**Colorado School of Mines — Field Session**

# Student Capstone Project Proposal

*Costpoint–Epsilon3 (CP-E3) Program Telemetry Pilot*

**Sponsor Organization:** ACME Satellite Manufacturing

**Industry Mentor / Point of Contact:** Ivan — Consulting Engineer, ACME Systems Integration

**Proposal Date:** August 12, 2026

**Submission Format:** This proposal is provided as a .docx file, consistent with the option to submit either .doc(x) or .pdf; it will be posted to students in .pdf format

## 1. Company Background

ACME Satellite Manufacturing designs and builds satellite subsystems and line-replaceable units (LRUs), including thrusters and power processing units (PPUs), for commercial and government aerospace customers. Manufacturing operations run on a Deltek Costpoint ERP system of record paired with an Epsilon3 manufacturing execution system (MES) that tracks work at the manufacturing-order (MO) and procedure-step level.

ACME's program management office (PMO) is responsible for tracking material availability and shop-floor progress across multiple simultaneous customer programs. Today that visibility is largely manual — built from spreadsheet reconciliation, walk-arounds on the shop floor, and word-of-mouth updates from production standups. As ACME's program count and contract complexity have grown, this manual approach has become a bottleneck for both day-to-day program management and for answering timely customer and executive questions about schedule and material status.

Over the past several months, ACME's integration engineering team has built the foundation this project extends: a working Costpoint–Epsilon3 (CP-E3) integration service that links ERP and MES data, including established program ↔ MO ↔ customer mapping and a Postgres data layer receiving structured, schema'd exports from Epsilon3. This capstone project is not a green-field effort — it builds directly on that existing, working integration layer.

## 2. Description of Work to Be Done

### 2.1 Problem Statement

ACME program managers currently lack a consolidated, trustworthy view of two things that matter most day to day: material availability (on-hand stock, PO status, lead times) and shop-floor status (what is causing delay — parts stuck in inspection, an MO waiting days on a quality sign-off, a slipped test date). Slips are often not communicated to PMs until they are already several days old, and the current process cannot easily answer the question a customer or executive is most likely to ask: 'how do you know, and when did that change?'

### 2.2 Guiding Philosophy

- By exception, not by browse — the goal is to tell a PM when something needs attention, not to hand them a dashboard to monitor. Step-level detail (an MO may have 50–60 steps) should never surface directly; only roll-ups at the MO / LRU / test level.

- No stepping on functional toes — an alert that fires inside a function's normal workflow tolerance is a false positive and costs the tool credibility. Alert thresholds must be agreed with production and quality leadership before going live.

## 2.3 Candidate Use Cases

Five candidate use cases have been scoped with ACME's PMO. Students will work with the sponsor to select an initial proof-of-point use case (US-03 is recommended as the lowest-risk starting point, since a comparable manual spreadsheet output already exists) and sequence the remainder based on data validation results during the course.

| ID | Use Case | Summary |
|---|---|---|
| US-01 | Stalled Work Exception Alert | Alert PMs when an MO/step shows no movement beyond an agreed dwell threshold (Epsilon). |
| US-02 | Test Schedule & Completion Visibility | Track scheduled/actual test dates and completion status per LRU (Epsilon). |
| US-03 | Automated Parts & PO Status Dashboard | Refresh parts/PO status on demand instead of manual report pulls (Costpoint). |
| US-04 | Kit Shortage Visibility | Show where and how often short-kitting occurs, at part level (Costpoint + Epsilon). |
| US-05 | Schedule Change Detection & 90-Day Trend | Detect schedule slips automatically; support 90-day trend queries. |

## 2.4 System Architecture — Existing Foundation and Pilot Layer

The diagram below shows how the student team's work sits on top of the existing, already-functioning CP-E3 integration layer. Students will not rebuild the ERP–MES connection; they will build the pilot application layer that consumes it.

| | | | | |
|---|---|---|---|---|
| **Costpoint (ERP) + Epsilon3 (MES)** | → | **CP-E3 Integration Service (FastAPI / Postgres)** | → | **Field Session Pilot Layer (Exceptions, Test View, Parts, Kits, Trends)** |

## 2.5 Agentic AI Component

ACME would like this pilot built as an agentic workflow rather than a fixed set of dashboards — students should treat the five use cases as modular skills an AI agent can invoke, not as five separate applications. This is a deliberate extension of ACME's existing internal work with local LLM deployment (an Azure-hosted Ollama/RAG stack already runs in another part of the organization), and follows an "Agent = Model + Harness" framing: a language model that reasons about intent, paired with a harness of tools, skills, and memory that constrains and grounds it against ACME's real data.

- Skills — each use case (stalled-work detection, test visibility, parts/PO status, kit shortage, schedule trend) is implemented as a discrete, independently testable skill the agent can select and invoke, rather than hard-coded screen logic. New skills should be addable without touching the orchestrator.
- Tools — the agent reaches Costpoint and Epsilon3 data only through defined, auditable tool calls (e.g., a query-snapshot tool, a threshold-lookup tool), never through free-form access — this preserves the

data-provenance and auditability expectations described throughout this section even as the interface becomes conversational.

- Memory — the agent must distinguish between short-lived conversational context (a PM's current question), configuration memory (agreed thresholds, tolerances, subscribed test types per PM), and the durable 90-day snapshot history that powers trend queries (US-05). These are different memory tiers with different retention rules, not one undifferentiated context window.
- A PM-facing agent should be able to answer natural-language questions such as "what changed on Program X since last week" or "what's short-kitting most often on the thruster line," grounded entirely in retrieved data and cited back to source system and timestamp — never generated from the model's general knowledge.

This is an enhancement layer on top of the core requirements in Sections 2.1–2.3, not a replacement for them. A working, non-agentic pilot that satisfies the underlying use case is still a successful outcome; the agentic framing is the preferred and encouraged approach, particularly for US-05's exploratory trend queries, where a natural-language interface is more useful than a fixed report.

| | | | | |
|---|---|---|---|---|
| **PM Query / Trigger (natural language or scheduled)** | → | **Agent Orchestrator (Model + Harness)** | → | **Skills Registry (Exception, Test, Parts, Kit, Trend skills)** |
| **Memory Layer (conversation state, threshold config, snapshot history)** | ↑ | **Tool-Calling Interface (Postgres queries, Costpoint / Epsilon3 reads)** | → | **Response to PM (exception, answer, or trend summary)** |

## 2.6 Scope of Student Work

- Validate which required data fields actually exist and are reliably populated in Costpoint and Epsilon3 before building against them (several open data-quality questions exist — see 2.7).
- Extend the existing Postgres data layer to support exception detection, trend snapshots, and roll-up queries.
- Build the pilot application layer for the selected use case(s): alerting logic, a parts/PO status view, and/or a schedule-trend query interface.
- Implement data provenance on every output — source system, extraction timestamp, and known gaps — so PMs never mistake a partial view for a complete one.
- Document the work so ACME's integration engineering team can continue development after the course ends.

## 2.7 Known Data Risks (Starting Points for Investigation)

- Whether Epsilon3 state-change timestamps are reliably populated (not backfilled or defaulted) — required for US-01.
- Whether Epsilon3 stores a scheduled test date, or only actuals — the single highest-risk assumption for US-02.
- Whether existing Costpoint PO and inventory report logic can be trusted, or needs to be validated against source tables — required for US-03.
- Whether a common MO key reliably joins kit-issue data across Costpoint and Epsilon3 — required for US-04.
- Snapshot cadence and the authoritative source of scheduled dates (Epsilon3 vs. the master production schedule) — required for US-05.

### 2.8 Deliverables

- A working pilot application for the selected first-proof-point use case(s), built on the existing CP-E3 data layer.
- An agent orchestration layer with at least one working skill invoked through defined tools, with clear memory-tier separation (conversational, configuration, and durable trend history).
- A data validation report identifying which required fields exist, are populated, and can be trusted.
- A configurable threshold/alerting framework (initial values may be placeholders pending leadership sign-off).
- Technical documentation covering architecture, data lineage, and setup/deployment.
- A final presentation and handoff to ACME's integration engineering team.

## 3. Desired Skill Set for Students

Preference will be given to students with some exposure to the areas below, but this is not a prerequisite — ACME expects and welcomes students learning new tools and technologies as part of the course, and the industry mentor will support that ramp-up.

- Python (the existing integration service is built in Python/FastAPI/SQLAlchemy)
- Basic SQL and relational database concepts (Postgres)
- REST APIs and working with structured/JSON data
- Exposure to LLM/agent frameworks and tool-calling patterns (e.g., function calling, agent orchestration, retrieval-augmented generation) is a strong plus given the agentic architecture described in Section 2.5 — but is explicitly not required going in; this is expected to be a learning area for the team.
- Comfort reading and reasoning about business/operational data (ERP, MES, or supply-chain concepts are a plus, not required)
- Basic front-end or dashboarding experience (e.g., a lightweight web UI or BI tool) is helpful for the presentation layer but can be learned during the course

## 4. Preferred Team Size

3–5 students. ACME prefers projects that can support at least three students, given the project spans data validation, backend/integration work, and a user-facing pilot layer — enough distinct workstreams to keep a team of three to five productively occupied.

## 5. Internship Potential

Yes — there is potential to extend an internship offer to one or more students at the conclusion of the course, contingent on performance during the project and ACME's hiring needs at that time. This is not a guarantee, but ACME views the Field Session team as a natural pipeline into its systems integration and program analytics work.

## 6. Location Where Work Should Be Performed

This project is designed to be performed remotely. The agentic architecture described in Section 2.5 is intentionally built to run against validated, appropriately-classified data extracts, which minimizes the need for students to be physically present on ACME's floor to make progress. ACME wants to be mindful of Fall session students' other academic responsibilities and has structured the engagement to keep on-site time to the minimum needed to establish data access and system context.

- On-site kickoff (single session) — system walkthrough, data access setup, and introductions to the integration engineering team.
- On-site presence otherwise limited to what's strictly necessary — e.g., access to systems or data that cannot practically be provisioned remotely, or the final demo/handoff, if ACME's team prefers that in person.
- All other work — development, data validation, skill/tool/memory implementation, and regular check-ins with the industry mentor — is expected to be done remotely.

Note: this proposal does not include compensation for commuting or mileage. Students should factor the limited on-site time described above into their planning.

## 7. Non-Disclosure Agreement

Yes — students assigned to this project will be asked to sign a non-disclosure agreement (NDA) before receiving access to ACME systems or data. This project involves ERP and MES data from an aerospace manufacturing environment, which may include program-specific, competitively sensitive, or export-controlled (CUI/ITAR) information depending on the customer program involved. ACME will provide the NDA for review prior to the start of the course and will scope any export-controlled data handling per applicable ACME and government requirements, separate from and in addition to the standard NDA.

## 8. Intellectual Property

Yes — ACME will ask students to assign intellectual property rights in work product created specifically for this project (including code, data models, documentation, and any pilot application built against ACME's Costpoint–Epsilon3 integration layer and related program data) to ACME. This is because the work is built directly on top of ACME's proprietary integration architecture and operates on ACME's internal business data; ACME needs to be able to continue, modify, and maintain that work after the course ends without an ownership question attached to it.

This assignment would be limited to project-specific deliverables. It would not extend to general skills, knowledge, or non-project-specific tools students bring to or develop outside the scope of this engagement. Exact terms will be documented in the same agreement covering the NDA, provided for student and Mines review before the course begins.

---

*Prepared by ACME Satellite Manufacturing for the Colorado School of Mines Field Session capstone program. Questions on this proposal may be directed to the industry mentor listed above.*