Course Launch Completion and Scale Your Intelligence

ECE 57000 — August 28, 2026

David I. Inouye

Wednesday focused on questions about the AI Product

- We walked through the AI Product Student Guide in depth.
- We discussed stakeholder paths, shared stakeholders, scope, evidence, and grading.
- The guide and milestone packages remain the authoritative detailed instructions.
- **Course material stated or explained in lecture—even when it does not appear on a slide—is fair game for quizzes and exams.**

Oral quizzes are closed-book during each attempt

- **During an attempt:** use no notes, AI, websites, classmates, or other sources.
- **Between attempts:** use AI, notes, course materials, and discussion to learn before retaking.
- **Never share or receive** specific quiz or exam questions, prompts, topics that appeared, or other assessment content.
- General discussion of course topics is welcome.

Using prohibited help—or sharing or receiving assessment content—will result in an F in the course under the syllabus policy.

We will now finish the syllabus and schedule walkthrough

- Open the required course syllabus
- Open the tentative course schedule

The syllabus is **required reading**. Quiz questions may assess syllabus content.

Open-to-all study groups can earn up to 3 bonus points

- Earn **0.3 percentage points** per qualifying week; ten weeks reaches the maximum.
- A group has at least two students, remains open to the class, and meets about 30–45 minutes.
- Submit one Gradescope group list, a meeting photo or screenshot, and an audio or video recording.
- Meetings are entirely in person or entirely online—not hybrid—and use one lead contact.
- Automated checks support manual review; ordinary warnings retain credit, but an unusable recording must be resubmitted.

See Piazza for the authoritative logistics, group directory, and submission details.

Evacuate to the assembly area across North Street

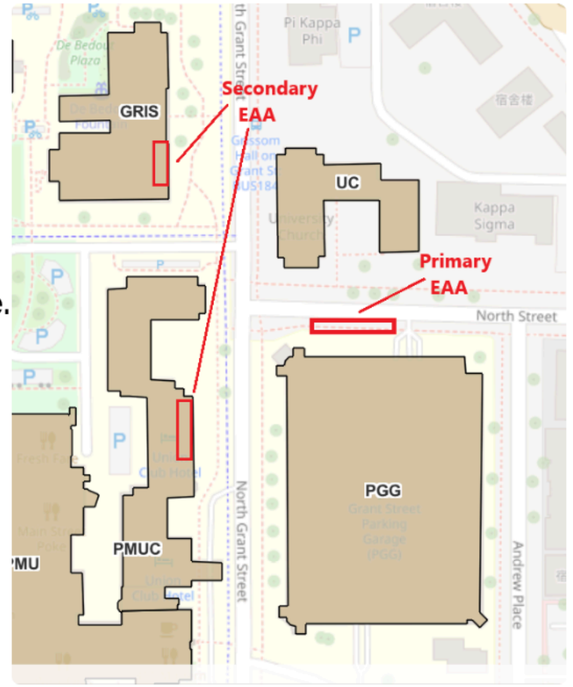
Once out of the building, occupants shall meet at the following Evacuation/Emergency Assembly Area (EAA) locations:

1. Primary EAA Location (should be outside, in an area away from the building):

Gather on sidewalk away from the building and building entrances across the street by parking garage.

2. Secondary EAA Location (should be inside a nearby building in case of inclement weather):

Purdue Memorial Union or Grissom Hall



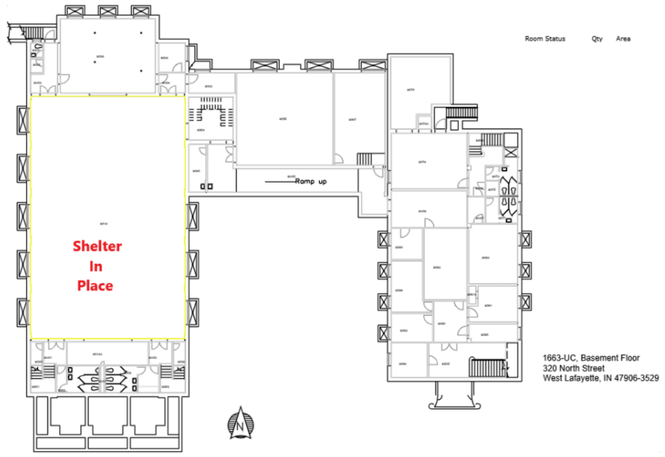
Shelter from severe weather in UC B014

SHELTER IN PLACE

During a severe weather incident (such as a tornado warning), occupants can take shelter in a safe location, such as an interior room with no windows, ideally in the lower level of the building.

Severe weather shelter-in-place options in this building include, but are not limited to:

Go to lowest level possible – study lounge in the basement UC B014.



Complete these first-week actions

1. Read the syllabus and tentative schedule.
2. Confirm access to Piazza and Gradescope; check Piazza regularly.
3. Complete the **ungraded prerequisite quiz by Monday, August 31**.
4. Check OpenAI's four-month student ChatGPT Plus offer now. Eligibility and availability are not guaranteed.
5. Maintain Claude Pro or **ChatGPT Plus** access—the approximately $20/month plans, not ChatGPT Pro; wait to buy Colab Pro until it is needed.

Scale Your Intelligence

AI scales the quality of your judgment

> **AI and agents can scale your intelligence—but the results will reflect the quality of your judgments.**

- Weak understanding and poor discernment can scale confusion, drift, or harm.
- Strong understanding and careful judgment can scale useful work.
- The human remains central: choose values, define what is good, shape the environment, and make consequential tradeoffs.

Goal: AI should optimize your thinking—not replace it.

MAGE Case Study

Scaling reliable agentic software engineering

Chat gathers and synthesizes information; agents change the work

Mode	Best suited to	Concrete examples
AI chat	Simple information gathering, explanation, summarization, and one-off synthesis	Claude chat; ChatGPT chat
AI agent	Inspecting artifacts, using tools, editing files, running checks, and pursuing a multi-step outcome	Claude Code; Codex

- Chat makes repeatable processes difficult because the work remains inside a conversation.
- Use chat for bounded information tasks—not sustained project work.
- Move consequential, multi-step work into an agentic environment.

Cheap code changes the engineering bottleneck

- Coding agents can produce and revise implementation extremely quickly.
- In a large system, more output creates more interactions, decisions, and possible failures.
- Requirements, architecture, evidence, deployment, maintenance, and accountability remain.

> Implementation capacity can grow faster than the judgment needed to direct it.

Paraphrase of Davis, *MAGE*, pp. 4, 18–23

Discussion: what lies between speed and certainty?

Vibe coding

- Move at breakneck speed
- Judge by visible output
- Accept hidden uncertainty

Line-by-line inspection

- Manually validate every change
- Seek maximum direct oversight
- Give up much of the speed advantage

Is there a third approach that preserves both leverage and reliability? What would it require?

Discussion framing by Inouye; engineering problem informed by Davis, *MAGE*, Part 1

Scale creates a reasoning-horizon problem

- A large system exceeds what any one human or agent can actively reason over.
- Larger context windows and retrieval move the boundary; they do not remove finite reasoning.
- Without useful external representations, each new task reconstructs system knowledge from code and history.

> **Repeated reconstruction creates churn—even when each local change looks plausible.**

Paraphrase of Davis, *MAGE*, pp. 28–45

MAGE answers the scaling problem with an engineering method

MAGE = **Model-Based Agentic Software Engineering**, a method developed by James C. Davis for engineering large software systems with coding agents.

> **Model consequential knowledge → give obligations authority → convert recurring lessons into engineering capital**

Davis, “Model-Based Agentic Software Engineering,” 2026 · Davis, *Model-Based Agentic Software Engineering*, pp. 4–6

1: Modeling makes intent explicit at useful scales

> “A model is a purposeful reduction of a system. It preserves the information one engineering question needs and suppresses the detail that question does not.” — James C. Davis, *MAGE*, p. 38

- **Explicit:** Make engineering knowledge and intent explicit.
- **Compact:** Suppress details that do not matter at that level or for that question.
- **Understandable:** Make the representation usable by both people and machines.

Definition and Modeling Principle from Davis, *MAGE*, §2.1.1, p. 38 and Part 2

1: Modeling uses different models for different questions

Engineering question	Example model
How is the system divided and connected?	Architecture, component, or dependency model
What can happen, and in what order?	State machine, workflow, scenario, or use-case model
What must or must not happen?	Policy, invariant, permission, privacy, or security model
What counts as acceptable evidence?	Measurement, metric, threshold, or assurance model
Who owns a decision or artifact?	Ownership, decision, or provenance model
How do facts connect across the system?	System knowledge graph

Examples adapted from Davis, *MAGE*, Part 2

2: Alignment gives modeled obligations authority

Models can state what should hold, but representation alone does not make the work follow the model. **Alignment gives selected obligations consequence.**

Mechanism	Role	Simple example
Constraint	Restrict action	A sandbox forbids a dangerous operation
Sensor	Produce evidence	A scan reports exposed endpoints
Validator	Judge evidence	A test checks required behavior
Gate	Control admission	CI blocks a release that fails checks

Agents remain free within permitted bounds; work that violates a governed obligation does not pass.

Paraphrase of Davis's Alignment Principle: Davis, *MAGE*, pp. 4–6 and Part 3

3: Governance conversion lets judgment accumulate as engineering capital

Churn

- Rediscover the same context
- Repair the same failure
- Repeat the same review comment
- Reconcile the same inconsistency

Governance conversion

- Capture the missing fact
- Improve the shared procedure
- Add the right evidence or control
- Let future work inherit the lesson

Paraphrase of Davis's governance-conversion loop: Davis, *MAGE*, pp. 4–6 and Part 4

MAGE builds a governed engineering environment

Model consequential knowledge → give obligations authority → delegate work → inspect evidence → convert recurring lessons

- **Modeling** counters reconstruction at scale.
- **Alignment** prevents probabilistic implementation from becoming its own authority.
- **Governance conversion** lets engineering effort accumulate over time.

Summary of Davis, *MAGE*, “MAGE on One Page,” p. 6

My Broader Synthesis

Scale your intelligence beyond software engineering

Effective CEOs already scale judgment through people and culture

- Focus on consequential choices: strategy, values, and tradeoffs.
- Delegate logistics, reporting, coordination, and routine decisions.
- Shape culture and feedback so others can act well without constant oversight.

Agents make this leverage accessible—but not automatically good

- Agents make delegation far cheaper and more widely available.
- Give them explicit goals and boundaries; require evidence and escalation.
- Start with bounded work, preserve lessons, and expand incrementally.

> **AI and agents can scale your intelligence—but the results will reflect the quality of your judgments.**

Offload work that does not require your attention so you can invest it where your judgment creates the most value.

Scaling intelligence requires two connected moves

> **To scale your intelligence, externalize the thinking that matters and engineer a system that can apply it reliably, test the results, and learn from corrections.**

Make your thinking explicit and structured

1. Preserve consequential context.
2. Organize it into useful models and levels.
3. Bring the right representation to each decision.

Turn judgment into a reliable system

4. Define what good means and what evidence counts.
5. Give important rules authority and preserve lessons.
6. Formalize repeatable checks where worthwhile.

Instructor synthesis by Inouye; relationship inspired by Davis's Modeling, Alignment, and governance-conversion loop in MAGE