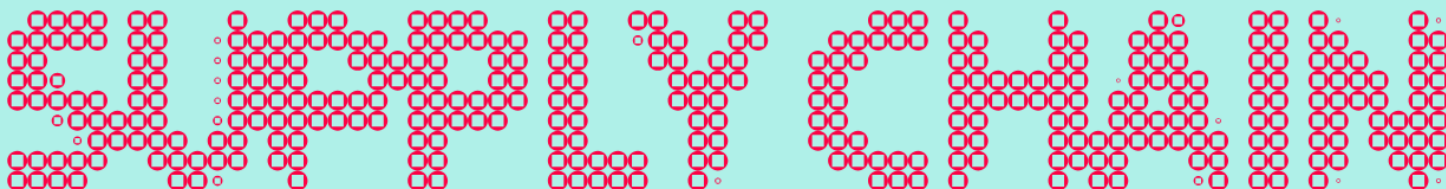


ศพพลายเซนที่ไม่ต้อง มีคนคอยไล่ตาม

80% ของเวลาหมดไปกับ 20% ของเคสที่ผิดปกติ
เล่มนี้ว่าด้วยการคืน 80% นั้นกลับมา

ซีรีส์ผังความเป็นเจ้าของงาน (Task Ownership Map) - เล่ม 03 จาก 4



สิ่งที่คุณจะได้จากเล่มนี้

ข้อค้นพบหลัก 3 ข้อ

01

งานที่กินเวลามากที่สุดไม่ใช้การวางแผน แต่คือการ “ไล่ตาม”

งานวางแผนมีเครื่องมือรองรับมาหลายสิบปี แต่งานที่กินเวลาจริงในแต่ละวันคือการ **ไล่ตามเคสที่หลุดแผน** — โทรตามของที่ยังไม่มา ทวงเอกสารที่ขาด ไล่หาว่าตัวเลขไม่ตรงตรงไหน ไม่มีระบบไหนตามให้จนจบ จึงตกกับคน และนี่คือจุดที่ Agent เข้ามาช่วยได้ตรงที่สุด

02

ซัพพลายเชนคือฟังก์ชันที่ Agent รับงานไปทำได้มากที่สุดในทั้งสี่เล่ม — 9 จาก 15 งาน

ไม่ใช่เพราะงานง่ายกว่า แต่เพราะงานส่วนใหญ่ **มีกฎชัดเจนที่สุด วัตถุประสงค์ชัดเจน และอ้างอิงจากข้อมูลที่มีอยู่ในระบบแล้ว** — คำถามแบบนี้ตอบได้ด้วยข้อมูล ไม่ต้องใช้ดุลพินิจ และงานที่ตอบได้ด้วยข้อมูล Agent รับไปปิดได้เลย

03

ตัวชี้วัดที่ถูกคือ “อัตรา exception ที่ปิดได้เอง” ไม่ใช่ความแม่นยำของ forecast

ทีมจะลงแรงทำ forecast เต็มที่แค่ไหน ก็ยังต้องมานั่ง **ดับไฟรายวัน** อยู่ดี เพราะของที่ไม่มาตามกำหนดมาจากความไม่แน่นอนฝั่งซัพพลายเออร์และการขนส่ง ไม่ใช่จากโมเดล องค์กรที่วัดถูกจึงวัดว่า เคสที่หลุดแผนที่เปอร์เซ็นต์ที่ระบบปิดเองได้

เล่มนี้เขียนสำหรับใคร

ถ้าคุณคือ	เล่มนี้ตอบอะไรให้คุณ
หัวหน้า / ผู้จัดการสายซัพพลายเชน (SCM)	งานไหนในทีมควรให้ระบบปิดเอง งานไหนต้องเก็บไว้ให้คน และจะขีดเส้นนั้นด้วยหลักอะไร
ทีมโลจิสติกส์ (Logistics)	ทีมที่ใช้เวลาครึ่งวันไปกับการโทรตามของ ทวงเอกสาร และประสานเคสที่ค้าง
ทีมวางแผนความต้องการ (Demand Planning)	ลงแรงกับ forecast เต็มที่แล้ว แต่ยังต้องมานั่งดับไฟรายวันอยู่ดี
ฝ่ายปฏิบัติการ / ผู้บริหาร	กำลังขังใจว่าจะเอา AI Agent เข้ามาในกระบวนการตรงไหนก่อน โดยยังคุมความเสี่ยงได้

สารบัญ

01	สถานการณ์จริง — วันที่แผนดี แต่เวลาหมดไปกับเรื่องที่ไม่อยู่ในแผน	3
02	Task Ownership Map — ผังความเป็นเจ้าของงาน	4
	Task Ownership Map ของซัพพลายเชน — 15 งาน	5
03	Agentic Design — จากของค้างส่ง สู่อัตโนมัติ	6
04	CoE — AI Center of Excellence เข้าไปทำงานอย่างไร	8
05	บทเรียน 4 ข้อ	9
06	Business Takeaways และจุดเริ่มต้น 90 วัน	10

หมายเหตุเรื่องตัวเลขในเล่มนี้

ตัวเลขผลลัพธ์และเส้นวงเงินตัวอย่างในเล่มนี้ (เช่น 80/20 · เส้นอนุมัติ [50,000] บาท · ต้นทุนเคสตัวอย่าง [35,000] บาท) เป็น illustrative benchmark เพื่อประกอบคำอธิบาย ไม่ใช่ผลที่วัดจากลูกค้ารายใดรายหนึ่ง ค่าที่อยู่ในวงเล็บเหลี่ยม [] เป็นตัวเลขตัวอย่างที่เรารวบรวมกันก่อนขึ้นเล่มจริง

ซัพพลายเชนที่ไม่ต้องมีคนคอยไล่ตาม

2

บทที่ 01

วันที่แผนดี แต่เวลาหมดไป กับเรื่องที่ไม่อยู่ในแผน

องค์กรส่วนใหญ่เก่งเรื่องวางแผน แต่เวลาของทีมกลับหมดไปกับการไล่ตามสิ่งที่ไม่เป็นไปตามแผน — และนั่นคือส่วนที่ไม่มีเครื่องมือชุดไหนเคยออกแบบมารองรับ

8:12 น. เปิดอีเมล 40 ฉบับ

เข้าวันจันทร์ Planner คนหนึ่งเปิดคอมพิวเตอร์ ในกล่องอีเมลมีข้อความค้างอยู่ 40 ฉบับตั้งแต่คืนวันศุกร์ ประมาณครึ่งหนึ่งเป็นเรื่องเดียวกัน — **ของไม่มาตามกำหนด** ซัพพลายเออร์เจ้าหนึ่งเลื่อนส่งสามวัน อีกเจ้าส่งมาแต่จำนวนขาด 200 ชิ้น คลังแจ้งว่าเลขล็อตไม่ตรงกับใบสั่งซื้อ ฝ่ายขายส่งต่ออีเมลลูกค้ารายใหญ่ที่ถามว่า “ออเดอร์เดือนนี้จะได้ครบไหม”

ก่อนเธอจะได้เริ่มงานที่ตั้งใจไว้ — กบฏวางแผนจัดซื้อไตรมาสหน้า — เข็มนาฬิกาก็เดินไปถึงเที่ยงแล้ว เข้าถึงเข้าหมดไปกับการไล่ตาม ไม่ใช่การ วางแผน นี่ไม่ใช่วันที่แยเป็นพิเศษ นี่คือวันปกติ

80%

ของเวลาหมดไปกับเคสที่หลุดแผน
(ตัวเลขเชิงประกอบ)

20%

ของเคสที่ผิดปกติ — แต่กินเวลาเกือบ
ทั้งหมด

2 งาน

วางแผน vs ไล่ตาม ถูกกองรวมเป็นงาน
เดียวจนแยกไม่ออก

สองงานที่ถูกนับรวมเป็นงานเดียว

งานวางแผน (planning) คืองานที่ต้องใช้สมองและประสบการณ์ — จะสั่งของล่วงหน้าเท่าไร กันสต็อกเพื่อแค่ไหน รับมือฤดูกาลอย่างไร นี่คือคุณค่าที่แท้จริงของทีม และเป็นงานที่องค์กรจ้างคนเก่งมาทำ

งานไล่ตาม (chasing / expediting) เกิด *หลัง* วางแผนไปแล้วแต่ความจริงไม่เป็นไปตามแผน งานนี้ไม่ต้องใช้ดุลพินิจมากนัก แต่ต้องใช้ **ความต่อเนื่องและความไม่ลืมหืม** — ต้องตามจนจบ ต้องเปิดหลายระบบมาปะติดปะต่อว่าเกิดอะไรขึ้น ในองค์กรส่วนใหญ่ งานประเภทที่สองกินเวลามากกว่าอย่างมีนัยสำคัญ

ทำไมระบบเดิมถึงไม่ช่วยตรงนี้

ระบบวางแผนทรัพยากรองค์กร (ERP) เก่งเรื่องบันทึกว่า *ควร* จะเกิดอะไร ระบบจัดการคลังสินค้า (WMS) เก่งเรื่องบันทึกว่า *ของ* อยู่ที่ไหน แต่เมื่อของ *ไม่* มาตามที่ ERP บอกว่าควรมา ไม่มีระบบไหนทำหน้าที่ “ไล่ตามจนจบ” — มันแค่แสดงว่ามีความไม่ตรงกัน แล้วปล่อยให้คนไปหาต่อเองว่าเกิดจากอะไร

ข้อสรุปของบทนี้

ช่องว่างระหว่าง “**ระบบรู้ว่าคืออะไร**” กับ “**มีคนแก้จนจบ**” คือที่ที่เวลาของทีมหายไป — และคือที่ที่เสมนีจะพูดถึงตลอดทั้งเล่ม

บทที่ 02

Task Ownership Map

ผังความเป็นเจ้าของงาน

ผั่งนี้ไม่ได้ถามว่า “AI ทำงานนี้ได้ไหม” เพราะเทคโนโลยีวันนี้ทำได้ทุกอย่างที่เป็นดิจิทัล คำถามที่ถูกต้องกว่าคือ “งานนี้ควรให้ใครเป็นเจ้าของ” — และเราตอบด้วยตัวแปรสามตัว

สามตัวแปรที่ใช้ตัดสิน

1 · ปริมาณและความถี่ งานนี้เกิดบ่อยแค่ไหน ทำซ้ำมากไหม งานที่เกิดขึ้นไม่กี่ครั้งต่อปีมักไม่คุ้มกับต้นทุนการทำให้ระบบ	2 · ความแปรผันของเคส แต่ละเคสหน้าตาเหมือนกันหรือต่างกัน ทุกครั้ง งานที่มีรูปแบบชัดและตัดสินด้วยกฎได้ Agent รับไปทำงานได้	3 · ผลกระทบถ้าผิด ถ้าผิดพลาด ความเสียหายรุนแรงแค่ไหน แก้กลับได้ไหม ตัวแปรนี้เป็นกติกาลบล้างที่อยู่เหนือสองตัวแรก
---	--	--

เอาปริมาณ × ความแปรผัน มาไขว้เป็นตาราง

ปริมาณต่ำ		ปริมาณสูง
แปรผันสูง	พิจารณาเป็นรายกรณี ดูเป็นงาน ๆ ไป	AI-assisted Agent เสนอ คนกดยืนยัน
แปรผันต่ำ	ยังไม่แตะ ทำระบบแล้วยังไม่คุ้ม	Agent-owned Agent ปิดงานเอง คนสุ่มตรวจ

กติกาขอล้าง (override) ที่สำคัญที่สุด

ผลกระทบถ้าผิดสูง → ต้องเป็น **Human-owned** เสมอ ไม่ว่าจะตกอยู่ช่องไหนก็ตาม — เส้นนี้อยู่เหนือทุกช่องในตาราง

ความต่างของสามคำที่จะเห็นในตาราง

Agent-owned Agent ทำงานจบเอง คนดูผลย้อนหลังได้ แต่ไม่ต้องกดอนุมัติทีละเคส	AI-assisted Agent เตรียมและ <i>เสนอคำตอบ</i> แล้วคนเป็นผู้กดยืนยัน	Human-owned คนเป็นผู้ตัดสิน Agent เพียงเตรียมข้อมูลให้พร้อม ไม่เสนอคำตอบแทน
---	--	---

บทที่ 02 · ตาราง

Task Ownership Map ของซัพพลายเชน — 15 งาน

ลงตารางจริงใน 4 กลุ่มงานหลัก — ตัวเลขและคำตัดสินเป็นการประเมินตั้งต้น องค์กรควรทำตารางนี้ด้วยงานของตัวเอง

#	งาน (task)	ปริมาณ	แปรผัน	ผลถ้าผิด	คำตัดสิน
A · การพยากรณ์และการวางแผน (Demand & Planning)					
1	รวบรวมยอดขาย · สต็อก · lead time จากหลายระบบมาไว้ที่เดียว	สูง	ต่ำ	ต่ำ	Agent-owned
2	สร้าง forecast ตั้งต้นจากข้อมูลย้อนหลัง	สูง	ต่ำ	กลาง	Agent-owned
3	ปรับ forecast ตามข้อมูลที่ระบบไม่รู้ — โปรโมชัน ฤดูกาล ข่าวก้าว	กลาง	สูง	สูง	AI-assisted
4	อนุมัติแผนจัดซื้อรอบใหญ่	ต่ำ	สูง	สูงมาก	Human-owned
B · การสื่อสารกับซัพพลายเออร์					
5	ส่งและติดตาม PO จนกว่าซัพพลายเออร์จะยืนยัน	สูง	ต่ำ	ต่ำ	Agent-owned
6	ทวงเอกสารที่ยังขาดจากซัพพลายเออร์	สูง	ต่ำ	ต่ำ	Agent-owned
7	สรุปสถานะการส่งมอบทั้งหมดเป็นรายงานประจำวัน	สูง	ต่ำ	ต่ำ	Agent-owned
8	เจรจาเมื่อซัพพลายเออร์ขอเปลี่ยนกำหนดส่ง	กลาง	สูง	สูง	Human-owned
C · การจับคู่เอกสาร					
9	จับคู่ PO – ใบส่งของ – ใบแจ้งหนี้ (3-way match)	สูงมาก	ต่ำ	กลาง	Agent-owned
10	ตรวจหารายการที่ไม่ตรงกันและแจ้งผู้เกี่ยวข้อง	สูง	ต่ำ	กลาง	Agent-owned
11	ตัดสินว่าความคลาดเคลื่อนรายการไหนยอมรับได้	กลาง	สูง	สูง	AI-assisted
D · การจัดการข้อยกเว้น (Exception Handling) — หัวใจของเล่มนี้					
12	ตรวจจับของขาด/ของค้าง แล้วเปิดเคสอัตโนมัติ	สูง	ต่ำ	กลาง	Agent-owned
13	ไล่หาต้นตอของ exception ข้ามระบบ — ERP · WMS · อีเมล	สูง	สูง	กลาง	AI-assisted
14	ตัดสินใจแก้เคสเร่งด่วนที่กระทบลูกค้ารายใหญ่	ต่ำ	สูง	สูงมาก	Human-owned
15	ปิดเคสและบันทึกสาเหตุเข้าคลังความรู้เพื่อกันซ้ำ	สูง	ต่ำ	ต่ำ	Agent-owned

ศัพท์ในตาราง: PO (Purchase Order) = ใบสั่งซื้อ · lead time = ระยะเวลาตั้งแต่สั่งจนถึง · 3-way match = การจับคู่เอกสารสามใบ (ใบสั่งซื้อ-ใบส่งของ-ใบแจ้งหนี้) ให้ตรงก่อนจ่ายเงิน · exception = เคสที่หลุดจากแผน

ทำไมซัพพลายเชนถึงมีสัดส่วน Agent-owned สูงที่สุดในซีรีส์

9 จาก 15 งานเป็น Agent-owned — สูงสุดในทั้งสี่เล่ม (HR 9 · Marketing 7 · Procurement 5) เพราะงานส่วนใหญ่มีเกณฑ์ถูก-ผิดชัด หลักฐานอยู่ในระบบแล้ว และทำซ้ำปริมาณมหาศาล ส่วนอีก 6 งานที่เหลือคืองานที่ต้องใช้ดุลพินิจ ที่ผั่งนี้จึงใจเก็บไว้ให้คน

บทที่ 03

Agentic Design — จากของค้างส่งสู่การปิดเคส

ซัพพลายเชนคือฟังก์ชันที่ **การจัดการข้อยกเว้น (exception handling)** เป็นหัวใจ — และนั่นคือที่ที่การออกแบบ Agent ต้องละเอียดที่สุด เพราะเคสที่หลุดแผนไม่มีหน้าตาเหมือนกันสองเคส

สามระดับความพร้อมก่อนถึงวงจรเต็ม

ระดับ 1	ระดับ 2	ระดับ 3
มองเห็น Agent รวบรวมข้อมูลข้ามระบบมาแสดงที่เดียว ตรวจสอบว่ามีเคสหลุดแผน แล้วแจ้งเตือน คนยังเป็นผู้แก้ทั้งหมด · เลิกเสียเวลาเปิดหลายจอมาปะติดปะต่อเอง	เสนอทางแก้ Agent ไล่หาต้นตอและ เสนอ ทางแก้พร้อมเหตุผล คนเป็นผู้กดเลือก · คนตัดสินใจได้เร็วขึ้นมาก เพราะไม่ต้องสืบเองตั้งแต่ต้น	ปิดเคสเองในกรอบ Agent ลงมือแก้และปิดเคสเองได้ ภายในกรอบที่มนุษย์ตั้งไว้ล่วงหน้า เก็บกรอบเมื่อไรจึงส่งให้คน · เหลือเฉพาะเคสที่ต้องใช้ดุลพินิจจริง ๆ มาถึงคน

เป้าหมายไม่ใช่ระดับ 3 กับทุกเคส แต่คือ ให้เคสส่วนใหญ่จบที่ระดับ 3 และเหลือเฉพาะเคสที่ควรถึงคนจริง ๆ มาถึงคน

วงจรเต็ม: เคสของค้างส่ง (backorder) หนึ่งเคส

สมมติ — วัตถุดิบ SKU (รหัสสินค้าหนึ่งรายการ) หมายเลข **A-233** ควรถึงคลังวันนี้ 1,000 หน่วย เพื่อป้อนไลน์ผลิตพรุ่งนี้ แต่ของยังไม่มาต่อไปนี่คือสิ่งที่ Agent ทำ ทีละขั้น

ขั้น 1 — ตรวจสอบและเปิดเคสอัตโนมัติ **งานข้อ 12 · Agent-owned**

เที่ยงคืนที่ผ่านมา Agent เทียบกำหนดส่งใน ERP กับของที่รับเข้าจริงใน WMS พบว่า A-233 ถึงกำหนดรับแต่ยังไม่มีมารับเข้า จึงเปิดเคส “ของค้างส่ง” อัตโนมัติทันที ไม่ต้องรอให้ Planner มาเจอตอนเช้า และตั้งระดับความเร่งด่วนจากข้อมูลปลายทาง — A-233 ป้อนไลน์ผลิตพรุ่งนี้ จึงเป็นเร่งด่วน

ขั้น 2 — ไล่หาต้นตอข้ามระบบ **งานข้อ 13 · AI-assisted**

Agent เดินสืบข้ามสามแหล่งพร้อมกัน:

- ใน **ERP** — ใบสั่งซื้อออกไปถูกต้อง ซัพพลายเออร์ยืนยันแล้วเมื่อสัปดาห์ก่อน
- ใน **WMS** — ไม่มีบันทึกรับเข้า และไม่มีของรอที่ท่ารับ
- ใน **อีเมลซัพพลายเออร์** — พบอีเมลคืนวันศุกร์ แจ้งว่ารถขนส่งเสียระหว่างทาง ของจะล่าช้า 2 วัน

Agent ประติดปะต่อได้ว่า ต้นตอคือการขนส่งของซัพพลายเออร์ ไม่ใช่ความผิดพลาดในการสั่งหรือการรับ นี่คืองาน “แปรผืนสูง” จึงเป็น AI-assisted — Agent เสนอ ข้อเสนอ แต่ยังไม่ตัดสินใจแทน

ขั้น 3 — เสนอทางแก้พร้อมเหตุผล

Agent ประกอบทางเลือกให้ Planner สามทาง พร้อมข้อมูลประกอบ:

- (ก) รอของเดิม 2 วัน → ไลน์ผลิตหยุด 1 วัน กระทบออเดอร์ลูกค้า 3 ราย
- (ข) สั่งทดแทนจากซัพพลายเออร์สำรองที่มีของพร้อม → ต้นทุนสูงกว่าเดิม 8% แต่ทันไลน์ผลิต
- (ค) สลับคิวผลิต เอา SKU อื่นที่วัตถุดิบพร้อมขึ้นก่อน → ไม่มีต้นทุนเพิ่ม แต่ต้องประสานฝ่ายผลิต

บทที่ 03 · ต่อ

เส้นวงเงิน — หัวใจของ governance

นี่คือจุดที่การออกแบบ agentic ที่ดีต่างจากการปล่อยให้ Agent ทำแทนคนทั้งหมด องค์กรตั้ง **เส้นวงเงิน (threshold)** ไว้ล่วงหน้าว่า Agent สิ่งซื้อทดแทนเองได้ถึงเท่าไร โดยไม่ต้องขออนุมัติ

ขั้น 4 — เส้นแบ่งการอนุมัติ (human-in-the-loop threshold)

human-in-the-loop = การออกแบบให้มนุษย์อยู่ในวงจรการตัดสินใจ ณ จุดที่กำหนดไว้ ไม่ใช่ทุกจุด

สมมติองค์กรตั้งเส้นไว้ว่า

- ทางแก้ที่ **ไม่มีต้นทุนเพิ่ม** (เช่นทางเลือก ค) → Agent ดำเนินการเองได้เลย บันทึกไว้ให้ตรวจสอบย้อนหลัง
- สิ่งซื้อทดแทน **ไม่เกิน [50,000] บาทต่อเคส** → Agent สั่งเองได้ ภายในชีพพลายเออร์สำรองที่อนุมัติไว้แล้วเท่านั้น
- **เกิน [50,000] บาท หรือกระทบลูกค้าที่ติดธงว่าเป็นรายสำคัญ** → หยุด ส่งให้คนตัดสินใจ (เข้าสู่งานข้อ 14 · Human-owned)

ในเคสนี้ ทางเลือก (ข) ต้นทุนเพิ่ม 8% สมมติอยู่ที่ **[35,000]** บาท — ต่ำกว่าเส้น และชีพพลายเออร์สำรองอยู่ในรายชื่อที่อนุมัติแล้ว Agent จึงสั่งทดแทนเองได้ แต่เพราะ A-233 ป้อนออเดอร์ที่มีลูกค้ารายใหญ่ติดธงอยู่ เคสนี้เข้าเงื่อนไข “กระทบลูกค้ารายสำคัญ” Agent จึง **หยุด และส่งให้ Planner ตัดสิน** พร้อมแนบทางเลือกทั้งสามและข้อมูลสรุป

Planner ใช้เวลา 3 นาทีอ่านสิ่งที่ Agent เตรียมมา แล้วเลือกทางเลือก (ข) — เทียบกับเดิมที่ต้องสืบเองครึ่งวันก่อนตัดสินใจได้

ขั้น 5 — ลงมือและปิดเคส งานข้อ 15 · Agent-owned

เมื่อได้รับเลือก Agent ออกใบสั่งซื้อทดแทน ยืนยันกำหนดส่ง อัปเดตแผนใน ERP แจ้งฝ่ายผลิตว่าวัตถุดิบจะถึงทันคิว แจ้งฝ่ายขายว่าออเดอร์ลูกค้าไม่กระทบ แล้วปิดเคส — ทั้งหมดเป็นงานทำซ้ำที่มีขั้นตอนชัด จึงเป็น Agent-owned

ขั้น 6 — บันทึกกันซ้ำ งานข้อ 15 · Agent-owned

Agent บันทึกเข้าคลังความรู้: ต้นทุน ทางแก้ที่เลือก ต้นทุนที่เกิด เวลาที่ใช้ปิดเคส และตั้งข้อสังเกตว่าชีพพลายเออร์รายนี้มีเคสขนส่งล่าช้าเป็นครั้งที่สามในไตรมาส ข้อมูลนี้จะไปโผล่ในรายงานให้คนทบทวน ทำให้เคสประเภทเดิม **มีโอกาสถูกจัดการที่ต้นเหตุ ไม่ใช่มาดับไฟซ้ำทุกเดือน**

สิ่งที่วงจรนี้แสดงให้เห็น

ในหนึ่งเคส งานสลับมือระหว่าง Agent กับคนถึงสามครั้ง — Agent ทำขั้น 1 · เสนอในขั้น 2–3 · หยุดให้คนตัดสินใจในขั้น 4 · แล้วรับกลับมาทำต่อขั้น 5–6 นี่คือนิยามจริงของ agentic design ที่ดี: **การส่งไม้ต่อที่ออกแบบไว้ว่าจุดไหนใครถือ** โดยมีเส้นวงเงินเป็นตัวกำหนดว่าเมื่อไรต้องส่งไม้กลับมาที่คน

บทที่ 04

CoE — AI Center of Excellence

เข้าไปทำงานอย่างไร

การมี Agent เก่งกับการทำให้ Agent สร้างผลได้ทั้งองค์กรเป็นคนละเรื่อง ตัวเชื่อมคือ **AI Center of Excellence (CoE)** — ศูนย์กลางที่ทำให้ความสามารถของ AI กลายเป็นผลลัพธ์ที่ทำได้ ไม่ใช่ของเล่นของทีมใดทีมหนึ่ง

สามชั้นของ CoE

ชั้นแพลตฟอร์ม (Platform)

โครงสร้างพื้นฐานที่ Agent ทำงานอยู่บนนั้น — การเชื่อม ERP/WMS/อีเมล การจัดการสิทธิ์เข้าถึงข้อมูล การบันทึกร่องรอย (log) และการควบคุมความปลอดภัย ฐานที่ทุก use case ยืนอยู่

ชั้นคู่มือปฏิบัติ (Playbooks)

วิธีทำงานที่ตกผลึกเป็นแบบแผน — เส้นวงเงินอนุมัติของแต่ละประเภทเคส เกณฑ์จัดระดับความเร่งด่วน ขั้นตอนส่งไม้ต่อ Agent-คน สิ่งตกที่ 3 อธิบาย อยู่ในชั้นนี้

ชั้นวิศวกรฝังตัว (FDE)

Forward-Deployed Engineer ที่เข้าไปนั่งกับทีมซัพพลายเชน เข้าใจงานจริง แล้วปรับ Agent ให้เข้ากับกระบวนการขององค์กรนั้น ไม่ใช่ยึดกระบวนการมาตรฐานให้องค์กรปรับตาม

โครงสร้างและจังหวะการทำงาน

CoE ทำงานเป็นคู่สองฝั่งเสมอ — ฝั่ง **Bricks (AI Builder)** ที่สร้างและปรับ Agent กับ ฝั่ง **องค์กร (เจ้าของกระบวนการจริง)** ที่รู้ว่างานควรจบอย่างไร ไม่มีฝั่งไหนทำสำเร็จได้ลำพัง

จังหวะการทำงาน (cadence) ที่แนะนำ

รายวัน ทบทวนเคสที่ Agent ส่งมาให้คนตัดสินใจ ว่าเส้นวงเงินตั้งไว้พอดีหรือไม่ · **รายสัปดาห์** ทบทวนอัตราการปิดเคสเอง และควรปรับ playbook ตรงไหน · **รายเดือน** ทบทวนเคสซ้ำเดิมจากคลังความรู้ ตัดสินใจเชิงโครงสร้าง เช่น จัดการซัพพลายเออร์ที่มีปัญหาซ้ำอย่างไร

มุมเฉพาะของซัพพลายเชน: ต้องมี Data / IT / Security ตั้งแต่วันแรก

ทุกฟังก์ชันต้องการ CoE แต่ซัพพลายเชนคือ ฟังก์ชันที่ต้องเชื่อมระบบมากที่สุดในทั้งสี่ฟังก์ชัน — เคสเดียวในบทที่ 3 แต่ละทั้ง ERP, WMS และอีเมลซัพพลายเออร์ และในองค์กรจริงมักมีมากกว่านั้น: ระบบขนส่ง (TMS) พอร์ทัลซัพพลายเออร์ ระบบการเงิน สเปรดชีตที่ทีมทำเอง ทุกการเชื่อมต่อคือทั้งประตูสู่ข้อมูลและทั้งความเสี่ยงด้านความปลอดภัย

ข้อสรุปเชิงปฏิบัติจริงชัด — ทีม Data, IT และ Security ต้องเป็นคู่งานตั้งแต่วันแรก ไม่ใช่มาเรียกตอนจะขึ้นระบบ การดึงสามทีมนี้เข้ามาแต่ต้นไม่ใช่พิธีการ แต่คือสิ่งที่กำหนดว่าโครงการจะเดินต่อได้หรือไม่

บทที่ 05

บทเรียน 4 ข้อ

รูปแบบความล้มเหลวของโครงการ AI ในซัพพลายเชนค่อนข้างซ้ำกัน และเกือบทั้งหมดไม่ใช่ปัญหาทางเทคนิค

01 ลงทุนกับ forecast แต่ไม่ลงทุนกับ exception

ทีมส่วนใหญ่ทุ่มงบทำให้ forecast แม่นขึ้นอีกไม่กี่เปอร์เซ็นต์ ทั้งที่ความเจ็บปวดจริงไม่ได้อยู่ที่ forecast ผิด แต่อยู่ที่ **ไม่มีใครจัดการเคสที่หลุดแผนจนจบ** forecast จะแม่นยำแค่ไหน ของก็ยังไม่มาตามกำหนด เพราะความไม่แน่นอนอยู่ที่ซัพพลายเออร์และการขนส่ง จุดที่คืนเวลาให้ทีมได้มากที่สุดคือ exception handling — เริ่มลงทุนตรงที่เจ็บจริง

02 ไม่ตั้งเส้นวงเงินให้ Agent ตั้งแต่แรก สุดท้ายต้องอนุมัติทุกเคสเหมือนเดิม

หลายองค์กรเปิด Agent ให้เสนองานแก่ได้ แต่ไม่กล้าตั้งเส้นให้ Agent ลงมือเอง ผลคือทุกเคส — แม้แต่เคสเล็กที่ไม่มีต้นทุนเพิ่ม — ก็ยังต้องรอคนกดอนุมัติ เก้ากับได้ประโยชน์แค่ครั้งเดียว **หัวใจไม่ได้อยู่ที่ให้ Agent ตัดสินทุกอย่าง แต่อยู่ที่กล้าตั้งเส้นให้ชัดเจนว่าเคสแบบไหน Agent ปิดเองได้** เส้นที่ตั้งต่ำเกินไปตอนแรกก็ยิ่งดีกว่าไม่มีเส้น เพราะปรับขึ้นได้เมื่อเห็นว่าระบบน่าเชื่อถือ

03 ไม่บันทึกสาเหตุตอนปิดเคส เคสเดิมจึงกลับมาทุกเดือน

ขั้นที่คนมองข้ามที่สุดคือขั้นสุดท้าย — การบันทึกสาเหตุตอนปิดเคส ถ้าปิดแล้วจบไปเฉย ๆ องค์กรก็ไม่มีทางรู้ว่าเคสประเภทเดียวกันเกิดซ้ำแค่ไหน ซัพพลายเออร์ที่ส่งช้าประจำจะยังส่งช้าต่อไป **การปิดเคสที่ดีไม่ได้จบที่การแก้เฉพาะหน้า แต่จบที่การทำให้ปัญหานั้นมีโอกาสไม่กลับมาอีก** ซึ่งเกิดได้ก็ต่อเมื่อสาเหตุถูกบันทึกอย่างเป็นระบบ

04 ปลอ่ยให้การใช้งานจริง (adoption) เป็นผลพลอยได้

บทเรียนสุดท้ายเป็นเรื่องคน ไม่ใช่ระบบ องค์กรจำนวนมากสร้าง Agent ที่ทำงานได้ดีในการทดสอบ แต่ทีมจริงไม่ใช้ เพราะไม่เชื่อใจหรือไม่เข้าใจว่าจะเข้ามาอยู่ในงานประจำวันตรงไหน การใช้งานจริงต้องถูกออกแบบ — ให้ทีมเห็นเคสแรก ๆ ที่ช่วยได้จริง ให้เขามีส่วนตั้งเส้นวงเงิน **ถ้าวัดแต่ว่าระบบทำงานได้ โดยไม่วัดว่าทีมใช้จริงแค่ไหน โครงการจะสำเร็จบนกระดาษแต่ไม่เปลี่ยนอะไรในทางปฏิบัติ**

บทที่ 06

Business Takeaways

และจุดเริ่มต้น 90 วัน

หน้านี้สรุปสาระของทั้งเล่มไว้ในหน้าเดียว เพื่อนำไปใช้ประกอบการนำเสนอและการตัดสินใจภายในองค์กรได้ทันที

4 ตัวเลขที่ควรวัด — และตัวชี้ขาด

ตัวชี้วัด	วัดอะไร และทำไมถึงสำคัญ
อัตรา exception ที่ปิดได้เองโดยไม่มีคนแตะ ← ตัวชี้ขาด	ในบรรดาเคสที่หลุดแผนทั้งหมด ที่เปอร์เซ็นต์ที่ระบบจัดการจนจบเองได้ — สะท้อนคุณค่าจริงของทั้งเล่ม ไม่ใช่ความแม่นยำของ forecast
เวลาเฉลี่ยตั้งแต่ตรวจพบจนปิดเคส	เคสที่ต้องถึงคน ใช้เวลาสั้นลงแค่ไหนเมื่อ Agent เตรียมข้อมูลมาให้ครบ
อัตราเคสซ้ำเดิม	เคสประเภทเดียวกันกลับมาบ่อยแค่ไหน สะท้อนว่าการบันทึกกันซ้ำได้ผลจริงหรือไม่
การใช้งานจริง (adoption)	ทีมใช้ระบบในงานประจำวันมากแค่ไหน ไม่ใช่แค่ตอนทดสอบ

ทุกตัวเลขในหน้านี้เป็นกรอบการวัด ไม่ใช่ผลที่รับประกัน — ค่าจริงขึ้นกับแต่ละองค์กร

จุดเริ่มใน 90 วันแรก — เริ่มที่ 3-way match (งานข้อ 9)

อย่าเริ่มที่ exception handling ทั้งวงจรร เพราะมันคือส่วนที่ยากที่สุด เริ่มที่ 3-way match เพราะเป็นงานที่ **ปริมาณสูงที่สุด** (เกิดทุกใบสั่งซื้อ เห็นผลเร็ว) · **กฎขัดที่สุด** (จับคู่เอกสารสามใบให้ตรง วัดได้ตรง ๆ) · **ความเสี่ยงต่ำที่สุด** (แก้ได้ก่อนจ่ายเงิน ไม่กระทบลูกค้าโดยตรง)

เมื่อ 3-way match เดินได้บ้างและทีมเริ่มเชื่อใจ จึงค่อยขยับไปตรวจจับและเปิดเคสอัตโนมัติ (ข้อ 12) แล้วจึงไต่ขึ้นสู่การใส่ต้นตอและปิดเคสทั้งวงจรร — สร้างความเชื่อใจไปพร้อมกับสร้างผล ไม่ใช่เติมพินทั้งหมดในก้าวแรก

5 คำถามก่อนเริ่ม

- เราวัด “อัตรา exception ที่ปิดได้เอง” อยู่หรือยัง — หรือยังวัดแต่ความแม่นยำของ forecast
- เส้นวงเงินอนุมัติ ของเราชัดหรือยัง — Agent สั่งซื้อทดแทนเองได้ถึงเท่าไร เกินจากนั้นถึงใคร
- Data / IT / Security อยู่ในทีมตั้งแต่วันแรกหรือไม่ — ชีพพลายเซ็นเชื่อมระบบมากที่สุด จุดนี้ตัดสินใจว่าโครงการเดินต่อได้หรือไม่
- เรา บันทึกสาเหตุตอนปิดเคส อย่างเป็นระบบหรือยัง — หรือปิดแล้วจบไป
- ทีมจริงมีส่วนออกแบบ หรือไม่ — หรือระบบถูกสร้างมาให้ใช้ ถ้าไม่มีส่วนตั้งแต่ต้น การใช้งานจริงจะไม่เกิด

90 วันในนี้คือกรอบเวลาแนะนำสำหรับการทยอยเริ่มด้วยตัวองค์กรเอง ไม่ใช่ข้อผูกมัดบริการใด ๆ

ซัพพลายเชนไม่ได้ฟังที่แผน แต่ฟังที่ข้อยกเว้น

เล่มนี้ไม่ได้เสนอให้ AI มาวางแผนแทนคุณ — เครื่องมือวางแผนที่ดีมีอยู่แล้ว เล่มนี้เสนอให้ AI Agent รับงานที่กินเวลายาวนานที่สุดแต่สร้างคุณค่าน้อยที่สุดไป นั่นคืองานไล่ตามสิ่งที่หลุดแผน เพื่อให้คนของคุณได้กลับไปทำงานที่ต้องใช้คนจริง ๆ

Task Ownership Map ไม่ได้บอกว่า AI เก่งแค่ไหน มันบอกว่างานไหนควรเป็นของใคร — และในซัพพลายเชน คำตอบนั้นชัดเจนกว่าที่อื่น เพราะงานส่วนใหญ่มีกฎชัดเจนและวัดผลได้ นั่นคือเหตุผลที่เล่มนี้มีสัดส่วน Agent-owned สูงที่สุดในซีรีส์

อยากคุยเรื่องการเริ่ม AI Center of Excellence ในองค์กรของคุณ?

ทีม AI Builder ของบริคส์ยินดีคุยเบื้องต้นเพื่อประเมินความพร้อมร่วมกัน โดยยังไม่มีข้อผูกมัด

ลงทะเบียนคุยกับทีมได้ที่

[ลิงก์ฟอร์ม]

01 · Human Resource

HR ที่ไม่ต้องเพิ่มคน — รู้ก่อนว่างานไหนเป็นของใคร

02 · Marketing

Content & Campaign Ops ที่ Agent ปิดงานเองได้

04 · Procurement & TOR

ตรวจ TOR ด้วย Agent ที่ตรวจย้อนหลังได้