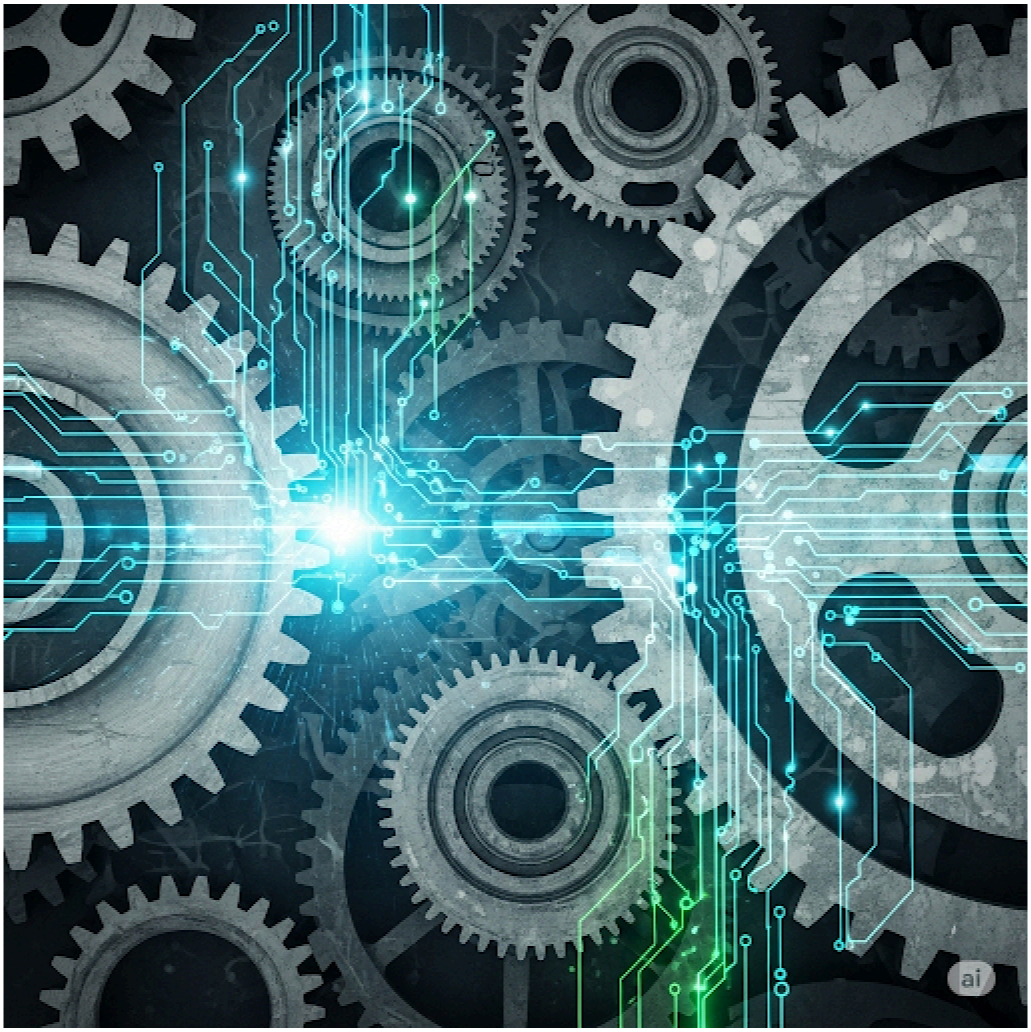
Project Proposal: Independent AI Integration Engineering Project

1. Executive Summary

Artificial Intelligence (AI) is a powerful tool with immense potential, but smaller manufacturing companies have been slow to adopt it due to practical barriers. In particular, the fragmentation of data and a lack of AI-related skills in the workforce have significantly impeded widespread AI implementation in smaller industrial firms (DesignNews, 2024). For example, 81% of companies cite siloed, fragmented data as a major reason for reluctance in adopting AI technologies, and the percentage of manufacturers planning to invest in AI fell from 93% in 2023 to just 53% in 2024 (DesignNews, 2024). Meanwhile, the AI technology itself is advancing at a breakneck pace. Global corporate investment in AI reached record highs (over $250 billion in 2024) and 78% of organizations reported using AI in 2024, up from 55% a year before (IBM, 2024; Stanford HAI, 2025). Generative AI in particular is booming, with forecasts projecting a $1.3 trillion market by 2032 (DesignNews, 2024). These advances are largely driven by massive R&D investments from big technology companies, resulting in ever-more capable AI models and tools. However, the focus of big tech has been on pushing the performance and capabilities of AI itself, rather than on integrating these new AI systems into legacy workflows. Often, new AI-powered platforms are built from the ground up (sometimes even reinventing existing functionalities) instead of tackling the messy challenge of compatibility with established industrial software and processes. This imbalance, rapid AI innovation versus sluggish adoption on the factory floor, is widening the gap between what AI can do and what is actually being used in operations. The value of bridging this gap is growing, as businesses that manage to integrate AI into existing processes stand to gain significant efficiency and competitive advantage. By extension, there is a growing need for engineers who understand both cutting-edge AI and the realities of industrial systems, and who can effectively integrate the two. This project proposes to cultivate exactly that skill set.



Proposal Overview: Instead of a traditional co-op, I propose a 13-month independent engineering project focused on AI integration in manufacturing. The goal is to develop the practical skills needed to apply state-of-the-art AI technologies as custom, high-value enhancements to existing industrial processes. In essence, this project will serve as a personal “co-op” in which I act as both engineer and researcher, learning by building. The project is structured with well-defined milestones and deliverables to ensure rigor comparable to a work placement. Under the mentorship of a faculty or industry sponsor, I will dedicate full-time effort to this project over the 13-month period, aiming to produce tangible results and documented learning. Key focus areas and outputs will include:

- AI Model Training and Adaptation: Develop and fine-tune AI models tailored to specific industrial tasks and datasets, focusing on performance, generalization, and efficiency.
- Data Engineering and Management: Build data pipelines to extract, clean, and structure fragmented industrial data for reliable AI usage.

- Robotics and Automation Fundamentals: Learn the hardware and software basics of automation systems relevant to modern manufacturing environments.
- AI-Physical System Integration: Create systems where AI software makes decisions that impact physical processes in real time.
- Middleware and Interface Development: Develop interface code that bridges AI systems with legacy industrial platforms and control software.
- Documentation: Produce documentation for a wide range of audiences and future collaborators.
- Tool Packaging and Distribution: Build and distribute general-purpose tools designed for reusability, usability, and ease of deployment.
- Open-Source Collaboration and Community Engagement: Collaborate with others to build, share, and improve open-source tools and ideas.
- Capstone System Integration: Deliver a final working system demonstrating the complete integration of AI with a physical or simulated process.

Overall, this independent project is designed to bridge the disconnect between cutting-edge AI and real-world manufacturing needs. By the end of the 13 months, I will have a portfolio of working integrations (software and hardware), a body of documentation/reports, and a final capstone system that demonstrates the integration of AI with a physical process. This experience will directly address the skills gap identified in industry.

---

2. Problem Statement

Despite rapid advances in AI capabilities, there is a clear disconnect between AI research and development and applied industrial systems. On one side, we see AI performance skyrocketing; for instance, researchers have dramatically improved benchmark scores in difficult tasks within just a year (Stanford HAI, 2025). Businesses are pouring record investments into AI, with private AI investment in the U.S. reaching $109 billion in 2024. In many domains, AI is no longer experimental, it's becoming embedded in everyday life, from autonomous vehicles to medical diagnostics (Stanford HAI, 2025). Yet on the other side, many companies (especially small and mid-sized manufacturers) struggle to translate these breakthroughs into their operations. There is evidence that after an initial hype cycle, industrial AI adoption has been slower than expected. For example, a recent industry report found that the proportion of manufacturers planning to increase AI spending dropped precipitously in the last year, from 93% to 53% (DesignNews, 2024). Another study noted that while 60% of manufacturers claim to be using AI, much of that use is confined to front-office functions like business analytics, rather than on the factory floor where the core operations happen (DesignNews, 2024). This suggests a hesitation or inability to integrate AI into production processes.

Two primary barriers contribute to this adoption gap. First is the fragmentation of data and systems in manufacturing environments. Industrial operations generate massive amounts of data (the manufacturing sector produces over 1,800 petabytes of data annually), but these data are often siloed across different machines, departments, or legacy software (DesignNews, 2024). A Cisco survey found that siloed data is a major reason why 81% of companies have been reluctant to adopt AI (DesignNews, 2024). In practical terms, an AI model is only as good as the data it can access. If quality data from production lines, QA systems, or supply chain systems cannot be aggregated and fed into the AI, the insights and automations AI promises simply don't materialize. Integrating AI often requires connecting to old PLCs or proprietary databases and cleaning inconsistent datasets, requiring the building of translation or filtering logic that many firms are not equipped to do.

The second barrier is the lack of AI-related skills in the current workforce. Introducing AI solutions calls for specialized knowledge in data science, machine learning, and software integration, skill sets that many traditional manufacturing IT and engineering teams do not yet possess (Markt-Pilot, n.d.). Smaller companies especially may not have dedicated data scientists or ML engineers on staff. Hiring such talent is competitive and expensive, and the existing engineers (experts in mechanical, electrical, or process engineering) may not have had exposure to AI in their training. A recent Thomson Reuters technology report likened the demand for AI/ML skills now to the rush for data science talent a few years ago, noting that many firms are struggling to find people who "can deliver results" with AI and have the "fabric of capabilities" needed for sustainable AI integration (Thomson Reuters, 2024). Amazon's Vice President of Data and AI, Swami Sivasubramanian, put it bluntly: "(AI) won't reach its full potential unless we really have the workforce ready to embrace it" (Thomson Reuters, 2024). In response, companies like Amazon, Microsoft, and IBM have launched major up-skilling initiatives to train their employees on AI (Thomson Reuters, 2024). This highlights that even the biggest companies recognize the skill gap problem. In smaller organizations, the gap is even more acute; they must either invest in training their existing staff, rely on external consultants, or risk falling behind as AI tools proliferate.

Compounding these issues is a cultural and structural disconnect: AI technology development vs. system integration. The cutting edge of AI is largely driven by research scientists and engineers at big tech firms or startups, whose primary goal is to push the envelope of what AI can do. They create new algorithms, larger models (e.g., Meta’s Llama series of large language models or OpenAI’s GPT series), and novel applications like OpenAI’s ChatGPT Agent that can browse the web or use tools autonomously (OpenAI, 2025; IBM, 2024). These innovators, however, are usually not focused on retrofitting their creations into legacy environments. They often build on modern tech stacks, assume cloud connectivity, and target users who are early adopters or tech-savvy.

This project is predicated on the belief that bridging this gap requires a new kind of engineer, one who understands the language of AI researchers and the practical reality of legacy systems. Today, if an organization wants to, say, deploy a computer vision system to detect product defects on a production line, they might struggle to find a single person or even a single team that has end-to-end expertise to do it. The task crosses multiple domains: data engineering (to gather and label images), machine learning (to train a vision model), software development (to integrate that model into a production workflow), and industrial engineering (to mount cameras, interface with existing QC systems, and trigger actuators for sorting rejects). It's rare to find all that expertise unified. Thus, companies either outsource to multiple vendors or not attempt it at all. This project aims to develop those cross-cutting skills.

In summary, the problem this project addresses is: How can we effectively connect the rapid advancements in AI technology with the immediate, practical needs of manufacturing systems? The ultimate thesis behind the project is that by focusing on integration, we can unlock significant untapped value.

---

3. Project Objectives

The overarching objective of this project is to develop a comprehensive set of skills and deliverables in AI integration engineering. These objectives cover technical proficiencies, practical deliverables, and the ability to work in a quasi-professional setting. The project is designed with multiple sub-goals that ensure I gain depth in specific areas while also learning to synthesize them into integrated solutions. Below are the key objectives:

- Train and Adapt AI Models: Fine-tune open-source models for specific tasks by preparing datasets, applying transfer learning, and evaluating performance on practical, real-world problems.
- Develop Middleware and Integration Layers: Build APIs, scripts, and adapters that bridge gaps between incompatible systems, translating between different tools, platforms, or protocols.
- Gain Hands-On Robotics Experience: Work directly with robotics systems, both hardware and software, to understand control, automation, and physical interaction in engineered systems.
- Design for Deployability: Focus on making systems usable, robust, and resilient, incorporating error handling, safety considerations, and a smooth user experience from the start.
- Build Cyber-Physical Systems: Integrate AI components with sensors, actuators, or control logic to create systems that observe, decide, and act in real time.
- Use Modern Development and Collaboration Tools: Build fluency with general-purpose tools used in technical projects, including version control, model hubs, visual programming environments, and cloud platforms.
- Execute with Engineering Discipline: Break down large goals into structured tasks, define milestones, and maintain a regular cadence of progress and review, treating the work with the rigor of a deliverable-driven project.

Together, these objectives form a cross-functional engineering program built around doing, not theorizing. By meeting them, I'll gain not just technical skills, but a demonstrable body of work (codebases, working systems, documentation, and presentations) that show I can deliver integrated AI solutions in real-world contexts.

3.5 Personal Development Objectives

This project is more than a technical challenge. It's also a chance to improve how I work, building better habits, communicating clearly, and becoming a more adaptable and collaborative developer.

1. Execution Discipline and Project Ownership
    - Work to Deadlines: Develop habits of working in structured cycles, delivering progress regularly instead of waiting for things to be "perfect." This builds momentum, better time management, and a consistent pace.
    - Adjust Plans with Purpose: Learn to assess progress honestly and update timelines as needed. The goal isn't to follow a rigid schedule, but to stay in control when conditions change.
    - Maintain and Improve Tools: Go beyond building something once. Revisit tools, incorporate feedback, and refine usability through iteration.
2. Communication and Accessibility
    - Design for a Range of Users: Build tools and documentation with layered access in mind.
        i. Hands-off users want a tool that just works: clear steps, smart defaults, no friction.
        ii. Curious users start simple but want to explore; they need walkthroughs and explanations that don't assume deep expertise.
        iii. Developers or power users want to extend the system; they need code access, documentation, and clarity.
    - This approach helps me think critically about how different people experience the same system and how to make that experience work at every level.
    - Prioritize Accessibility and User Experience: As someone with dyslexia, I know how small design choices, like missing spell check or awkward file paths, can become real barriers. I'll focus on building tools that are easy to use from the start, especially in environments that usually assume technical knowledge.

    - Curate a Public Portfolio: Present my work as a cohesive body of code, demos, documentation, and design decisions, collected into a portfolio that shows what I've done and how I work.

3. Collaboration and Community Integration
    - Build Experience with Collaborative Tools: Use platforms like GitHub to manage code, track progress, and document work with structure and clarity.
    - Participate in Developer Communities: Treat dev forums, Discords, and open-source platforms as active workspaces. I'll contribute, ask questions, and engage with others as part of the development process.
4. Long-Term Mindset
    - This project is a chance to level up how I solve problems. I want to manage complexity, finish things without being chased by deadlines, build tools that are actually usable, and keep my work clear and maintainable.

---

4. Scope

This 13-month project covers a broad set of technical skills, unified by a single focus: integrating AI into real-world systems through practical engineering. The aim is to explore and execute across the full pipeline, from model training and data prep to physical implementation and tool packaging, while maintaining clarity, usability, and relevance.

In Scope

- Core Engineering Activities:
    - Adapting and applying AI models to specific use cases
    - Creating, curating, and preparing data sets for integration
    - Programming and deploying simple robotic behaviors
    - Connecting AI software to physical systems (sensors, actuators, microcontrollers)
- Tool and System Development:
    - Building software tools with clear interfaces and reusability in mind
    - Prioritizing maintainable code, documentation, and packaging
    - Designing and implementing physical components using methods like 3D printing
- Communication and Documentation:

    - Creating layered documentation for technical and non-technical audiences
    - Publishing work in accessible formats (e.g., GitHub repos, project site)
    - Tracking progress, decisions, and lessons learned throughout the process

Out of Scope This project does not cover:

- Pure research into AI algorithms or model architecture development
- Full-scale industrial deployments requiring enterprise-level infrastructure
- Complex robotics platforms beyond what can be built or simulated at a small scale
- Automated systems that involve safety-critical processes or regulatory constraints

Due to the expansive nature of applied AI, managing scope will be an ongoing challenge. Section 8 outlines combative strategies and mitigating factors. Any promising but out-of-scope directions will be logged for future development after the core project is complete.

This integrated scope, spanning AI, physical systems, development tooling, and communication, is ambitious but manageable within 13 months. Defining these boundaries upfront keeps the project focused and execution-ready.

5. Project Plan

This 13-month independent engineering project (September 1, 2025 through September 30, 2026) is structured to develop practical, applied skills in integrating modern AI technologies into existing industrial systems. The work is divided into three main phases, AI software, robotics hardware, and final system integration, with the final phase tying all the skills gained together in one project.

An unphased 13th month exists as a buffer. With long timelines and shifting technology, this allows for unexpected delays, rapid tool changes, or last-mile polishing without derailing the rest of the plan.

Alongside the three main phases, the following work will continue throughout the project:

- Documentation and Writing: Documentation captures decisions, trade-offs, and system behavior, turning work into something transferable. This supports both Community Integration and the objective to Design for Deployability, making it easier for others (and future me) to use and improve the systems I build.

- Tool Development: Developing project-specific tools reinforces the objective to Build Cyber-Physical Systems by creating the infrastructure needed to train models, automate workflows, and interface with hardware. These tools also support Execution Discipline by keeping work modular and scalable.
- Presentations and Communication: Regular presentations align with the objective to Execute with Engineering Discipline, enforcing iteration and review. They also serve Communication and Accessibility goals by making technical progress understandable at multiple levels of depth.
- Sponsor Collaboration: This supports the Project Ownership and Professional Standards components of the plan. Regular sponsor feedback ensures scope control, and treating this as a quasi-professional project adds rigor that mimics real-world accountability. The Sponsor’s role is defined in Appendix 1.
- Tool Maintenance: Maintenance aligns with a Long-Term Mindset and Deployability, treating tools as evolving systems instead of one-off proofs-of-concept.

What follows outlines the phased structure of the project and the key technical focus areas in each. Deliverable details are covered in section 6.

Phase 1: AI & Software (Months 1-4)

This phase focuses on building a foundation in AI model behavior and software tooling. Each learning area is selected to develop both theoretical understanding and practical implementation skills that will be essential in later integration work.

- Fine-tuning: Modify pre-trained models to perform specific tasks using custom or curated datasets. This is the cornerstone of applied AI. Rather than building models from scratch, I’ll learn how to adapt existing ones for specific problems, making them useful in niche or resource-limited scenarios. This also builds skill in dataset handling, overfitting management, and performance evaluation, all of which will be critical when models move from a sandbox to real-world systems.
- Quantization: Compress models for faster inference on low-power or embedded devices. Later phases of the project will involve deploying models onto hardware with limited compute. Quantization lets me prepare for that constraint by learning how to shrink models without destroying their performance. It’s also an intro to model efficiency and trade-off decisions between accuracy and speed.
- Model merging: Combine the strengths of multiple models to create task-specific hybrids. Merging teaches how to reuse trained systems creatively. Whether for ensembling, capability fusion, or experimentation, this technique is a way to blend solutions instead of reinventing them. It also gives insight into how fluid, or non-fluid, a model's characteristics are.

- Embedding models: Train or adapt models that generate semantic embeddings for search, classification, or similarity. Embeddings are how models turn complex data into something measurable. I'll need this skill to support tasks like anomaly detection, similarity search, or clustering, where raw input isn't directly labelable but still needs intelligent grouping or tagging.

- Dataset synthesis: Generate artificial datasets where real data is limited or sensitive. In many use cases, especially early prototypes, real-world data won't exist or be accessible. Synthesizing data lets me build and test systems under controlled conditions. This also teaches how to simulate variability, edge cases, and input noise, which is valuable in robotics and manufacturing scenarios.

- Data cleaning: Develop pipelines to prepare and sanitize input data for model training or inference. Most of the engineering work in AI is data wrangling. This objective reinforces that by teaching how to handle corrupted files, missing values, inconsistent labeling, and messy formats. It's also the gateway to automation, cleaning once manually, then building reusable preprocessing tools.

- Tokenization: Explore and implement tokenization strategies that affect model performance. Understanding tokenization helps demystify what language models "see" when they read inputs. It's especially important when using or modifying models for specialized domains, where default tokenization might fail. This topic also touches on encoding limits and model compatibility.

- Attention mechanisms: Study and experiment with transformer components to understand what drives model behavior. This gives insight into how modern models prioritize information. Rather than treating models as black boxes, I'll learn to read and tweak their inner workings, which is helpful for debugging behavior, improving reliability, and understanding why a model makes certain predictions.

Phase 2: Robotics & Hardware (Months 5-8)

This phase shifts the focus from software to physical systems. The goal is to build functional, programmable hardware components that can operate independently or interface with AI tools developed in Phase 1. These learning areas cover the fundamental skills needed to build real-world, AI-ready hardware systems.

- Microcontrollers: Program low-level control logic using Arduino or similar platforms. Microcontrollers serve as the interface between digital logic and physical behavior. Learning how to program and communicate with these systems is essential for building responsive devices, whether for sensor data collection, motor control, or basic automation tasks. This builds the foundation for embedded intelligence in later phases.

- Stepper motors: Control precise, incremental movement in physical systems. Stepper motors offer fine-grained control over motion, which is crucial for building positioning systems like gantries, conveyors, or robotic arms. This topic introduces motion planning, power management, and synchronization, which are core concepts in any automated hardware system.

- Servo motors: Calibrate and use servos for more flexible mechanical actuation. Servos allow smooth, controlled movements for use cases like object manipulation, sorting, or camera alignment. Unlike steppers, they require feedback tuning and angle-based control, teaching key lessons in analog-to-digital interfacing and actuator feedback loops.

- Camera modules: Capture real-time visual data to feed into AI models. This is the bridge between the physical and digital world. Getting live video or stills into a model pipeline is the foundation for vision-based AI. It also introduces challenges like frame synchronization, image quality control, and latency, which are real problems in real systems.

- Object tracking: Implement basic CV pipelines to track and identify objects in motion. Tracking allows AI systems to engage with dynamic environments. Learning to detect, follow, and interpret object movement forms the base of applications like sorting, inspection, or robotic response. It also creates a meaningful use case for integrating vision tools from Phase 1.

- Actuator calibration: Tune mechanical systems to behave predictably under AI guidance. Calibrating actuators ensures that the system responds accurately to control signals, especially when the AI layer is deciding those signals. This teaches how to close the gap between theoretical control and physical reliability, which is critical for building systems that don't just "run," but run well.

- 3D printing: Design, print, and modify hardware components for sensors, mounts, or robot enclosures. Rapid prototyping enables fast iteration of mechanical components tailored to specific tasks. Whether mounting a camera, housing a motor, or designing an arm extension, 3D printing gives control over the physical structure of the system and ties into the practical realities of integration and testing.

Phase 3: Integration & Final Project (Months 9-12)

This phase brings together the AI and hardware developed in earlier stages into a unified, working system. The goal is to create an autonomous setup that reacts to its environment, processes data in real time, and takes meaningful actions. Everything up to this point feeds into building something coherent, responsive, and durable.

- Control systems: Integrate AI into the logic that governs robotic motion or response. Controlling physical devices with AI outputs requires blending intelligence with precision. This means pairing perception with motion, translating predictions or classifications into physical actions, with attention to timing, constraints, and synchronization.

- Feedback loops: Build systems where outputs from sensors or actuators influence the next round of decision-making. Rather than just reacting once, the system should adapt based on its own behavior and results. Feedback loops create the foundation for learning, adjustment, and stability, whether that means re-centering a camera, retrying a grasp, or modifying behavior based on input changes.
- Real-time inference: Ensure that model predictions happen fast enough to guide hardware decisions as they occur. Inference that arrives too late is useless in a live environment. The focus here is on optimizing models and pipelines to deliver decisions quickly enough to be actionable, especially when running on edge devices with limited compute.
- Model deployment: Transition models from development into real-world environments with all the constraints that entails. Packaging a model so that it actually runs outside of controlled notebooks and simulated inputs is a nontrivial skill. This includes model conversion, dependency management, and testing in production-like conditions.
- Agent interface: Let the user control the system through a conversational interface powered by AI. Instead of relying on fixed buttons, sliders, or scripted interfaces, this phase explores natural language control. The goal is to allow a user to give verbal or text-based commands, such as "sort red objects" or "check alignment," and have the system interpret and act on them.
- Tool selection: Build logic for the robot to choose which sensors or tools to use depending on the task. Real-world systems must prioritize and adapt. This involves designing a structure where the system decides whether to use visual input, physical probes, or internal logic depending on context. It mirrors how a human decides whether to look, feel, or test something before acting.
- Error handling: Build resilience into every layer of the system so it can recover, retry, or adapt when things go wrong. Unexpected inputs, hardware misfires, or model misclassifications will happen. This section focuses on catching those events, reacting intelligently, and ensuring that the system remains safe, stable, and usable through failure.

6. Deliverables

This project will produce a wide range of deliverables across three categories: technical systems and tools, documentation, and presentations/communication.

Deliverables are structured to ensure transparency, rigor, and a continual record of progress. They will be shared in both private and public formats (e.g., GitHub, portfolio site), with formal and informal checkpoints built into the schedule.

Major Milestone Deliverables These are the final or phase-end artifacts that capture the core outcomes of the project:

- Fully Integrated AI-Robotics System as Final Project: A complete system demonstrating real-time decision-making and actuation. It includes trained models, embedded logic, and hardware control.
- Phase-End Reports (3 total): One at the end of each main phase. These reports will include design decisions, technical challenges, workarounds, and lessons learned. They will also cover what was deprioritized or cut.
- Final Portfolio Website: A publicly viewable showcase of all major work, containing all work done throughout the project and built progressively.

Recurring Progress Deliverables These deliverables are produced on a regular cadence to track development and support accountability:

- Monthly Presentations: A 30-60 minute formal presentation with a slide deck covering milestones, metrics, blockers, and next steps. Open to feedback and peer review.
- Bi-Weekly Progress Papers: A 1-2 page informal status report summarizing completed tasks, current focus, short-term plans, and any blockers. Includes visuals where relevant.
- Weekly Sponsor Check-Ins: Verbal updates (except on presentation weeks) covering current progress, questions, reflections, and scope alignment.

Tool Development Deliverables Tool-building is treated as a formal deliverable stream throughout Phases 1 and 2:

- Two Tools per Month: New or improved software tools that assist with AI training, automation, integration, etc.
- Each Tool Includes:
  - Executable Package: An installable codebase (e.g., pip package, Docker container, or script with dependencies clearly documented).
  - GitHub Repository: With versioned code, issue tracker, and commit history showing development over time.
  - User Guide: A clear README explaining usage, parameters, and expected output, with examples or visual aids.

Purpose of Deliverables Deliverables serve three key functions:

1. Execution Discipline: Forces regular progress checkpoints and prevents last-minute rushes.
2. Transparency: Sponsors and reviewers can see the work evolve in real-time.
3. Portfolio Value: Creates a body of work (codebases, system demos, documentation, and tooling) that can be reused, extended, or showcased beyond the academic context.

---

8. Risk Assessment and Mitigation

This project carries the expected risks of solo technical execution: complexity, time, and burnout. Here's how I'll handle them.

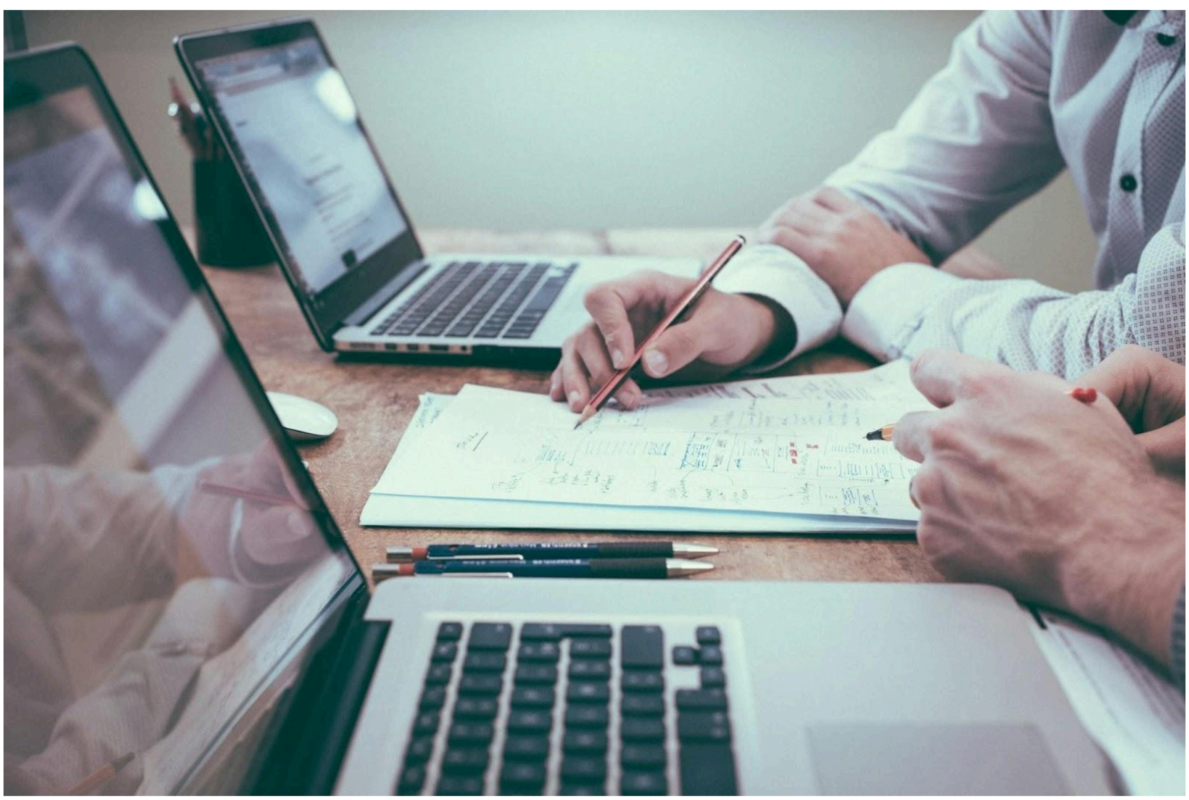
- Technical Risk - Integration Complexity: Bringing AI and robotics together introduces real fragility, including synchronization delays, sensor failures, and control feedback chaos.

These aren't "solved" problems, just managed. My general strategy is to reduce interdependencies: design clean interfaces between subsystems, test those interfaces in isolation, and try to integrate only when each part is stable. This makes failures easier to isolate and debug. This mitigation strategy and the one-month buffer help alleviate the effects of issues.

- Timeline Risk - Slippage and Misallocation: Long timelines give a false sense of slack. This project runs on a fixed set of deliverables with non-negotiable monthly presentations. If something's not ready, it still gets shown; that pressure keeps pace steady. I've also built in a 13th-month buffer and set milestone reviews that allow cutting or scaling features as needed. If something runs long, it has to justify itself or get cut.

- Personal Risk - Burnout: This project is flexible by design. If I burn out on tuning models, I can 3D print mounts. If hardware gets frustrating, I can rewrite docs or refactor tools. That variability keeps things fresh. Regular deliverables provide momentum and prevent the dead time that usually breeds burnout. If morale dips, I shift tasks instead of pushing through walls.

Overall Strategy: This project is ambitious at some points but not brittle. Risk is managed through modular design, progress tracking, and flexibility in execution. Problems are expected, but the structure is able to handle them.



9. Viability

This section establishes that the proposed project is not theoretical or speculative; it is feasible, educationally valid, and aligned with current industry demands. It draws on prior work, live trends, and structural alignment with academic expectations to support that claim.

Personal Proof-of-Concept This project doesn't start from zero, it scales up from work already in motion. Over the past month and a half, I've been independently exploring the core themes of this project: learning how AI works, building tools, refining them for usability, and documenting the process. These efforts validate both technical capability and workflow fit. Appendix 2 contains a fully packaged software tool, an informal update paper, and reflection writings. Together, these show that the systems proposed in this project aren't aspirational, they're extensions of work I'm already doing. The learning curve is underway, and the methodology is proven at a small scale.

Industry Integration AI isn't a future trend, it's already adopted by most major companies. In 2024, 78% of organizations reported using AI, but a consistent problem remains: implementation (Stanford HAI, 2025). The tools exist, but integrating them into meaningful workflows, especially those involving physical systems, is the choke point. This project is designed around that exact friction point. It focuses not on inventing new AI models, but on building systems that turn AI outputs into usable, physical action. That's the skill gap most organizations actually care about: taking models out of notebooks and into production.

AI + Robotics Integration Trend We are witnessing the dawn of smarter robots in industries beyond just programming repeatable tasks. A prime example is Miso Robotics' "Flippy" robot, which is essentially an AI-powered robotic fry cook. Flippy's success (being deployed in White Castle and other kitchens) validates that adding AI (computer vision and machine learning) to robots performing mundane tasks can bring real value in cost savings and consistency (Miso Robotics, n.d.). Flippy is explicitly designed for "quick and hassle-free integration" into existing kitchen setups, showing that a key selling point was its ability to drop into a legacy environment (a fast-food kitchen) without requiring a complete overhaul (Miso Robotics, n.d.). This directly parallels my project's ethos to build AI enhancements that integrate rather than replace wholesale.

Another striking development was recently reported from China: the first autonomous AI robot football match where teams of humanoid robots played soccer fully controlled by AI (The Guardian, 2024). While the robots "stumbled" and were not about to beat human teams, experts noted it's impressive how year-over-year these AI-physical integrations improve (The Guardian, 2024).

Educational and Accreditation Validation From an academic standpoint, replacing a co-op with a project is unusual, so I sought to validate that it can still meet educational standards. Looking at peer institutions, Stevens Institute's CS program explicitly allows an "individual capstone project working on real research projects solving real world problems," which is essentially what I'm doing, just earlier in my curriculum (Stevens Institute of Technology, n.d.). Bob Jones University's engineering program requires students to complete "a solo design project" in addition to team projects (Bob Jones University, n.d.). Both programs are ABET-accredited. This validates that a well-structured individual project can fulfill learning outcomes comparable to a team co-op or capstone, as long as it's comprehensive. I've incorporated elements (like community interaction for teamwork and rigorous documentation for communication) to ensure this project does the same.

References

- ABET. (n.d.). ABET criterion 3: Student outcomes. University of Colorado Boulder. Retrieved August 1, 2025, from https://www.colorado.edu/ecee/abet-criterion-3-student-outcomes

- ABET. (n.d.). Student outcomes. Cedarville University. Retrieved August 1, 2025, from https://www.cedarville.edu/academic-schools-and-departments/engineering-and-computer-science/student-outcomes

- Andresen, J. E. (2025, July 15). Overcome siloed and fragmented data - just in time for AI. IndyKite. Retrieved August 1, 2025, from https://www.indykite.com/blogs/overcome-siloed-and-fragmented-data-just-in-time-for-ai

- Bob Jones University. (n.d.). Engineering, BSE. Retrieved August 1, 2025, from https://www.bju.edu/academics/programs/engineering/

- Center for the Development and Application of Internet of Things Technologies (CDAIT). (2023, July 15). Digital transformation through IoT technologies. Georgia Institute of Technology. Retrieved August 1, 2025, from https://cdait.gatech.edu/sites/default/files/2023-07/Digital_Transformation_Through_IoT_Technologies_July_15_2023.pdf

- Spiegel, R. (2024, July 23). AI Makes Slow but Sure Progress in Manufacturing. *Design News*. Retrieved August 1, 2025, from https://www.designnews.com/automation/ai-makes-slow-but-sure-progress-in-manufacturing

- Cisco. (2023, November 7). Cisco launches new research highlighting seismic gap in companies' preparedness for AI. Cisco Newsroom. Retrieved August 1, 2025, from https://investor.cisco.com/news/news-details/2023/Cisco-Launches-New-Research-Highlighting-Seismic-Gap-in-Companies-Preparedness-for-AI/default.aspx

- Dangelo, M. (2023, December 6). Needed AI skills facing unknown regulations and advancements. Thomson Reuters. Retrieved August 1, 2025, from https://www.thomsonreuters.com/en-us/posts/technology/needed-ai-skills/

- Derrick, M. (2025, July 7). AI football robots compete in first autonomous tournament. Technology Magazine. Retrieved August 1, 2025, from https://technologymagazine.com/news/ai-football-robots-the-worlds-first-autonomous-tournament
- IBM. (2024, September 25). Meta Llama 3.2 models, including new multimodal and lightweight versions, are now available in IBM watsonx.ai. IBM Think. Retrieved August 1, 2025, from https://www.ibm.com/think/news/meta-llama-3-2-models
- KPMG. (2021). Thriving in an AI world: A survey of business leaders on their artificial intelligence plans. Retrieved August 1, 2025, from https://assets.kpmg.com/content/dam/kpmg/es/pdf/2021/04/thriving-ai-world-2021.pdf
- Markt-Pilot. (n.d.). Use of AI in machine manufacturing: Key barriers to implementation. Retrieved August 1, 2025, from https://www.markt-pilot.com/en/resources/blog/barriers-ai-in-machine-manufacturing
- Mauran, C. (2024, August 29). OpenAI announces ChatGPT agent for web browsing. Mashable. Retrieved August 1, 2025, from https://mashable.com/article/openai-announces-chatgpt-agent-web-browsing
- Miso Robotics. (n.d.). Flippy. Retrieved August 1, 2025, from https://misorobotics.com/flippy/
- OpenAI. (2024, August 29). Introducing ChatGPT agent. Retrieved August 1, 2025, from https://openai.com/index/introducing-chatgpt-agent/
- Stanford University Human-Centered Artificial Intelligence (HAI). (2025, April 7). Artificial intelligence index report 2025. Retrieved August 1, 2025, from https://hai.stanford.edu/ai-index/2025-ai-index-report
- Stevens Institute of Technology. (n.d.). Computer science, B.S.. Retrieved August 1, 2025, from https://www.stevens.edu/program/computer-science-bachelor-degree
- The Guardian. (2024, June 29). Humanoid footballers stumble through their first tournament in China [Video]. YouTube. Retrieved August 1, 2025, from https://www.youtube.com/watch?v=W9KCnSUCHbo
- Warren, Z. (2025, July 24). Bridging the AI gap: How professionals can turn awareness into action. Thomson Reuters. Retrieved August 1, 2025, from https://www.thomsonreuters.com/en-us/posts/technology/bridging-ai-gap/