Solo project description for ECE 50874 / 59595

Spring 2026

Profs. Davis & Torres-Arias

Summary and Ground rules

- 15% of grade
- Individual work
- Everyone does the same analysis, but you choose different subjects for analysis

What is the project about?

- Thesis: Real software engineering knowledge is largely learned from failure
- Most catastrophic failures are *systemic*, not coding bugs
- Post-hoc analysis can reveal assumptions invisible during development
- Study other failures to learn how not to do it yourself

Due dates

Week 3: Incident Selection & Feasibility
Week 7: Failure Reconstruction
Week 12: Failure Analysis & Responsibility
Finals Week: Mitigation & Ethics (and polishing)

Everything will use an Overleaf template.
→ Which will be shared with your Purdue email by Alex the TA

Let’s take a peek at the Overleaf template!

Deliverable 1 (~2 hours): Incident Selection & Feasibility Analysis

Purpose. Before committing to an incident, you must demonstrate that it is feasible to analyze at the required depth. This prevents late discovery that insufficient technical detail exists.

What to submit (1 page).

1. **Candidate incident description**
    - What system failed?
    - What harm resulted (human, environmental, safety, legal, or significant economic harm)?
2. **Source inventory**
    - List *at least two* technically substantive sources you plan to use (postmortem, investigation report, regulatory filing, technical blog, etc.).
    - You should also list at least 1 news article that describes this incident. Preferably more. These will give you a better sense of the public impact as well as the variations in analysis depending on analysis and availability of data.
3. **Technical depth assessment**
    - What software component(s) appear to have failed?
    - Is there enough publicly available information to model the system and reason about failure modes?
4. **Feasibility judgment**
    - Briefly argue that you can complete Deliverables 1–3 using available information.

Approval at this stage is required to proceed.

Deliverable 2 (~5 hours): Failure Reconstruction

Goal. Reconstruct *what happened* without yet proposing solutions.

Length. 3–4 pages.

Content.

- System overview
  - Architecture, major components, and stakeholders.
  - Should include at least 2 models: System architecture and Use case view
- Operating context
  - Normal operation and assumptions
  - This should include description of relevant background technology and business or government context, which may not be present in the technical sources initially identified in Deliverable 1.
- Failure timeline
  - Events leading up to the failure, during the incident, and immediate response
- Harm characterization
  - Who was harmed and how
- Initial causal hypotheses
  - Plausible technical and organizational contributors (no mitigations yet)

This deliverable should resemble the **descriptive half** of the Cloudflare chapter.

You are permitted to reuse analyses from the sources you reference, but all prose and drawings should be your own.

Deliverable 3 (~5 hours): Failure Analysis & Responsibility

Goal. Explain *why the failure occurred* using software engineering concepts.

Length. 3-4 pages.

Content.

- Failure modes
  - How the software behaved incorrectly
- Anticipatability
  - Was this failure mode foreseeable?
  - Was it discussed or documented prior to the incident?
- Swiss Cheese / multi-layer analysis
  - Technical, human, organizational, and process contributors
- Responsibility and agency
  - Who had the ability to prevent or mitigate the failure, and at what stage?

Requirement. Include at least **two diagrams** (1 swiss cheese, 1 fishbone). These may be slightly fictionalized depending on the available data, but you should be able to support any fiction you introduce.

This deliverable should resemble the **analytical core** of the Cloudflare chapter.

Deliverable 4 (~8 hours): Mitigation, Process, Ethics, and Polishing of Your Writeup

Goal. Practice foresight by proposing credible mitigations and reflecting on ethical responsibility.

Length. 3-5 pages.

Content.

- Mitigations
  - Technical changes
  - Process changes (e.g., safety lifecycle, CI/CD safeguards, validation)
  - Organizational or regulatory changes
- Mapping to standards (IEC 61508 and others covered in this course)
  - For example, IEC 61508 concepts such as hazard identification, safety requirements, and validation
- Reflection
  - Would your proposal have prevented or reduced harm?
- Ethics
  - Did engineers act unethically?
  - Or was the failure better explained by systemic constraints?

During this deliverable, you may find it necessary to revisit and modify your analyses during Deliverables 1-3 to ensure coherence and full information.

Where information is not available, you should be clear about its lack. Conjectures should be stated as such, and alternative explanations should be included.

Acceptable Sources

You must work from sources that contain technical detail sufficient to support modeling and analysis.

Recommended to look through these databases

- **Industry postmortems**
  - Cloudflare (primary reference example)
  - Google SRE postmortems
  - AWS Service Health Dashboard reports
- **Government investigations and disclosures**
  - **FDA** medical device recalls, safety communications, MAUDE reports
  - **SEC** cybersecurity disclosures, enforcement actions, Form 8-K filings
  - **NTSB** accident investigation reports and safety recommendations
- **Security and vulnerability analysis**
  - Google Project Zero blogs
  - Detailed CVE writeups with technical analysis

Alternative incident discovery sources

These sources may help you *identify* incidents, but you must also locate a substantive technical writeup.

- AI Incident Database
- RISKS Digest

Note that news articles alone are insufficient for this project, but you may be able to trace these to more technical writeups. However, I am not sanguine on this point.