👉 Most automation efforts begin with tools. But the successful ones begin with systems thinking.

🎗️ A skill that is quietly becoming a leadership superpower is Systems Thinking.

📰 *The Future of Jobs Report 2025* flags it as a FAST-RISING core skill.

👉 Systems thinking leaders:

✅ See the bigger picture

✅ Understand how different parts influence each other

✅ Avoid quick fixes that create bigger problems later

As division manager of Accounts Payable, I oversaw the RPA implementation for invoice processing.

It sounded simple: automate repetitive tasks.

But the reality? Complexity, interconnections, and up/downstream effects.

👉 It was a successful implementation but there were some lessons learned along the way:

💡 9 Systems Thinking Practices for SMARTER Automation Leadership

1️⃣ **Check what's connected**

Invoices lagging after a launch? Might not be a bug. Vendors may not be looped into how the new bots work.

→ It's not always broken technology. Could be misaligned people.

2️⃣ **Look for repeat loops**

If repeat problems keep bouncing back to vendors or ops, you haven't solved them.

You've just shifted them.

→ Faster doesn't imply it's smoother.

3️⃣ **Slow down your judgements**

Delays might seem like tech glitches, but the real issue could be strict rules, bad inputs, or thresholds no one flagged.

→ Don't jump to blame. Trace the trail first.

4️⃣ **Watch for delayed impact**

Auto-approvals work fine on Day 1. Two weeks later, stakeholders are stuck with rigid logic that won't bend.

→ Design and build for both now and later.

5️⃣ **Scan for up/downstream effects**

Bots clean up one task. Suddenly vendors are pinging procurement nonstop.

→ Trace consequences across teams.

6️⃣ **Spot the bigger pattern**

If every automation stalls, it may not be the tools. It's how change lands across teams.

→ If it keeps happening, it's not random, it's systemic.

7️⃣ **Steer the change. Don't just hit send.**

A flawless rollout doesn't mean people adopt it. Until they feel clarity and value, it won't

stick.

→ Rollouts need buy-in, not just instructions.

8️⃣ **Welcome multiple perspectives**

An AP analyst notices what the bot keeps missing: details the project team overlooked.

→ Those closest to the work often hold the feedback loop.

9️⃣ **Balance action with reflection**

Go-live isn't the end. Skipping the "what did we learn?" moment brings the same pain next time.

→ Reflection isn't optional. It has to be a strategy.

🎗️ ***Personal note:***

We learned how easy it was to "fix" one problem and unknowingly cause another.

Building a habit of mapping connections and leaning into systems thinking can change how you approach tech implementations and leadership.

👉 Which of these practices do you use most? Which one will you try next? 💬