 24 hours, they must Call You.
- Slack/Email is banned for "Urgent" approvals.
- Social friction kills false urgency.



The Latency Cheat Sheet

1. **Utilization:** If you are too busy, the team is too slow.
2. **Latency:** Wait time $>$ Work time.
3. **Constraints:** Set - and follow - rules for prioritization and attention.

Push accountability to the team, and focus on what you can control - your time.

The Paradox of Importance

- Your ego wants to be **vital** (the one who solves the problems).
- The business needs you to become **obsolete** (the one who built the system).
- If the machine grinds to a stop when you step away, you didn't build a high-performance team...
- ...you built a dependency.

*Your "slow" team is your fault.
So fix it.*

For help or information connect
via LinkedIn or my site.

RobbinsNextGen.com

