ธรรมาภิบาล AI Coding Agent สำหรับมหาวิทยาลัย: กรอบควบคุมสิทธิ์ ความรับผิดชอบ และการตรวจสอบ Claude Code

AI Coding Agent Governance for Universities: A Framework for Permission, Accountability, and Auditability in Claude Code

พระมหาประสพสุข ปิยภาณี (สุขล้วน)
นักวิชาการคอมพิวเตอร์ สำนักงานวิชาการ วิทยาลัยสงฆ์พุทธโสธร มหาวิทยาลัยมหาจุฬาลงกรณราชวิทยาลัย
Email: prasopsuk.suk@mcu.ac.th • DOI: 10.14457/MCU.IT.2569.006

บทคัดย่อ (Abstract) — การเปลี่ยนผ่านกระบวนการวิศวกรรมซอฟต์แวร์ระดับอุดมศึกษาสู่ยุคตัวแทนปัญญาประดิษฐ์เชิงปฏิบัติการแบบอัตโนมัติ (Autonomous AI Coding Agent) ที่ทำงานผ่าน Command Line Interface (CLI) อย่าง Claude Code ได้ยกระดับความเร็วในการพัฒนาระบบสารสนเทศอย่างก้าวกระโดด อย่างไรก็ดี ความสามารถในการเข้าถึงเทอร์มินัล การแก้ไขไฟล์โค้ดทั้งโครงการ และการรันสคริปต์อัตโนมัติ ก่อให้เกิดความเสี่ยงระดับวิกฤติต่อความมั่นคงปลอดภัยไซเบอร์ ข้อมูลส่วนบุคคล และการหลุดรอดของช่องโหว่ซอฟต์แวร์ บทความวิชาการนี้มีวัตถุประสงค์เพื่อศึกษา วิเคราะห์ และนำเสนอ **"กรอบธรรมาภิบาล AI Coding Agent สำหรับมหาวิทยาลัย" (Higher Education AI Coding Governance Framework: U-GovCode)** ซึ่งประกอบด้วย ๓ เสาหลัก: ๑) กรอบการควบคุมสิทธิ์แบบจำแนกขอบเขต (Permission Control & Least Privilege) ๒) ความรับผิดชอบและการกำกับดูแลโดยมนุษย์ (Accountability & Human-in-the-loop: HITL) และ ๓) การตรวจสอบย้อนกลับและบันทึกประวัติแบบไม่สามารถปลอมแปลงได้ (Auditability & Immutable Trajectory Logs) สอดคล้องกับมาตรฐาน ISO/IEC 42001, NIST AI RMF และ พ.ร.บ. คุ้มครองข้อมูลส่วนบุคคล พ.ศ. ๒๕๖๒ จากการสังเคราะห์แนวทางการประยุกต์ใช้ในระบบสารสนเทศมหาวิทยาลัยชี้ให้เห็นว่า กรอบ U-GovCode ช่วยวางกลไกยับยั้งคำสั่งที่มีความเสี่ยงสูงต่อระบบ สร้างประวัติการตรวจสอบย้อนกลับที่สมบูรณ์ และลดความเสี่ยงการละเมิดข้อมูลส่วนบุคคลได้อย่างเป็นระบบ

คำสำคัญ: ธรรมาภิบาลปัญญาประดิษฐ์, AI Coding Agent, Claude Code, กรอบควบคุมสิทธิ์, Human-in-the-loop, การตรวจสอบย้อนกลับ, ISO/IEC 42001, ระบบสารสนเทศมหาวิทยาลัย

๑. บทนำและความเป็นมา (Introduction)

ความก้าวหน้าอย่างก้าวกระโดดของโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) ได้พัฒนาจากการเป็นผู้ช่วยสร้างโค้ดแบบ Copilot ในโปรแกรมแก้ไขข้อความ สู่การเป็น **ตัวแทนอัตโนมัติเชิงปฏิบัติการ (Autonomous Agentic AI)** เช่น Claude Code ซึ่งทำงานในระดับ Command-line Interface (CLI) สามารถอ่านทำความเข้าใจสถาปัตยกรรมทั้งระบบ ทำการแก้ไข ค้นหา วินิจฉัยข้อผิดพลาด และรันคำสั่ง Bash ได้อย่างเป็นอิสระ

สำหรับสถาบันอุดมศึกษา การนำ AI Coding Agent มาใช้ช่วยให้ทีมพัฒนาขนาดเล็กของมหาวิทยาลัยสามารถส่งมอบบริการดิจิทัลได้อย่างรวดเร็ว อย่างไรก็ตาม ความสามารถดังกล่าวกลับแฝงด้วยความเสี่ยงระดับสูง ๓ ประการ ได้แก่ ๑) *ความเสี่ยงด้านการเข้าถึงสิทธิ์เกินจำเป็น (Privilege Escalation)* เช่น การเผลอรันคำสั่งทำลายล้างข้อมูล หรือลบตารางฐานข้อมูล ๒) *ความรับผิดชอบที่ไม่ชัดเจน (Accountability Void)* เมื่อเกิดข้อผิดพลาดในการประมวลผล และ ๓) *การขาดร่องรอยการตรวจสอบ (Lack of Auditability)* ซึ่งขัดต่อข้อกำหนด ISO/IEC 27001 และ PDPA

๒. แนวคิดและมาตรฐานอ้างอิง

บทความวิชาการนี้ได้บูรณาการกรอบมาตรฐานสากล ๓ ฉบับเข้าด้วยกัน ได้แก่:

๑) ISO/IEC 42001:2023 (Artificial Intelligence Management System): กำหนดกรอบการบริหารจัดการความเสี่ยงปัญญาประดิษฐ์ การประเมินผลกระทบต่อองค์กร และความโปร่งใสของวงจรชีวิต AI

๒) NIST AI Risk Management Framework (AI RMF 1.0): มุ่งเน้นการควบคุมความถูกต้อง น่าเชื่อถือ ปลอดภัย และการตอบสนองต่อเหตุการณ์ฉุกเฉิน

๓) OWASP Top 10 for LLM Applications: เน้นการป้องกันความเสี่ยงด้าน Prompt Injection, Insecure Output Handling, และ Excessive Agency (การมอบอำนาจให้ Agent มากเกินขอบเขต)

ผู้เขียนจึงได้สังเคราะห์โมเดล **U-GovCode** เพื่อเป็นสะพานเชื่อมระหว่างข้อกำหนดด้านนโยบายสถาบัน สู่มาตรการควบคุมทางเทคนิคที่สามารถฝังในเครื่องมือ Claude Code ได้จริง

๓. กรอบการควบคุมสิทธิ์ (Permission Control)

หัวใจสำคัญของธรรมาภิบาล AI คือการจำกัด *ขอบเขตอำนาจกระทำ (Agency Boundaries)* ไม่ให้ตัวแทน AI สามารถทำสิ่งที่เกินขอบเขตงานได้ กรอบ U-GovCode กำหนดสถาปัตยกรรมควบคุมสิทธิ์เป็น ๓ ระดับ (Tiered Permissions):

๓.๑ ระดับการอนุญาตและขอบเขตการทำงาน (Tiered Execution)

- **Tier 1: Read-Only Operations (สิทธิ์ระดับอ่าน):** สิทธิ์ในการค้นหาไฟล์ อ่านเนื้อหา สรุปโครงสร้าง และตรวจสอบ Log การทำงาน ตัวแทน AI สามารถทำได้โดยอัตโนมัติ ไม่ต้องร้องขอการอนุมัติ
- **Tier 2: Workspace Mutation (สิทธิ์แก้ไขในพื้นที่ทำงาน):** การสร้าง แก้ไข หรือลบไฟล์ซอร์สโค้ดภายใน Repository ของโครงการ ตัวแทน AI สามารถดำเนินการได้ แต่ต้องแสดงการเปลี่ยนแปลง (Diff) ชัดเจน
- **Tier 3: Privileged Execution (สิทธิ์ระดับวิกฤติ):** การสั่งรันคำสั่งเทอร์มินัล การแก้ไขฐานข้อมูล การติดตั้งโปรแกรมส่วนกลาง และการส่งโค้ดขึ้น Git Remote ต้องผ่านการอนุมัติจากวิศวกรผู้ควบคุมทุกครั้ง

> **หลักการ Least Privilege ใน Claude Code:**
> ระบบจะไม่อนุญาตให้รันคำสั่งด้วยสิทธิ์ Superuser (sudo) และบล็อกคำสั่งกลุ่มทำลายล้าง เช่น `rm -rf /`, `DROP DATABASE`, `TRUNCATE TABLE` โดยเด็ดขาดผ่าน Regular Expression Hook

๓.๒ ทรัพยากรคุ้มครองพิเศษ (Protected Assets)

กรอบ U-GovCode กำหนดให้ไฟล์คอนฟิกูเรชันที่มีความอ่อนไหวสูง เช่น `.env`, รหัสผ่านฐานข้อมูล, API Keys และใบรับรองดิจิทัล เป็น "Protected Assets" ซึ่งตัวแทน AI ไม่สามารถอ่านค่าดิบ (Plaintext) หรือเขียนทับได้ หากต้องการคอนฟิก ระบบจะส่งมอบเป็นตัวแปรจำลอง (Mocked Credentials) ในช่วงพัฒนา

๔. ความรับผิดชอบ (Accountability & HITL)

การนำ AI มาใช้พัฒนาซอฟต์แวร์ต้องไม่ทำให้ความรับผิดชอบสูญหายไปสู่ "กล่องดำ" (Accountability Void) กรอบ U-GovCode วางโครงสร้างความรับผิดชอบผ่าน ๒ กลไกหลัก:

๔.๑ เมทริกซ์ความรับผิดชอบ RACI สำหรับ AI Code Generation

ตารางที่ ๑: RACI Matrix สำหรับการพัฒนาระบบด้วย Claude Code

กิจกรรม	AI Agent	Developer	SecOps
วิเคราะห์และเสนอแนะสถาปัตยกรรม	R	A	C
เขียนและแก้ไขซอร์สโค้ด	R	A	I
อนุมัติการรันคำสั่งเทอร์มินัล	-	A/R	C
ตรวจสอบช่องโหว่ความปลอดภัย	R	C	A
อนุมัติการ Deploy สู่ Production	-	C	A

* R = Responsible (ผู้ปฏิบัติ), A = Accountable (ผู้รับผิดชอบสูงสุด), C = Consulted (ผู้ให้คำปรึกษา), I = Informed (ผู้รับทราบ)

๔.๒ ประตูกำกับดูแลโดยมนุษย์ (HITL Gateways)

มนุษย์ยังคงเป็นศูนย์กลางการตัดสินใจ (Human-on-the-loop / Human-in-the-loop) ในกรณีที่ Claude Code เสนอคำสั่งที่มีผลกระทบกว้างขวาง ระบบจะต้องหยุดรอการยืนยันจากวิศวกร (Explicit Human Consent) พร้อมคำอธิบายเหตุผลของการกระทำ (Rationale Summary) เสมอ

๕. สถาปัตยกรรมกรอบธรรมาภิบาล U-GovCode Framework

รูปที่ ๑ แสดงความสัมพันธ์เชิงสถาปัตยกรรมระหว่าง Claude Code Engine, Security Gateways, และการควบคุมสอดคล้องตามมาตรฐานสากล:



รูปที่ ๑: แผนภาพสถาปัตยกรรมธรรมาภิบาล AI Coding Agent (U-GovCode Architecture)

๖. การตรวจสอบย้อนกลับ (Auditability)

ตามมาตรฐาน ISO/IEC 42001 ระบบ AI จะต้องสามารถอธิบายการตัดสินใจ (Explainability) และตรวจสอบย้อนกลับได้ (Auditability & Traceability) กรอบ U-GovCode กำหนดกลไก **Immutable Trajectory Ledger** ซึ่งบันทึกข้อมูล ๔ องค์ประกอบ:

1. **Prompt & Intent:** คำสั่งของมนุษย์พร้อมบริบทและข้อจำกัดที่กำหนดไว้
2. **Thought & Reasoning:** ลำดับความคิดและการตัดสินใจของโมเดลก่อนลงมือปฏิบัติการ
3. **Tool Invocation & Arguments:** รายการคำสั่ง พารามิเตอร์ และไฟล์ที่ถูกเรียกใช้จริง
4. **Diff & System Impact:** ผลลัพธ์การเปลี่ยนแปลงไฟล์ สถานะความปลอดภัย และสถานะ Git Commit

ไฟล์ Log จะถูกจัดเก็บในรูปแบบ JSON Lines (JSONL) มีการเข้ารหัสเชิงตัวเลข (Cryptographic Hash) ป้องกันการดัดแปลงย้อนหลัง เพื่อใช้เป็นหลักฐานประกอบการตรวจประเมินความมั่นคงปลอดภัยประจำปี

๗. การคุ้มครองข้อมูลสถาบันและ PDPA

ในการพัฒนาระบบสารสนเทศมหาวิทยาลัย มักเกี่ยวข้องกับข้อมูลส่วนบุคคลของนิสิต คณาจารย์ และผลการเรียน กรอบ U-GovCode กำหนดมาตรการคุ้มครองข้อมูล ๓ ระดับ:

- **Data Anonymization at Ingestion:** การแทนที่ข้อมูลจริงด้วยชุดข้อมูลจำลอง (Synthetic / Mock Data) ก่อนส่งบริบทให้โมเดลประมวลผล
- **Telemetry Redaction:** สกัดกั้นหมายเลขประจำตัวประชาชน เบอร์โทรศัพท์ และอีเมลส่วนบุคคลไม่ให้ปรากฏใน Prompt หรือ Trajectory Logs
- **On-Premises / Enterprise API Boundary:** กำหนดให้เชื่อมต่อเฉพาะโมเดลที่ให้คำรับรองว่าจะไม่นำข้อมูล Prompt ไปฝึกฝนต่อ (Zero Data Retention / No Training Policy)

๘. แนวทางการประยุกต์ใช้ในระบบสารสนเทศมหาวิทยาลัย

ผู้เขียนได้สังเคราะห์แนวทางการนำกรอบ **U-GovCode** ไปประยุกต์ใช้ในโครงการพัฒนาระบบสารสนเทศของวิทยาลัยสงฆ์พุทธโสธร ครอบคลุมระบบบริหารวิชาการ ระบบแจ้งซ่อม ระบบจองห้องปฏิบัติการ และระบบคลังความรู้ดิจิทัล

ในการวิเคราะห์เปรียบเทียบเชิงโครงสร้าง ผู้เขียนจำแนกการดำเนินงานออกเป็น ๒ รูปแบบ:

- **รูปแบบปกติ (Default / Autonomous Mode):** ใช้งาน Claude Code โดยไม่มีการวางกรอบควบคุมสิทธิ์และไม่มีตัวดักจับคำสั่งล่วงหน้า
- **รูปแบบภายใต้กรอบธรรมาภิบาล (U-GovCode Mode):** บังคับใช้ Tiered Permissions, HITL Gateways และ Immutable Audit Ledger ครบทุกขั้นตอน

เกณฑ์การวิเคราะห์ความมั่นคงปลอดภัยและธรรมาภิบาลประกอบด้วย ๔ มิติ: การสกัดกั้นคำสั่งอันตราย (Dangerous Command Interception), ความสมบูรณ์ของบันทึกการตรวจสอบ (Audit Traceability), การป้องกันช่องโหว่ความปลอดภัย และการคุ้มครองข้อมูลความลับของสถาบัน

๙. การวิเคราะห์เปรียบเทียบกลไกการกำกับดูแล

เมื่อพิจารณาเหตุการณ์ความเสี่ยงที่มักเกิดขึ้นในการใช้ AI Coding Agent เช่น การเขียนทับไฟล์คอนฟิก, การรันคำสั่งลบข้อมูล, การติดตั้งแพ็กเกจภายนอก และการส่งโค้ดขึ้น Production สามารถเปรียบเทียบกลไกการควบคุมได้ดังตารางที่ ๒:

ตารางที่ ๒: การเปรียบเทียบกลไกควบคุมระหว่างการใช้งานแบบปกติและภายใต้กรอบ U-GovCode

มิติธรรมาภิบาล	โหมดปกติ (Default)	กรอบ U-GovCode	ผลลัพธ์เชิงระบบ
การสกัดกั้นคำสั่งอันตราย	พึ่งพาการสังเกตด้วยมือ	Regex Hook บล็อก 100%	ปลอดภัยสูงสุด
ความสมบูรณ์ของ Audit Trail	เฉพาะข้อความ Git Commit	JSONL Trajectory ครบถ้วน	ตรวจสอบย้อนกลับได้
การควบคุมช่องโหว่ซอฟต์แวร์	ตรวจภายหลังเมื่อเกิดปัญหา	ตรวจทานก่อนรวมโค้ด	ลดช่องโหว่หลุดรอด
การป้องกันข้อมูลลับ/กุญแจรั่วไหล	เสี่ยงอ่านไฟล์ .env โดยตรง	Protected Asset Masking	สอดคล้อง PDPA
ความชัดเจนของความรับผิดชอบ	คลุมเครือ (Accountability Void)	กำหนด RACI & HITL ชัดเจน	สอดคล้อง ISO 42001

จากการวิเคราะห์เปรียบเทียบในตารางที่ ๒ ชี้ให้เห็นว่า กรอบ U-GovCode ช่วยปิดช่องว่างความเสี่ยงจากการปฏิบัติการเกินสิทธิ์ของ AI Coding Agent ได้อย่างเป็นรูปธรรม โดยเปลี่ยนการควบคุมตามดุลยพินิจรายบุคคลให้เป็นกลไกมาตรฐานระดับสถาบัน

๑๐. การอภิปรายประเด็นสำคัญ (Academic Discussion)

จากการสังเคราะห์กรอบแนวคิดชี้ให้เห็นว่า การใช้งาน Agentic AI อย่าง Claude Code ให้ประสบความสำเร็จในระดับองค์กร ไม่ได้ขึ้นอยู่กับความฉลาดของตัวโมเดลเพียงอย่างเดียว แต่ขึ้นอยู่กับ *"ความแข็งแกร่งของสถาปัตยกรรมกำกับดูแล" (Governance Rigor)*

๑) การรักษาสมดุลระหว่างความเร็วกับความปลอดภัย: หากปล่อยให้ AI ทำงานอย่างอิสระโดยไร้กรอบควบคุม แม้การเขียนโค้ดช่วงแรกจะทำได้รวดเร็ว แต่วิศวกรต้องเสียเวลาไล่ตรวจสอบและแก้ไขปัญหาภายหลัง ในขณะที่การใช้กรอบ U-GovCode มีการตรวจสอบที่ชัดเจนตั้งแต่ต้น (Shift-left Verification) จึงช่วยลดความเสี่ยงและเวลาการแก้ไขข้อผิดพลาดโดยรวมลง

๒) การตอบสนองต่อเกณฑ์มาตรฐาน ISO 42001: การมี Immutable Trajectory Logs ช่วยปลดล็อกข้อกังวลในการนำ AI มาใช้กับระบบสำคัญของมหาวิทยาลัย เนื่องจากสามารถระบุได้ชัดเจนว่าโค้ดบรรทัดใดถูกสร้างโดยใคร และคำสั่งใดได้รับการอนุมัติโดยวิศวกรคนใด ทำให้มีหลักฐานโปร่งใสพร้อมรับการตรวจประเมิน

๑๑. ข้อเสนอแนะเชิงนโยบายสำหรับสถาบันอุดมศึกษา

จากกรอบแนวคิดที่นำเสนอ ผู้เขียนขอเสนอแนวทางปฏิบัติ ๔ ประการสำหรับผู้บริหารเทคโนโลยีสารสนเทศระดับสูง (CIO / CTO) ของสถาบันอุดมศึกษา:

1. **จัดทำนโยบาย AI Coding Agent Policy:** กำหนดบทบาทหน้าที่ และระดับความลับของข้อมูลที่อนุญาตให้ AI เข้าถึงอย่างเป็นลายลักษณ์อักษร
2. **บังคับใช้กลไก Technical Interceptors:** ไม่ควรพึ่งพาเพียงข้อตกลงทั่วไป แต่ต้องติดตั้ง Git Hooks, Safe Shell Execution, และ Pre-commit Guards ในเครื่องของนักพัฒนาทุกคน
3. **พัฒนาทักษะวิศวกรเป็น "AI Supervisor":** ปรับเปลี่ยนบทบาทของโปรแกรมเมอร์จากการเป็นผู้เขียนโค้ดบรรทัดต่อบรรทัด สู่การเป็น "ผู้ตรวจสอบและกำกับดูแลความถูกต้องเชิงสถาปัตยกรรม" (Architectural Reviewer)
4. **เชื่อมโยงกับมาตรฐานสากล:** ผลักดันให้ระบบสารสนเทศของสถาบันสอดคล้องกับมาตรฐาน ISO/IEC 42001 เพื่อสร้างความเชื่อมั่นแก่ประชาคมวิชาการและสังคม

๑๒. บทสรุป (Conclusion)

กรอบธรรมาภิบาล AI Coding Agent สำหรับมหาวิทยาลัย (U-GovCode) ที่นำเสนอในบทความวิชาการนี้ เป็นแนวทางเชิงโครงสร้างในการบริหารจัดการความเสี่ยงของ Claude Code ในสภาพแวดล้อมสถาบันอุดมศึกษา การประสาน ๓ เสาหลัก อันได้แก่ การควบคุมสิทธิ์ตามลำดับขั้น ความรับผิดชอบที่ยึดมนุษย์เป็นศูนย์กลาง และการตรวจสอบย้อนกลับอย่างโปร่งใส ช่วยให้มหาวิทยาลัยสามารถใช้ประโยชน์จากศักยภาพของ AI ได้อย่างเต็มที่ โดยยังคงรักษาความมั่นคงปลอดภัย ข้อมูลส่วนบุคคล และมาตรฐานธรรมาภิบาลทางวิชาการไว้อย่างสมบูรณ์

กิตติกรรมประกาศ (Acknowledgments):

บทความวิชาการนี้ได้รับการสนับสนุนทรัพยากรระบบสารสนเทศและข้อมูลวิชาการจากสำนักงานวิชาการ วิทยาลัยสงฆ์พุทธโสธร มหาวิทยาลัยมหาจุฬาลงกรณราชวิทยาลัย ประจำปี พ.ศ. ๒๕๖๙ ผู้เขียนขอขอบคุณผู้บริหาร คณาจารย์ และบุคลากรทุกท่านที่ให้ข้อเสนอแนะอันเป็นประโยชน์ยิ่ง

เอกสารอ้างอิง (References)

1. Anthropic. (2025). *Claude Code: An Agentic Coding Tool for the Terminal Architecture & Security Overview*. Anthropic Research Technical Whitepaper.
2. ISO/IEC. (2023). *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*. International Organization for Standardization.
3. National Institute of Standards and Technology (NIST). (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST Trustworthy and Responsible AI, SP 1270.
4. OWASP Foundation. (2025). *OWASP Top 10 for Large Language Model Applications v2.0*. Open Web Application Security Project.
5. พระมหาประสพสุข ปิยภาณี. (๒๕๖๙). กรอบการพัฒนาระบบสารสนเทศมหาวิทยาลัยอย่างมั่นคงปลอดภัยด้วย Claude Code: การบูรณาการ Human-in-the-loop, DevSecOps และธรรมาภิบาลปัญญาประดิษฐ์. *วารสารนวัตกรรมเทคโนโลยีสารสนเทศและการสื่อสาร มจร*, ๓(๑), ๔๕-๕๖.
6. พระมหาประสพสุข ปิยภาณี. (๒๕๖๙). การพัฒนา Skill-based Governance สำหรับควบคุมการสร้างระบบสารสนเทศมหาวิทยาลัยด้วย Google Antigravity. *วารสารวิชาการพุทธนวัตกรรมดิจิทัล*, ๒(๑), ๑-๑๔.
7. ราชกิจจานุเบกษา. (๒๕๖๒). พระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล พ.ศ. ๒๕๖๒. เล่ม ๑๓๖ ตอนที่ ๖๙ ก. ๒๗ พฤษภาคม ๒๕๖๒.
8. Sanders, M., & Roberts, K. (2025). Permission Models and Least Privilege Boundaries in Autonomous Developer Agents. *IEEE Software*, 42(2), 34-43.
9. Venkatesh, V., & Bala, H. (2024). Human-Agent Collaboration in Enterprise Software Engineering: A Governance Perspective. *MIS Quarterly*, 48(4), 1120-1145.
10. Zhang, L., et al. (2025). Auditability and Verifiability of Agentic Code Modifications in Safety-Critical Systems. *ACM Transactions on Software Engineering and Methodology*, 34(3), 1-28.