Axiom-PMO

v2.2.0 · MIT

The governance control plane for AI-assisted software delivery.

ศูนย์กลางกำกับดูแลงานพัฒนาซอฟต์แวร์ที่มี AI ร่วมทำงาน

WHAT IT IS / มันคืออะไร

A repo-native, deterministic governance layer for AI coding agents.

ชั้นกำกับดูแลที่อยู่ใน repository ทำงานแบบตรวจสอบได้แน่นอน

AI agents can write code. They should not invent the project.

AI เขียนโค้ดเก่ง แต่ไม่ควรเป็นคนกำหนด requirement, scope หรืออนุมัติงานของตัวเอง

THE PROBLEM IT SOLVES

Axiom-PMO is the mechanism that stops an AI agent from inventing anything the evidence does not support.

Axiom-PMO คือกลไกที่กันไม่ให้ AI agent "คิดเอง" นอกเหนือจากสิ่งที่มีหลักฐานจริง

Approval gates / ลำดับ Gate การอนุมัติ

แกนกลางของระบบ — งานเดินหน้าได้เมื่อผ่าน gate เท่านั้น

01 **Draft** — อ่านต้นฉบับ ตั้งต้นโครงการ — START

→ 02 **Scope** — ทุก requirement มี source_ref — HUMAN · SCOPE APPROVED

→ 03 **Design** — ออกแบบเท่าที่จำเป็นต่อการสร้าง — HUMAN · DESIGN READY

→ 04 **Handoff** — ตรวจความพร้อมก่อนส่งต่อ — READINESS CHECK · ไม่ใช่จุดอนุมัติใหม่

→ 05 **Release** — ตรวจหลักฐาน อนุมัติ ปิดงาน — HUMAN · RELEASE APPROVED

3 human approval points Scope Approved · Design Ready · Release Approved — **AI ไม่มีสิทธิ์อนุมัติงานของตัวเองในจุดใดเลย** · แถบสีแดง = จุดที่ต้องมีมนุษย์ผู้รับผิดชอบลงชื่ออนุมัติ

3 Cores / โครงสร้างการทำงาน

แต่ละ core มี artifact ที่ต้องส่งมอบชัดเจน

CORE 1

Discovery & Product Design

เข้าใจต้นฉบับ ยืนยัน scope ออกแบบเท่าที่จำเป็น

`PROJECT.md` `DESIGN/FLOW.puml` `wireframe (ถ้ามี)`

CORE 2

Delivery & Engineering

แตกงาน ส่งต่อให้ทีม/AI ทำ ตรวจความพร้อมทาง engineering

`DELIVERY.md` `HANDOFF.md`

CORE 3

Quality & Release

ตรวจสอบ ทดสอบ อนุมัติ ปิดงาน release อย่างปลอดภัย

`RELEASE.md` `RAID-log.md` `decision-log.md`

Does / Does not — ทำ / ไม่ทำ

DOES / ทำ	DOES NOT / ไม่ทำ
แปลงต้นฉบับ (MOM / Transcript / Requirement) ให้เป็น requirement ที่ตรวจสอบย้อนกลับได้ (traceable)	ไม่เขียนระบบให้คุณ — ไม่ใช่ execution framework
ตรวจสอบความพร้อมของ design และความครบถ้วนของ handoff	ไม่แทนที่ทีมพัฒนา
ตรวจสอบ scope, test, หลักฐาน และสิทธิ์การอนุมัติ	ไม่แทนที่ Jira / Azure DevOps / Linear
ตรวจสอบว่าสิ่งที่ AI รายงานว่าทำ ตรงกับสถานะจริงใน repository หรือไม่	ไม่ใช่ project management tool ที่มี dashboard / portfolio management / KPI tracking

3 Modes — โหมดตามความเสี่ยง

Lite RISK · LOW	งานความเสี่ยงต่ำ เอกสารขั้นต่ำ
Standard RISK · NORMAL	งานปกติ มี design / flow ตามที่งานต้องการ
Strict RISK · HIGH	งานความเสี่ยงสูง (การเงิน, ข้อมูลส่วนบุคคล, สิทธิ์การเข้าถึง, ระบบที่แก้คืนไม่ได้) ต้องมี RAID-log, decision-log และการอนุมัติแยกจากคนทำ

หลักการ: เลือกโหมดที่เล็กที่สุดเท่าที่ยังคุมความเสี่ยงจริงได้ — ไม่ใช่ยิ่งเข้มยิ่งดีเสมอไป

Evidence system / ระบบหลักฐาน

ทุก requirement / decision / test ต้องมี `source_ref` และ `evidence_status` หนึ่งค่า

verified	supported	inferred	missing	conflict
มีต้นฉบับ + มนุษย์อนุมัติแล้ว	มีต้นฉบับ รออนุมัติ	อนุมานจากข้อมูลบางส่วน ต้องรีวิว	ไม่พบในต้นฉบับ — ห้ามใช้เป็น requirement	ต้นฉบับขัดแย้งกัน ต้องแก้ก่อน

ARTIFACTS
เอกสารที่ระบบสร้าง

`PROJECT.md` `DESIGN/FLOW.puml` `DESIGN/WIREFRAME` `DESIGN/BUILD-SPEC.md` `DELIVERY.md` `HANDOFF.md` `RELEASE.md` `RAID-log.md` `decision-log.md`

เอกสารให้ความรู้ (educational) — ไม่ใช่สื่อขาย