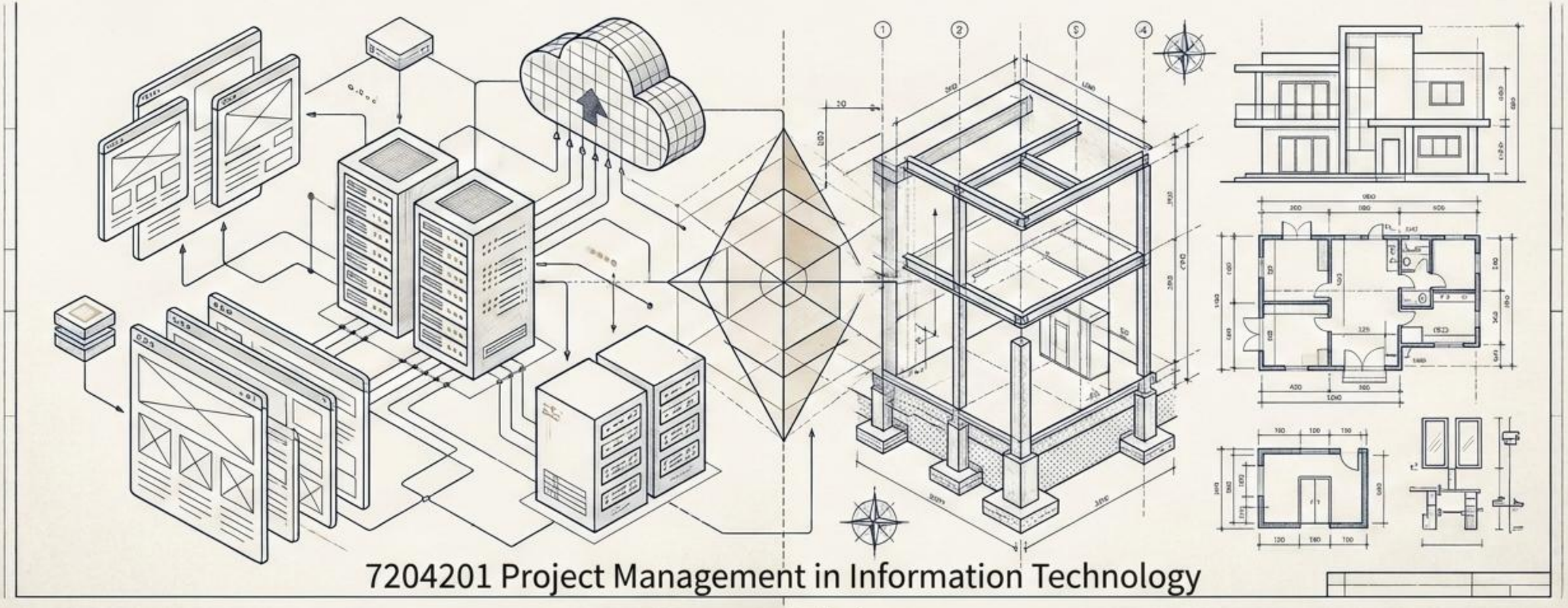
การเขียนโครงการและการบริหารโครงการ IT

สร้างรากฐานสถาปัตยกรรมซอฟต์แวร์ด้วยการวางแผนและการควบคุมอย่างเป็นระบบ

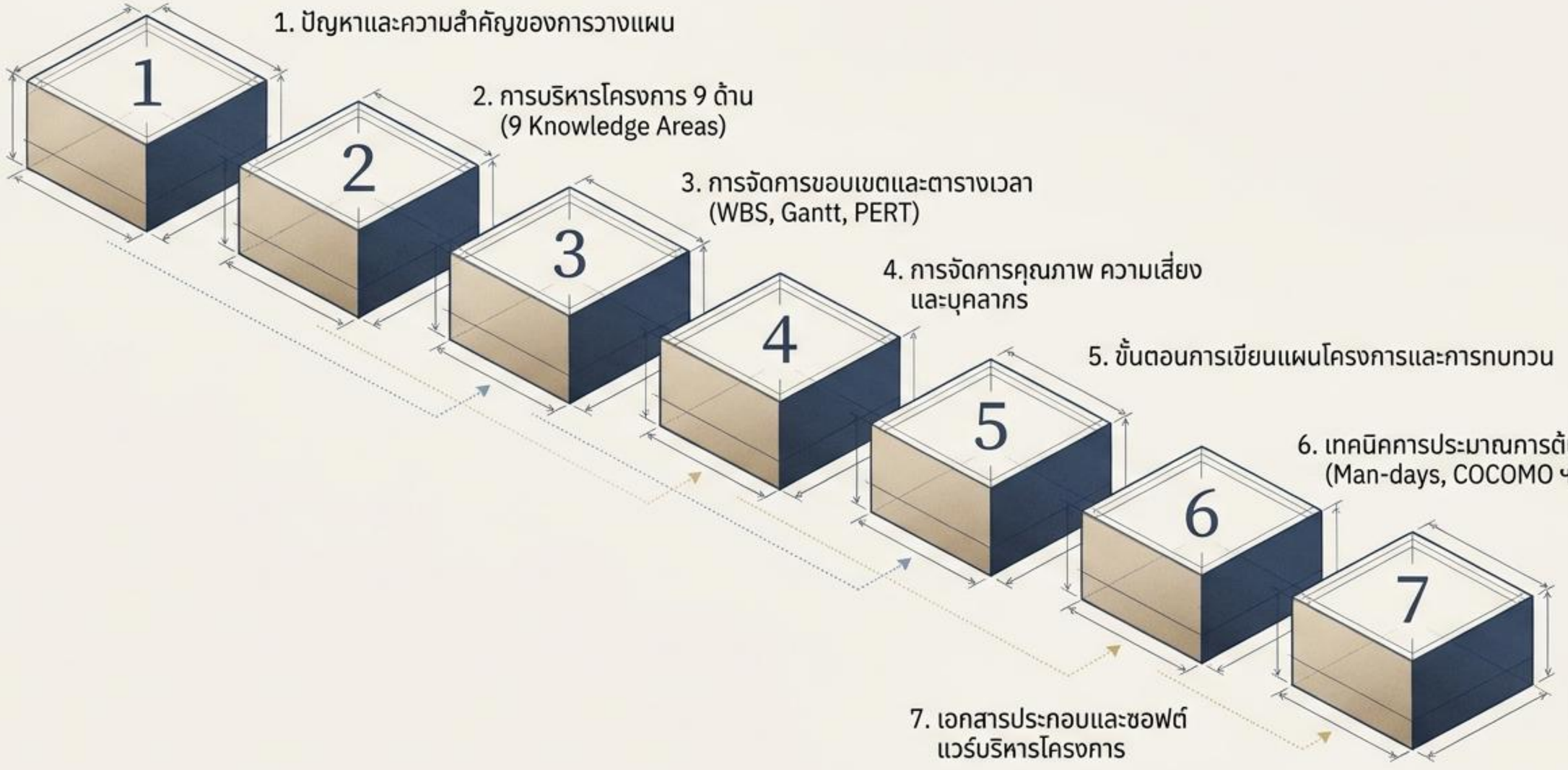


7204201 Project Management in Information Technology

Paranya Palwisut

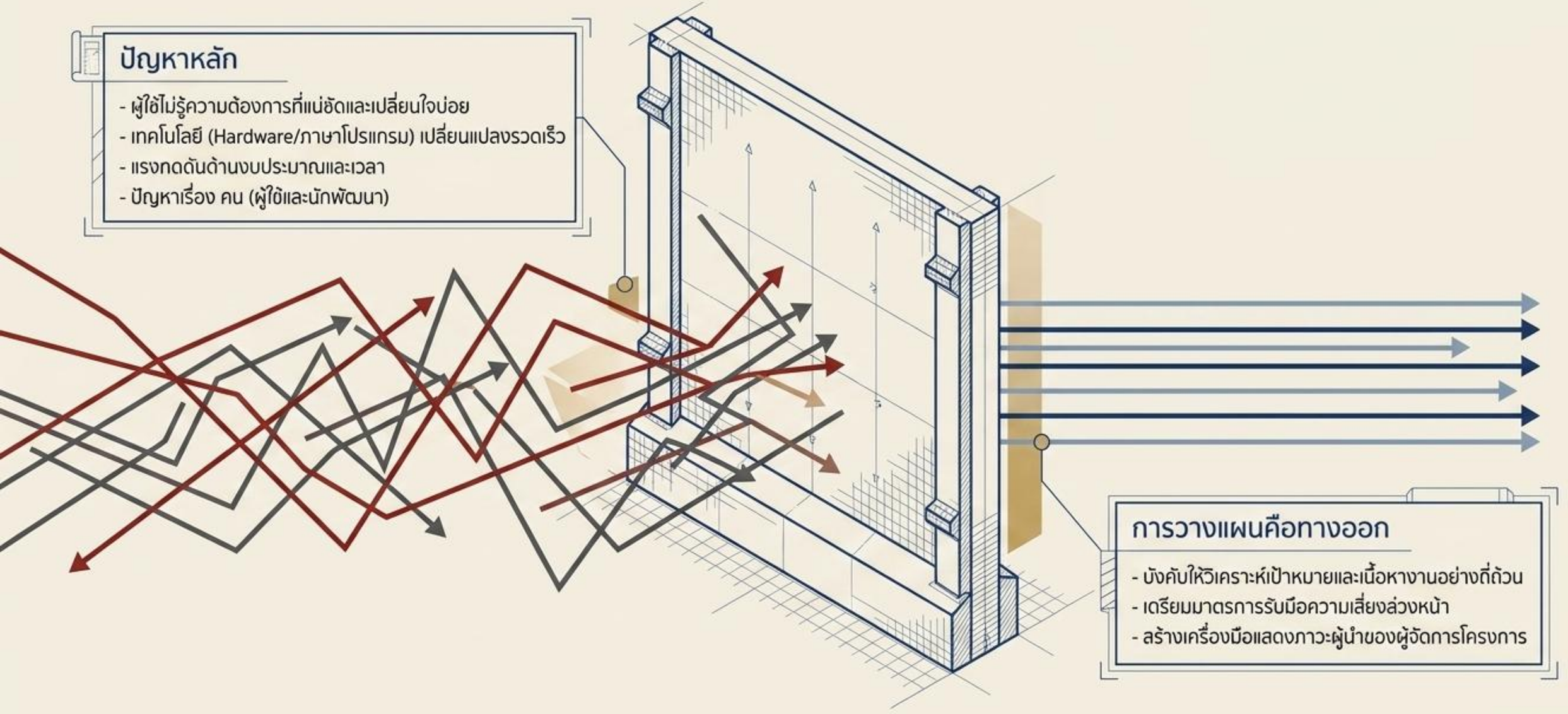
NotebookLM

สรุปหัวข้อนำเสนอ (Agenda)



NotebookLM

วิกฤตการณ์งาน IT และหัวใจของการวางแผน



ปัญหาหลัก

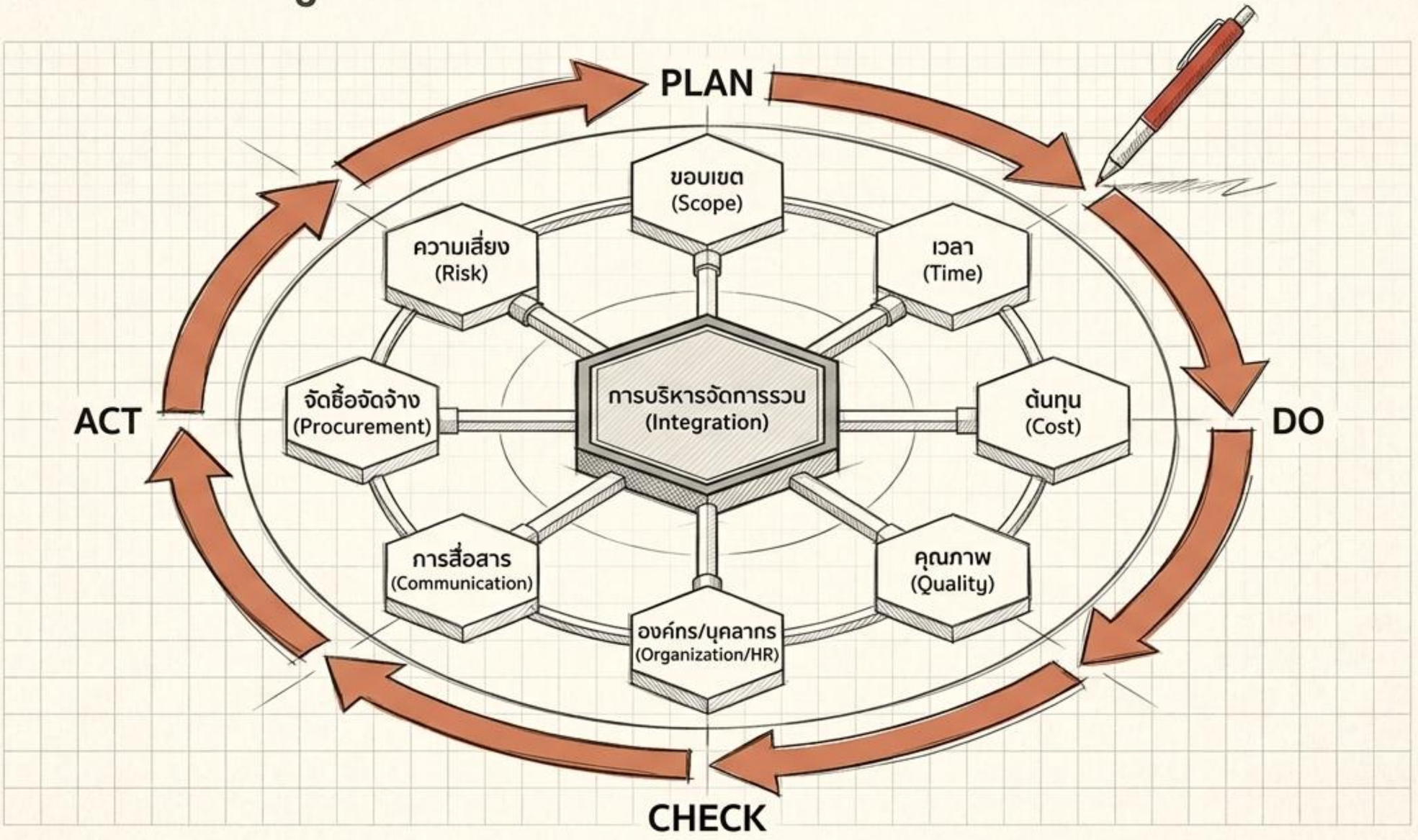
- ผู้ใช้ไม่รู้ความต้องการที่แน่ชัดและเปลี่ยนใจบ่อย
- เทคโนโลยี (Hardware/ภาษาโปรแกรม) เปลี่ยนแปลงรวดเร็ว
- แรงกดดันด้านงบประมาณและเวลา
- ปัญหาเรื่อง คน (ผู้ใช้และนักพัฒนา)

การวางแผนคือทางออก

- บังคับให้วิเคราะห์เป้าหมายและเนื้อหางานอย่างถี่ถ้วน
- เตรียมมาตรการรับมือความเสี่ยงล่วงหน้า
- สร้างเครื่องมือแสดงภาวะผู้นำของผู้จัดการโครงการ

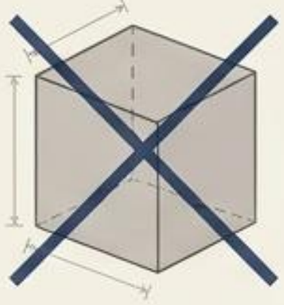
NotebookLM

เข็มทิศผู้จัดการโครงการ: การบริหาร 9 ด้าน

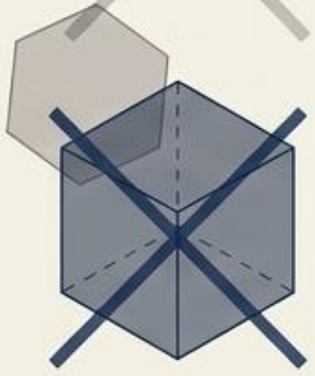


NotebookLM

การจัดการขอบเขต (Scope Management)



Unapproved Features



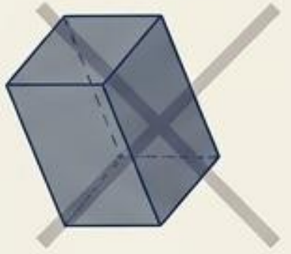
In-Scope

การกำหนดขอบเขต (Scope Planning):
จุดเริ่มต้นที่มาจากความต้องการและความคาดหวังของลูกค้า (สิ่งที่จับต้องไม่ได้ สู่สิ่งที่วัดผลได้)

การบริหารผลงานส่งมอบ (Deliverables):
กำหนดชัดเจนตั้งแต่งานระหว่างกาล จนถึงผลงานขั้นสุดท้าย (เช่น โปรแกรม, เอกสารออกแบบ, คู่มือผู้ใช้)

การบริหารการเปลี่ยนแปลง (Change Management):
กระบวนการรักษาคุณภาพร่วมกับลูกค้าเมื่อขอบเขตต้องเปลี่ยน

Out-of-Scope



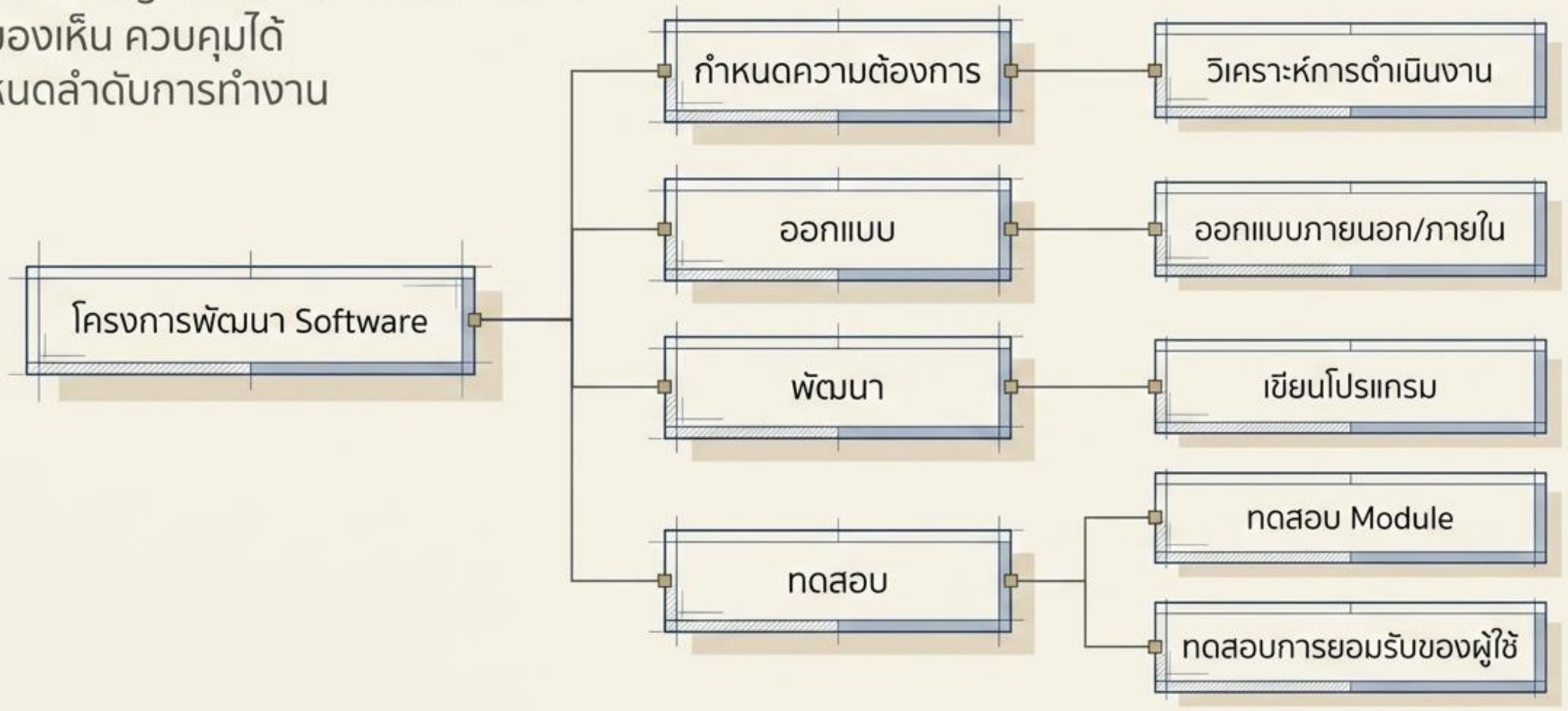
Excessive Documentation

Non-essential Tasks

NotebookLM

สถาปัตยกรรมโครงสร้างงาน: WBS (Work Breakdown Structure)

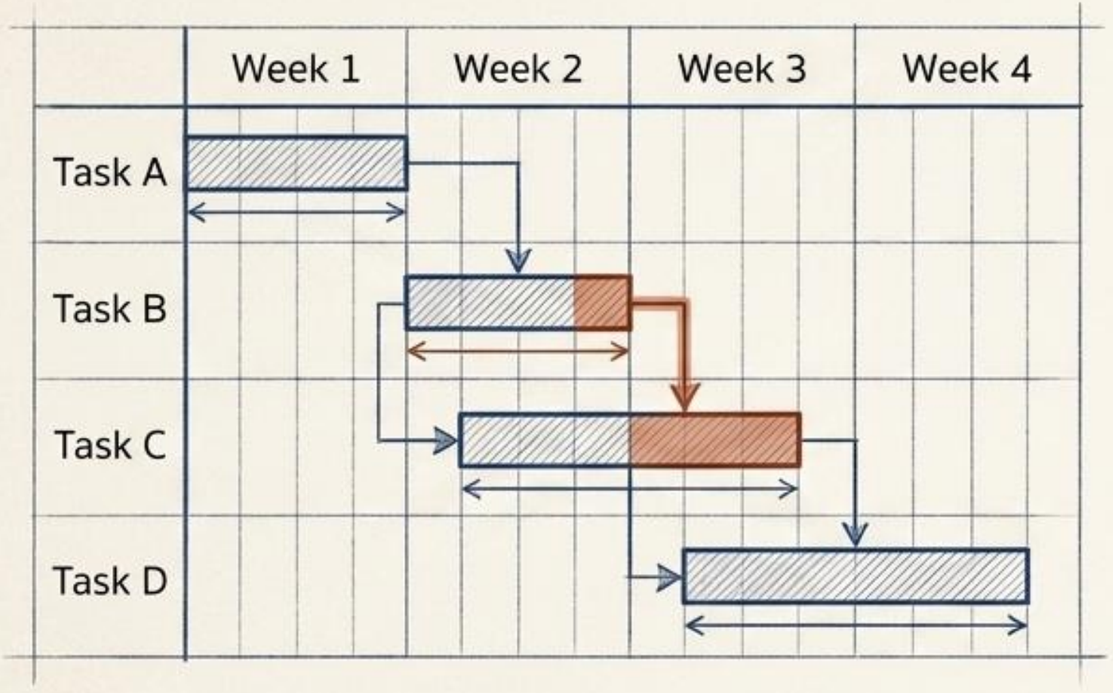
การแตกงานใหญ่ที่ซับซ้อนออกเป็นงานย่อย
เพื่อให้มองเห็น ควบคุมได้
และกำหนดลำดับการทำงาน



NotebookLM

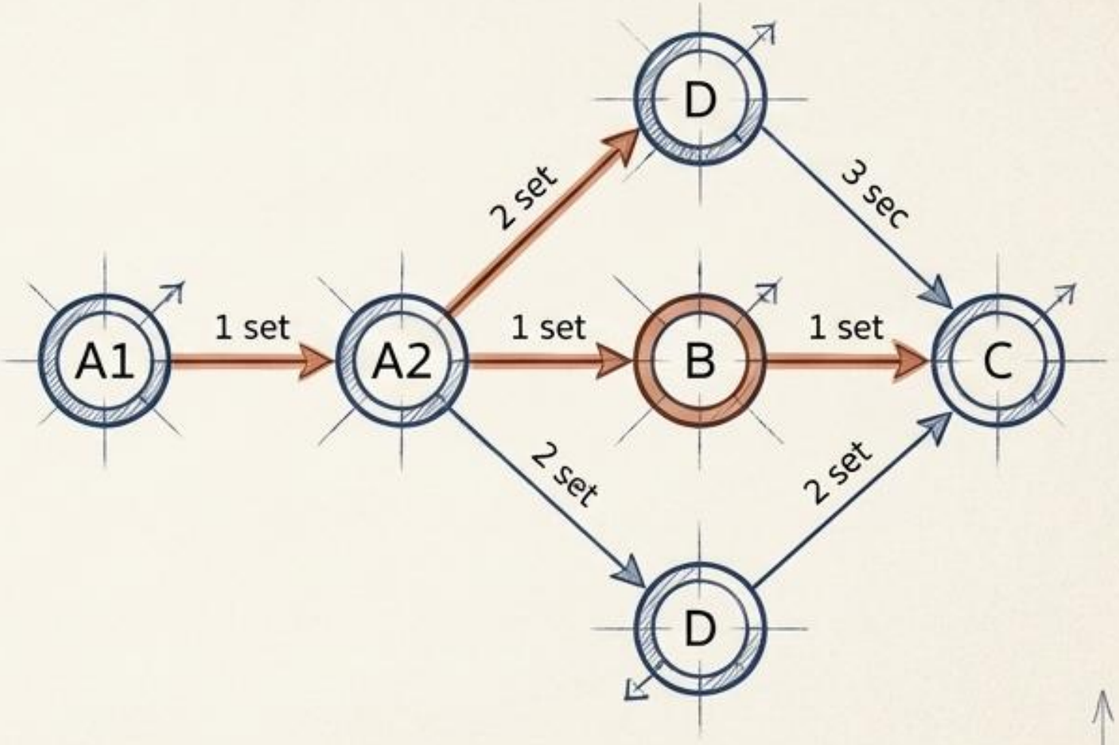
เครื่องมือควบคุมเวลา: Gantt Chart vs. PERT Diagram

Gantt Chart / Bar Chart



ภาพรวมตามปฏิทินเวลา: แสดงระยะเวลาเริ่มต้นและสิ้นสุดของแต่ละงานชัดเจน เหมาะสำหรับดูภาพรวมความคืบหน้า

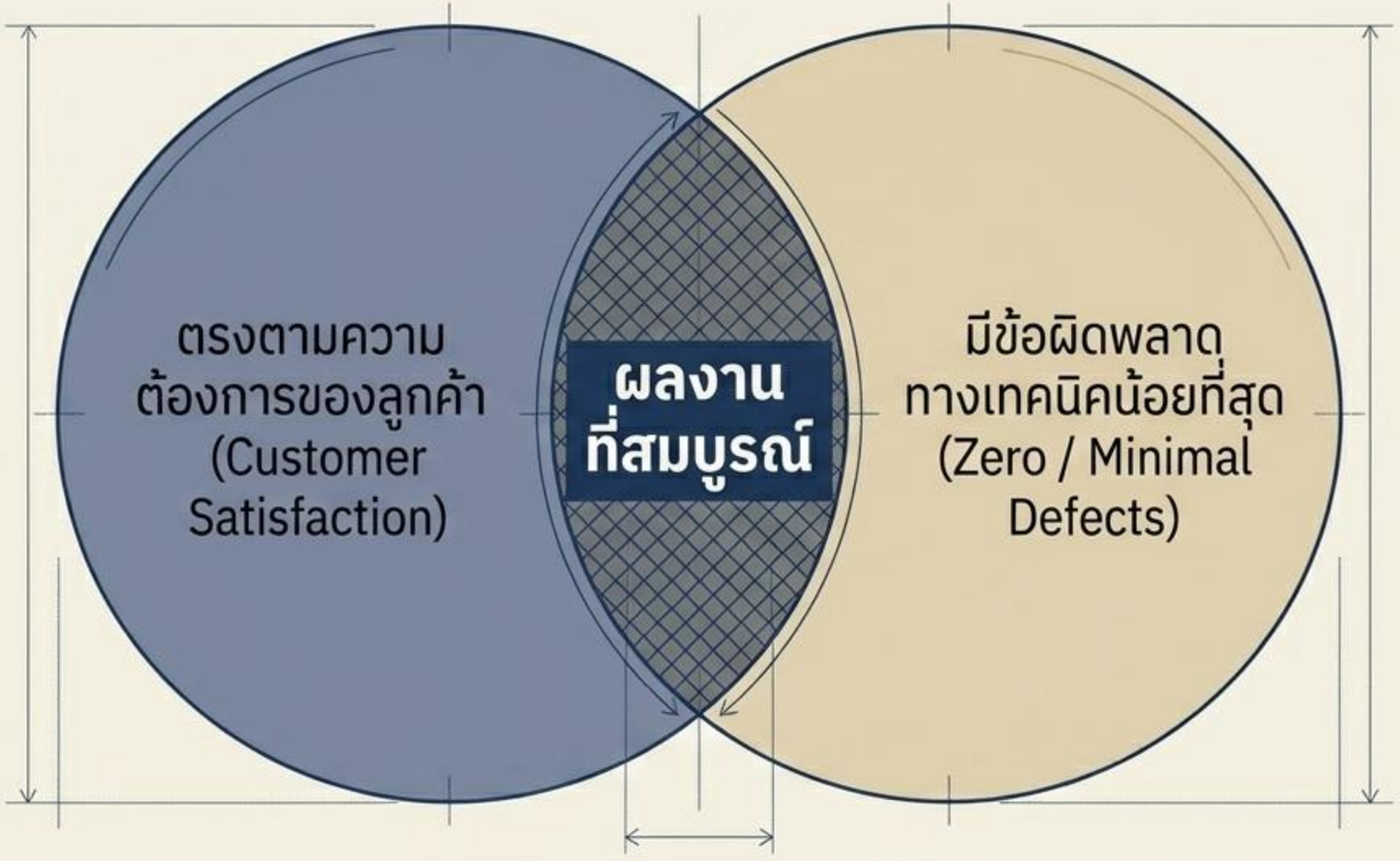
PERT Diagram (Network Diagram)



ความสัมพันธ์และลำดับงาน: แสดงจุดวิกฤต (Critical Path) และงานใดต้องรองานไหนเสร็จก่อน (เช่น A1 -> A2 -> B)

NotebookLM

การจัดการคุณภาพ (Quality Management)

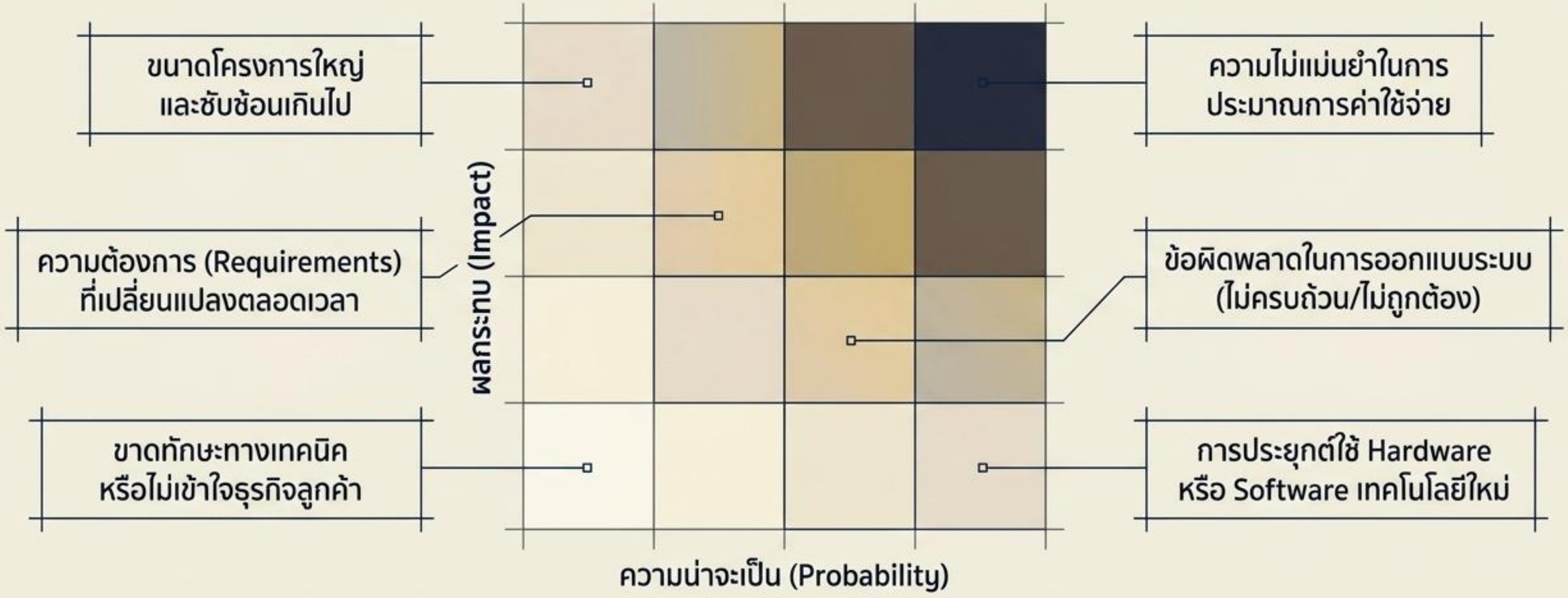


การวางแผนคุณภาพ (Quality Planning) เป็นรากฐานที่สำคัญกว่าการประกันคุณภาพ (Quality Assurance) ที่ปลายทาง

NotebookLM

การจัดการความเสี่ยง (Risk Management)

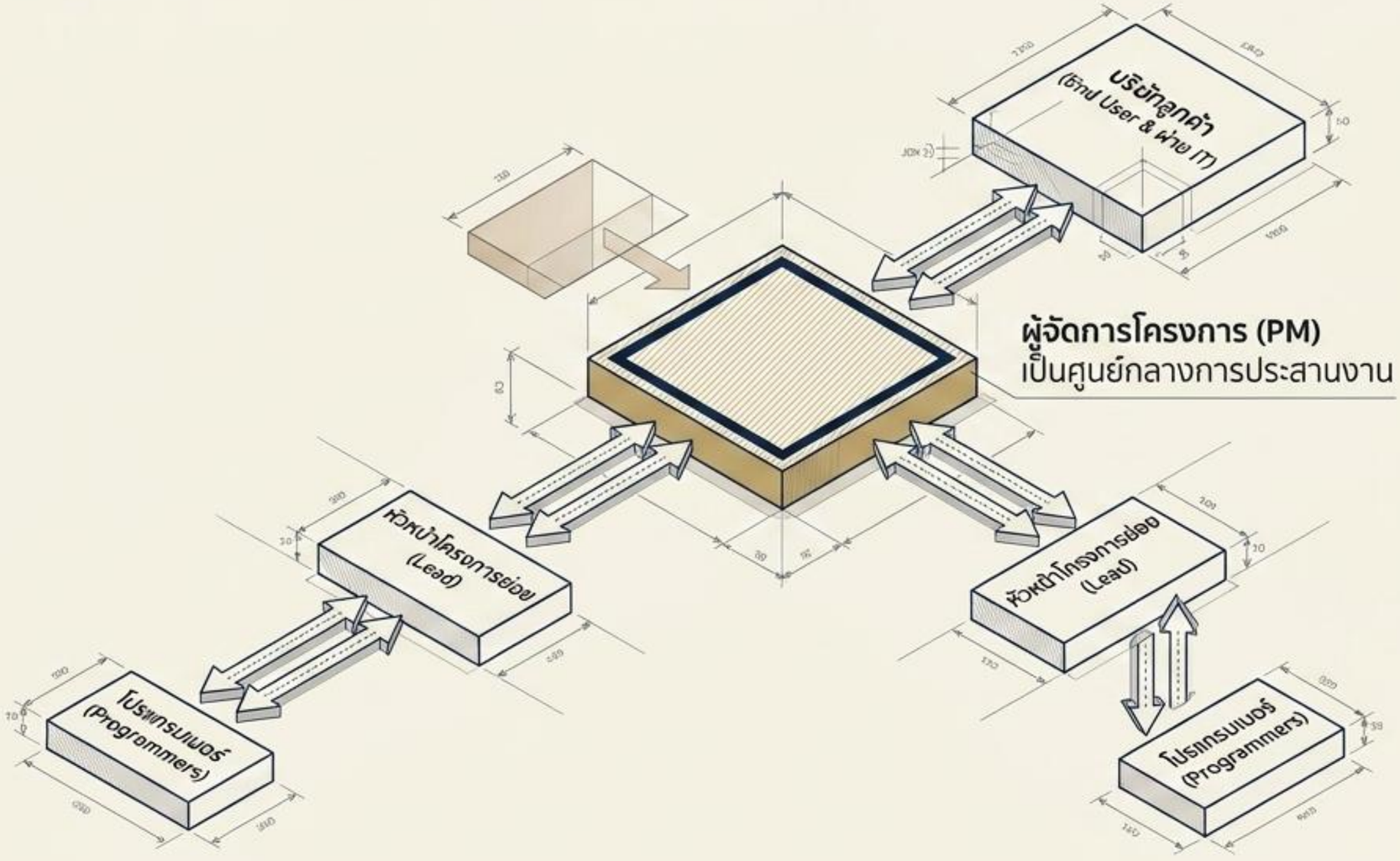
ปัจจัยที่อาจทำให้โครงการล้มเหลวและต้องเตรียมมาตรการลดความเสี่ยง



NotebookLM

สถาปัตยกรรมบุคลากรและการสื่อสาร (HR & Communication)

การบริหารบุคลากรเพื่อป้องกันปัญหา คือสิ่งสำคัญที่สุด เริ่มจากการเตรียมทีมงานให้พอดีกับขนาดโครงการ และวางแผนการสื่อสารที่รัดกุม



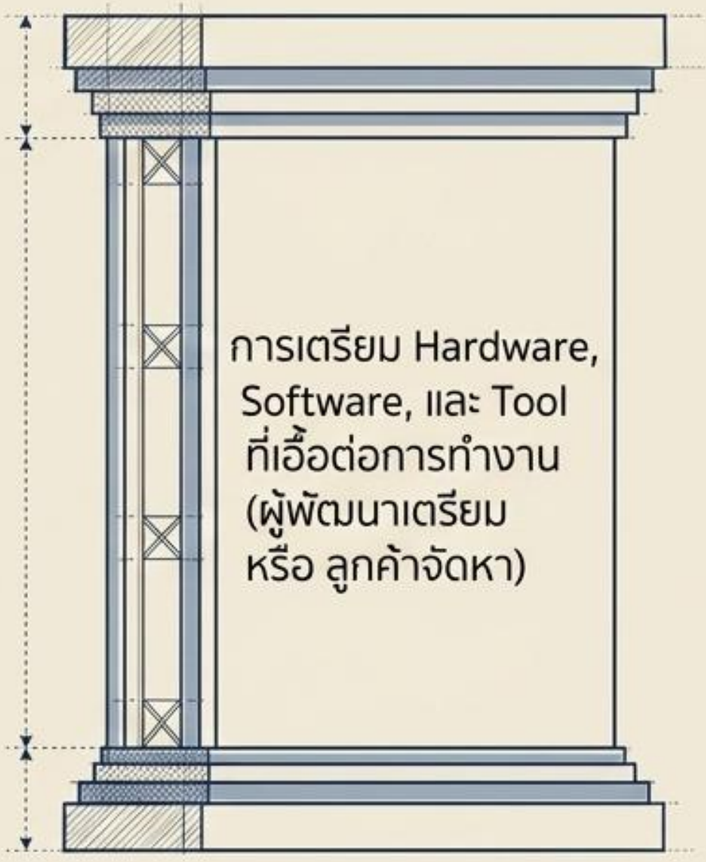
NotebookLM

การจัดซื้อจัดจ้าง โครงสร้างต้นทุน และสภาพแวดล้อม

การจัดการ Sub-contract (Procurement)

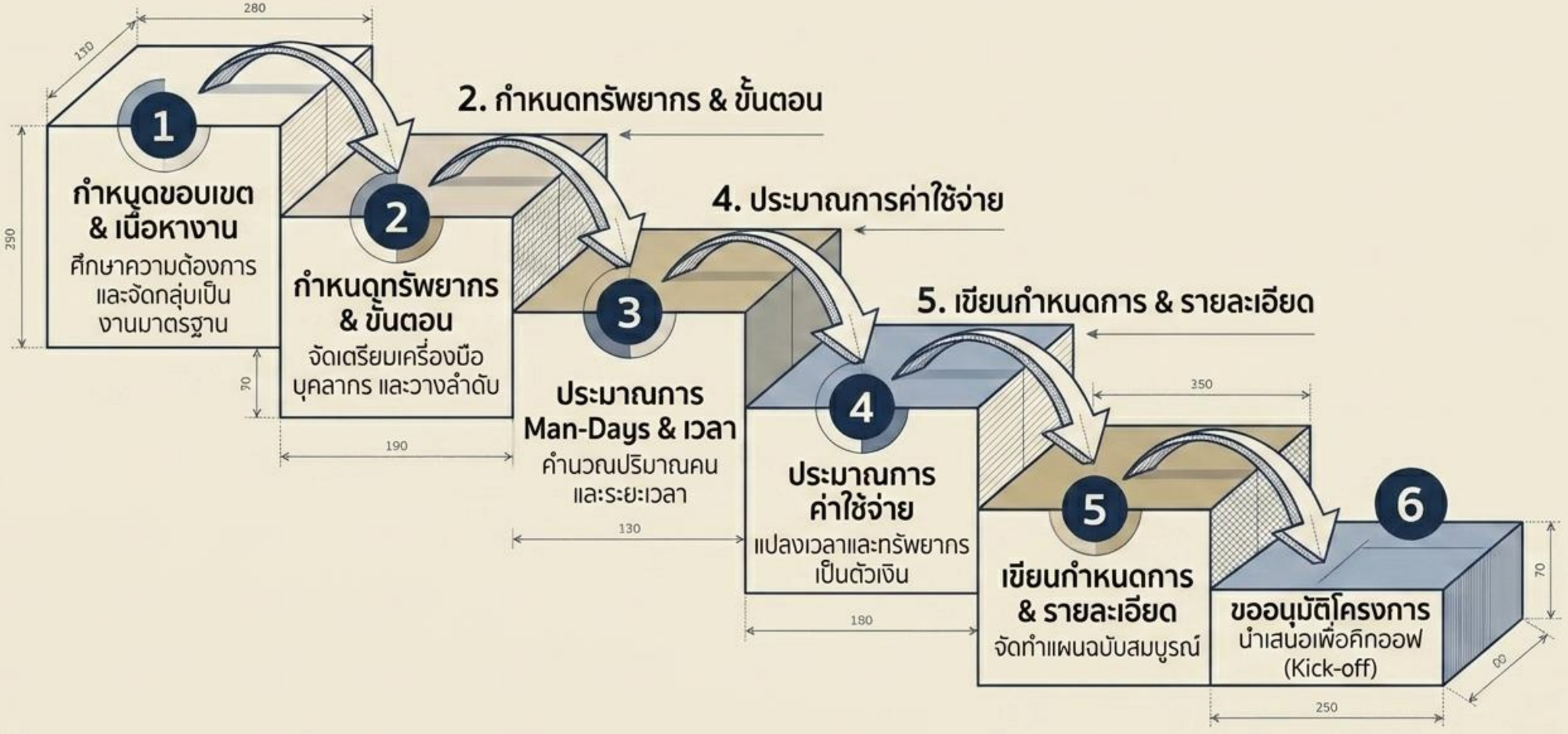
โครงสร้างต้นทุน (Cost Structure)

สภาพแวดล้อมการพัฒนา (Environment)



NotebookLM

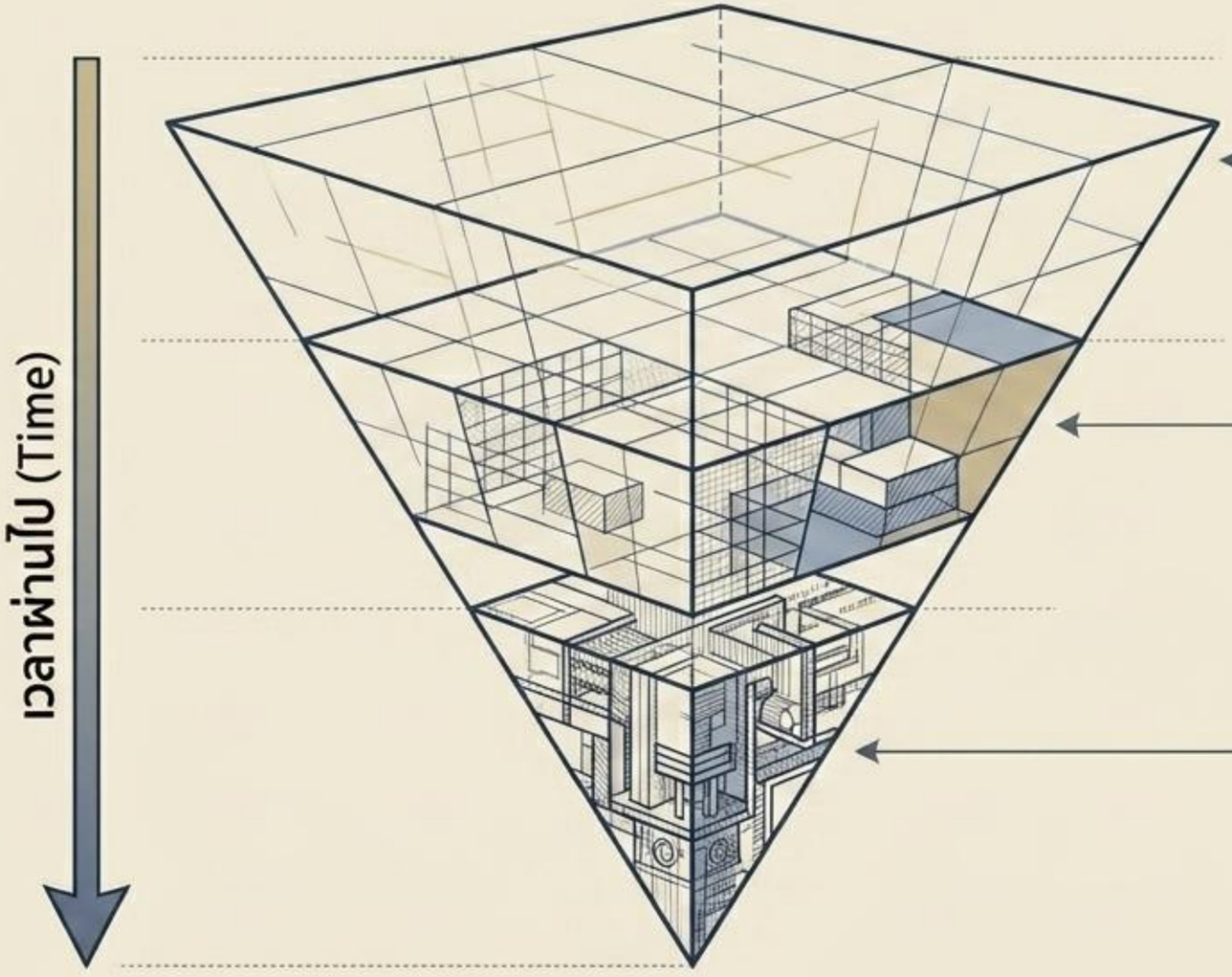
ขั้นตอนมาตรฐานในการเขียนแผนโครงการ



NotebookLM

การทบทวนและเพิ่มรายละเอียด (Progressive Elaboration)

แผนโครงการไม่ได้เขียนเสร็จในครั้งเดียว แต่จะมีความละเอียดเพิ่มขึ้นเมื่อเวลาผ่านไป



Phase 1: Master Plan (ช่วงเริ่มต้น)
- แผนคร่าวๆ เพื่อขออนุมัติโครงการ

Phase 2: Action Plan (หลังอนุมัติ)
- ผู้จัดการโครงการลงรายละเอียดเชิงลึกเพื่อรีดศักยภาพทีม

Phase 3: Refinement (หลังออกแบบ)
- ทบทวนและปรับปรุงแผนอีกครั้งเมื่อได้สถาปัตยกรรมระบบที่ชัดเจน

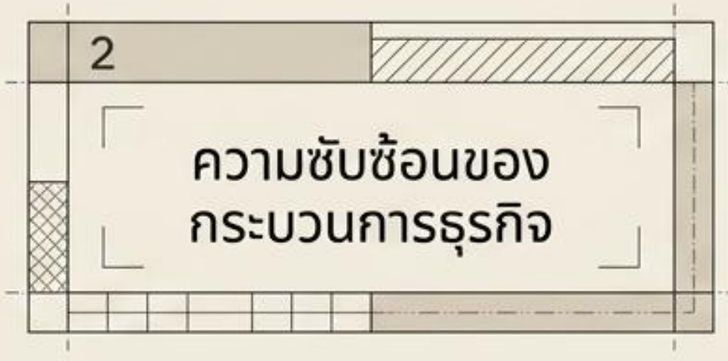
NotebookLM

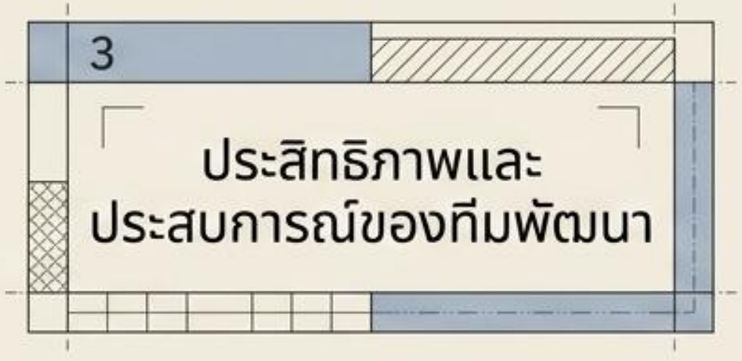
ปฐมบทการประมาณการต้นทุน: แนวคิด Man-Days

$$[\text{ขนาดของ Software}] \times \frac{[\text{ความซับซ้อน}]}{[\text{ทักษะทีม}]} = [\text{จำนวน Man-Days}]$$

Man-days (หรือ Manday) คือการประมาณ ‘จำนวนคน’ และ ‘เวลา’ ที่ต้องใช้ในการสร้างซอฟต์แวร์ ต้องอาศัยทั้งศาสตร์ (ข้อมูล) และศิลป์ (ประสบการณ์)

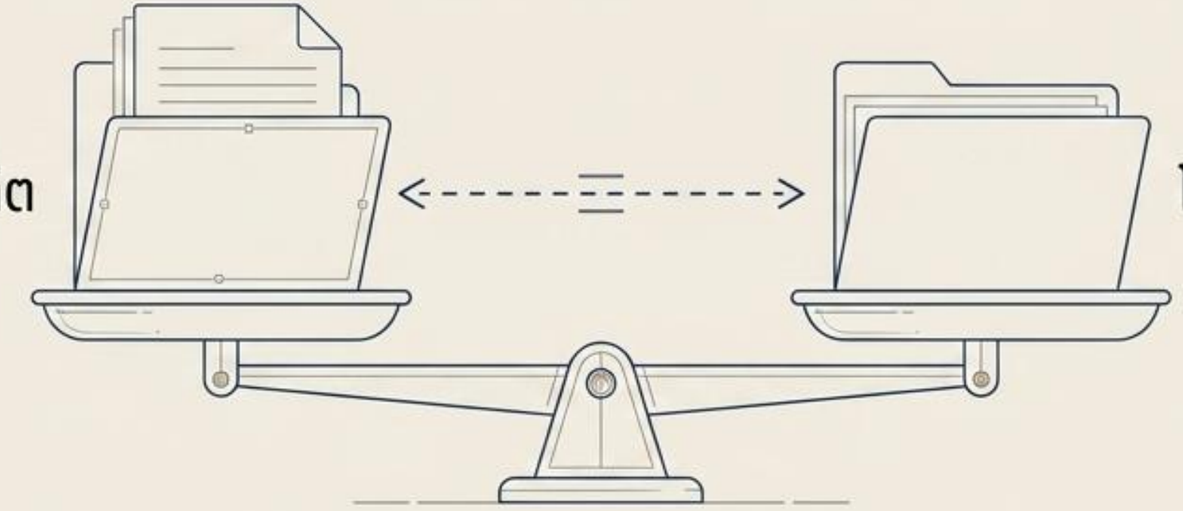
ตัวแปรที่มีผลกระทบ





NotebookLM

เทคนิคที่ 1: การประมาณการคร่าวๆ (Comparative Estimation)



หลักการทำงาน

เทียบเคียงกับโครงการในอดีตที่มีขอบเขต
เทียบเคียงกับโครงการในอดีตที่มีขอบเขต
ฟังก์ชัน และภาษาโปรแกรมคล้ายคลึงกัน

จุดแข็ง

ต้นทุนในการประเมินต่ำที่สุดและรวดเร็ว

ข้อกำหนด

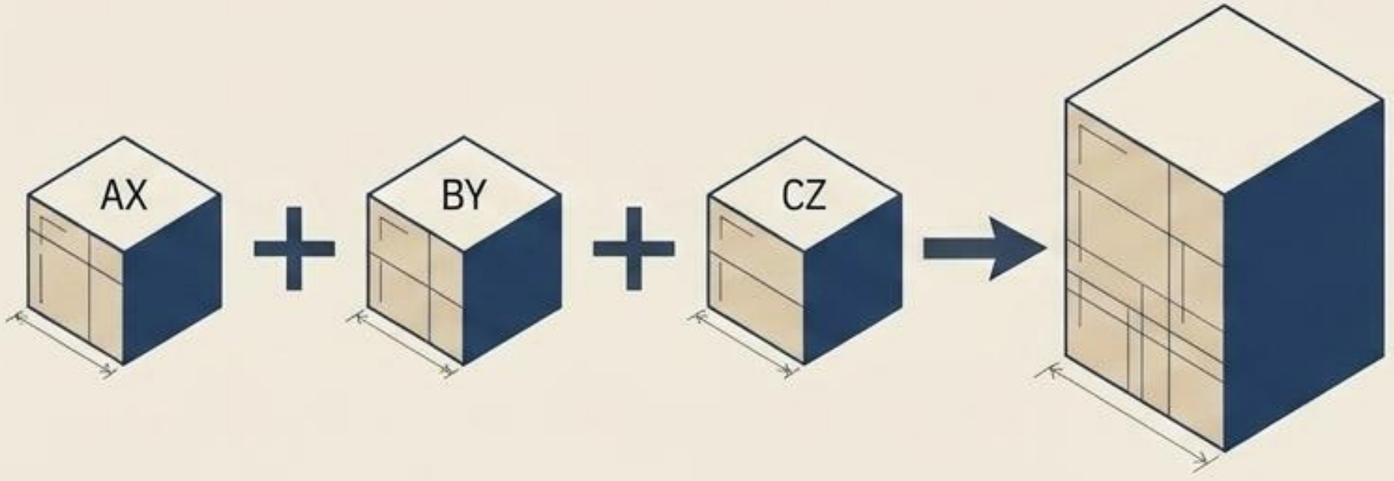
ผู้ประเมินต้องเคยร่วมงานในอดีต และมีการ
จัดเก็บข้อมูลโครงการเก่าอย่างเป็นระบบ

ข้อควรระวัง

ความแม่นยำต่ำ เหมาะสำหรับใช้เป็น
“การประมาณการเบื้องต้น” เท่านั้น

เทคนิคที่ 2: การรวม Man-Day ของงานย่อย (WBS Estimation)

หลักการ: แบ่งกระบวนการพัฒนาเป็นงานย่อยระดับล่างสุด
ประเมิน Man-day แต่ละจุด แล้วนำผลลัพธ์มารวมกัน



Summation Concept

Total Man-days

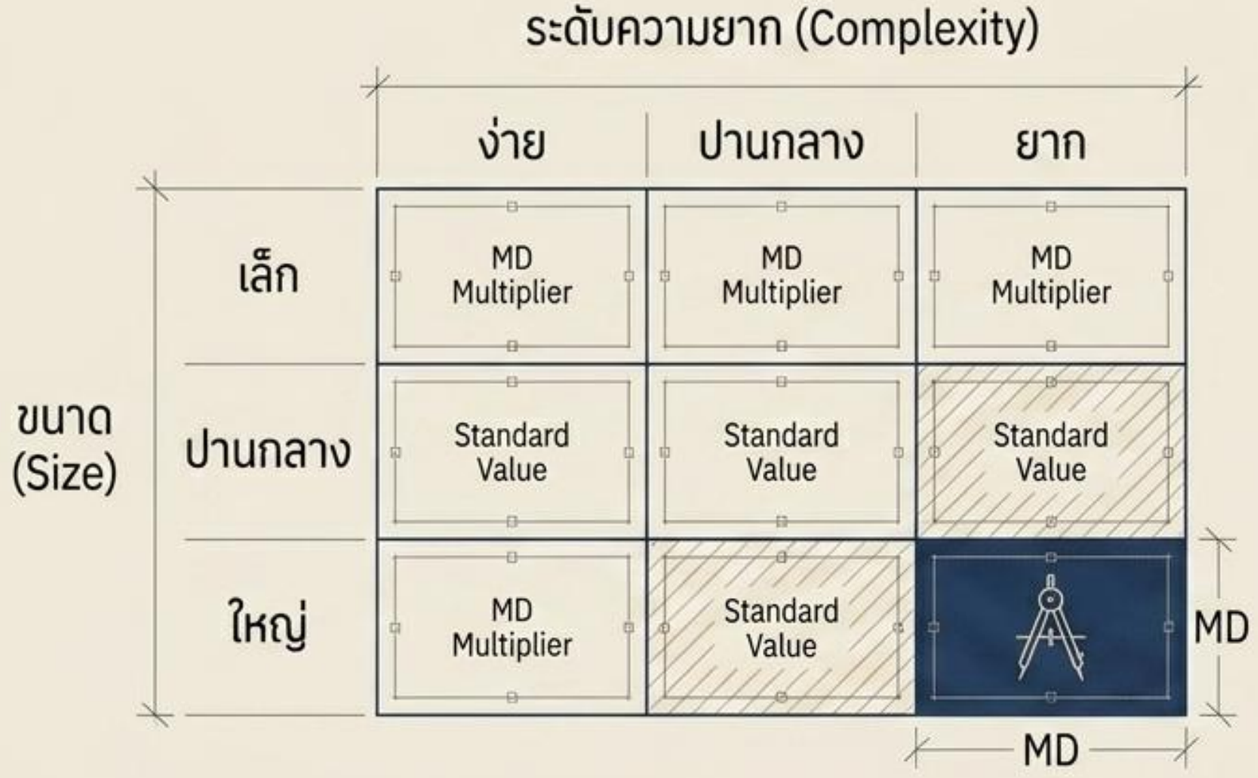
ออกแบบหน้าจอ Input (X หน้าจอ)	=	AX Man-days
ออกแบบหน้าจอ Output (Y หน้าจอ)	=	BY Man-days
Total	=	AX + BY + ...

ยิ่งแบ่งย่อยได้ละเอียด ความแม่นยำยิ่งสูง แต่แลกมาด้วยต้นทุนและเวลาในการประเมินที่สูงขึ้นตาม

NotebookLM

เทคนิคที่ 3: การประมาณการด้วยงานมาตรฐาน (Standard Tasks)

หลักการ: นำงานย่อยไปจับคู่กับ ตารางงานมาตรฐาน ที่กำหนดค่า Man-days ไว้ล่วงหน้าจากสถิติขององค์กร



1. ประเมินระดับความยาก (ง่าย / ปานกลาง / ยาก)
2. ประเมินขนาด(เล็ก / ปานกลาง / ใหญ่)
3. หาผลคูณ (เช่น ออกแบบหน้าจอขนาดใหญ่-ยาก = ใช้ Man-days สูงสุด)

ทำให้การคำนวณง่ายและเป็นมาตรฐานเดียวกันทั้งองค์กร

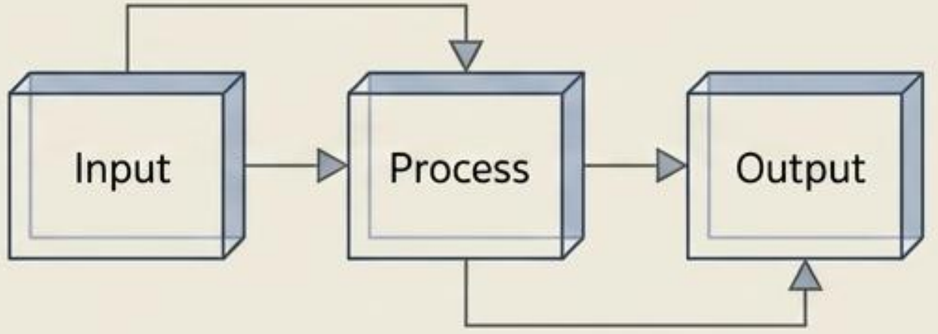
NotebookLM

เทคนิคที่ 4: เครื่องมือประเมินขั้นสูง (COCOMO & Function Point)
(COCOMO & Function Point)

COCOMO (Constructive Cost Model)

- คิดค้นโดย Dr. Barry Boehm (1981)
- ประเมินโดยนับจำนวนบรรทัดของ Source Code (LOC: Line of Code)
- เหมาะกับระบบใหญ่ (หลักหมื่น-แสนบรรทัด) แบบ Waterfall แบ่งเป็น Basic, Middle, Detail

Function Point (FP)



- ประเมินโดยนับจำนวนฟังก์ชัน เช่น หน้าจอ Input, รายงาน Output, External Interface
- **จุดเด่น:** ประเมินขนาดได้โดยไม่พึ่งพาสภาพแวดล้อมหรือภาษาโปรแกรมที่ใช้พัฒนา

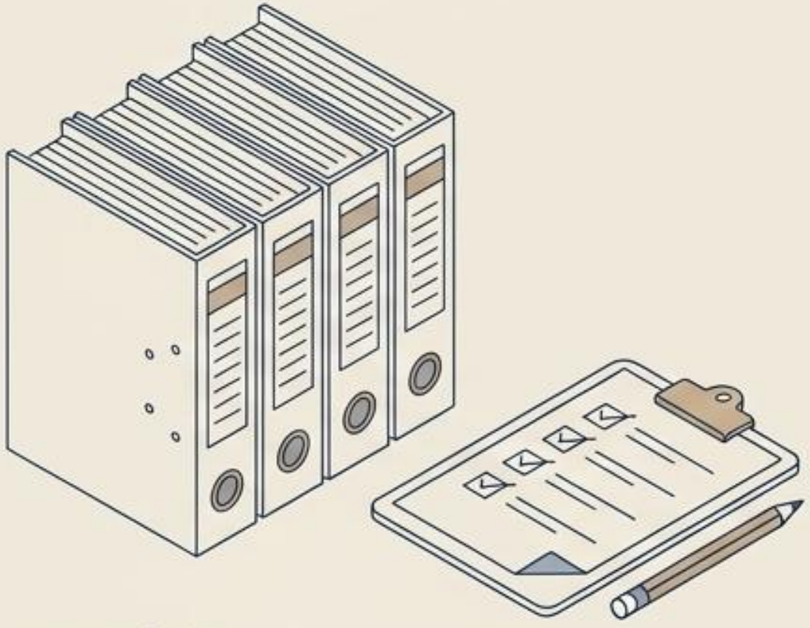
NotebookLM

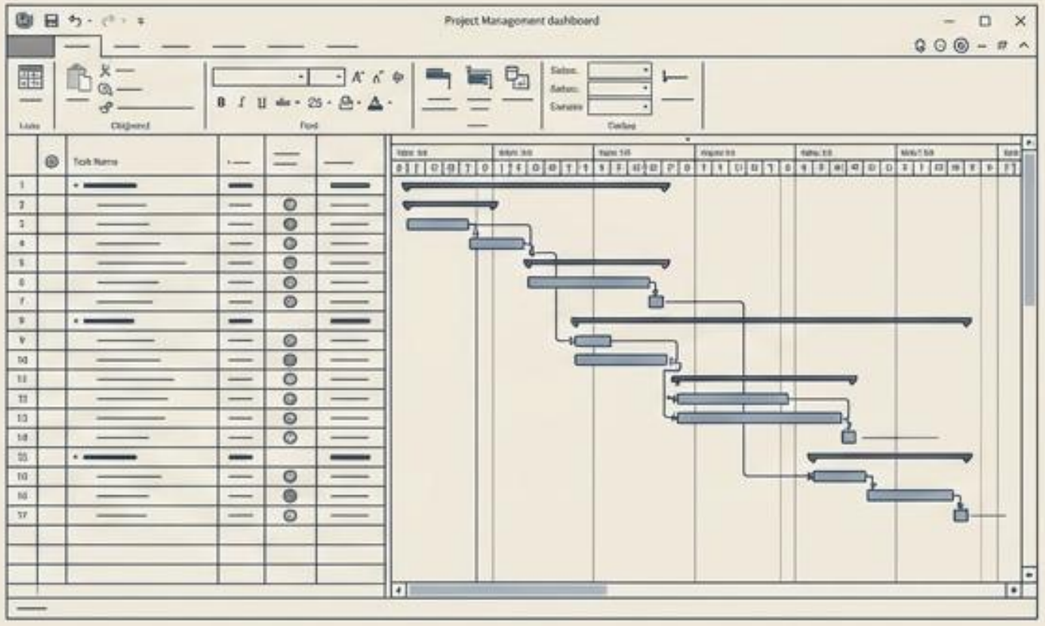
[Synthesis] ตารางวิเคราะห์เทคนิคการประมาณการต้นทุน

Technique	Details	Caution
Comparative	ต้นทุนต่ำสุด / เหมาะกับเฟสเริ่มต้น	⚠ ระวัง: ความแม่นยำต่ำ
WBS	แม่นยำสูง / เห็นภาพรายละเอียด	⚠ ระวัง: ใช้เวลาและค่าใช้จ่ายในการประเมินสูง
Standard Tasks	เป็นระบบมาตรฐาน / คำนวณง่าย	⚠ ระวัง: ต้องปรับปรุงค่าสถิติองค์กรให้ทันสมัยเสมอ
COCOMO	เป็นวิทยาศาสตร์ / เหมาะกับโปรเจกต์โค้ดมหาศาล	⚠ ระวัง: ประมาณการยากถ้าไม่ใช่ Waterfall
Function Point	ไม่ผูกติดกับเทคโนโลยี / อิงจากฟังก์ชั่นผู้ใช้	⚠ ระวัง: ต้องแปลงคะแนน (Point) กลับเป็น Man-days อีกชั้นหนึ่ง

NotebookLM

การส่งมอบ: เอกสารประกอบโครงการและซอฟต์แวร์ควบคุม





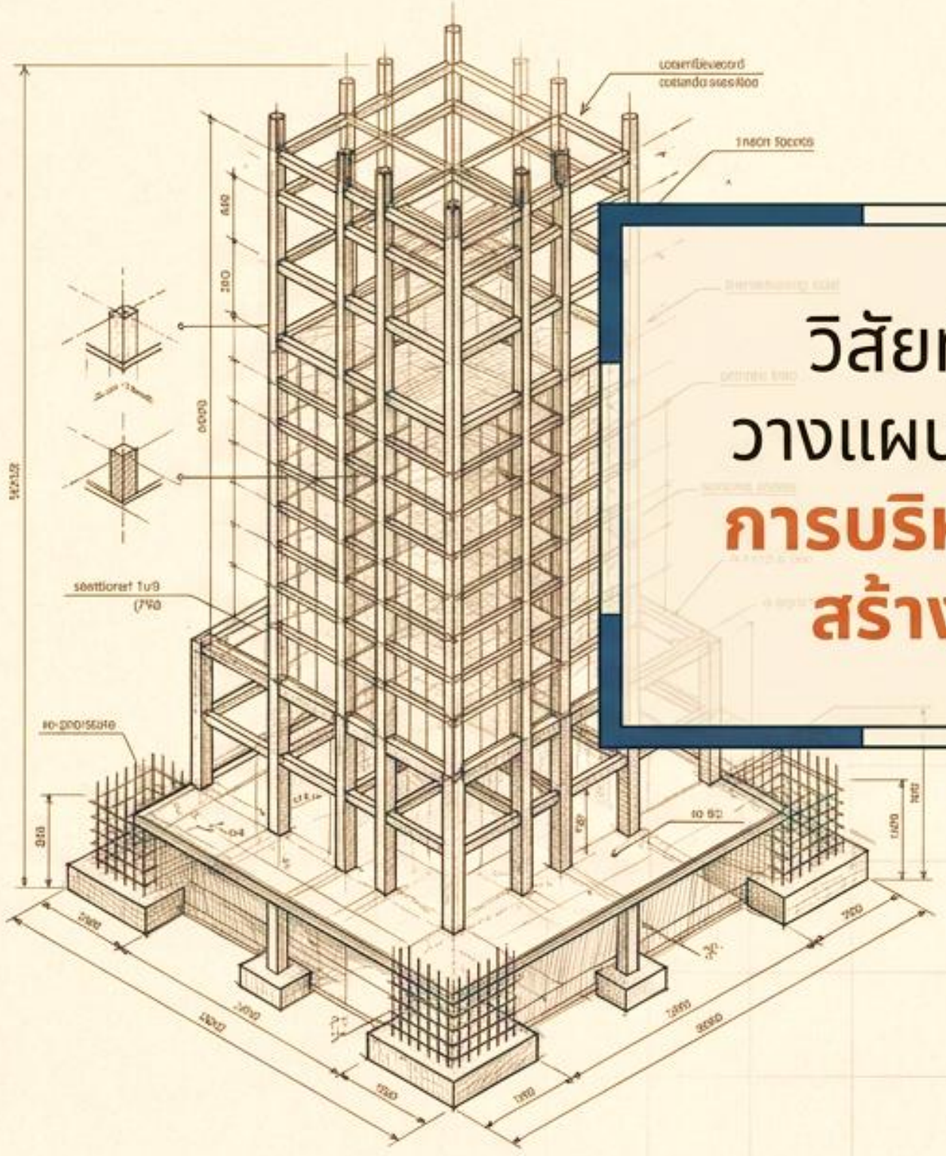
เอกสารสำคัญ:
- แผนบริหารโครงการ & แผนทีมงาน
- ผลผลิตโครงการ (Deliverables)
- แผนเนื้อหางาน / WBS
- แผนการจัดซื้อจัดจ้าง

ซอฟต์แวร์บริหารจัดการ:
- เครื่องมืออย่าง Microsoft Project ช่วยเชื่อมโยง Gantt & PERT
- ทำให้ผู้มีส่วนได้ส่วนเสียเห็นภาพรวมความคืบหน้าตรงกัน

“วิสัยทัศน์ที่ปราศจากการวางแผนเป็นเพียงแค่ความฝัน — การบริหารโครงการ IT คือการสร้างพิมพ์เขียวให้เป็นจริง”

NotebookLM

เปลี่ยนวิสัยทัศน์ให้กลายเป็นความจริง



วิสัยทัศน์ที่ปราศจากการวางแผนเป็นเพียงแค่ความฝัน — **การบริหารโครงการ IT** คือการ**สร้างพิมพ์เขียว**ให้เป็นจริง

NotebookLM