Cases - 3rd AI Hackathon - Apr 17-19, 2026

G.E.A.R. x Danfoss @ SDU Sønderborg

The 3rd AI Hackathon has6 cases for students to choose from. At least 1 student team should work on each case.

CASE TEMPLATE

Case 1: [Title]

Case description: [Describe the case – think 3-10 lines]

Challenge Area: [Describe briefly the areas involved]

Danfoss Context: [Describe the context / background for the use case]

The Challenge: [Describe the overall challenge]

Your Mission: [Describe how you envision the solution]

Potential Tools: [Describe the relevant tools and software programs]

Potential Skills: [Describe the skills, the team needs to solve the challenge]

Provided Data: [Describe the data sets, that will be made available to the students – if any]

Deliverable(s):

1. [Deliverable 1]
2. [Deliverable 2 …etc]

Success Looks Like: [Describe what the solution should be capable of achieving/outputting]

Case Owner: [Case Owner name + Segment / BU]

Case 1: AI-Driven Hazard Analysis, Risk Prediction & Safety Dashboards

Case Description

Imagine your code helping prevent workplace accidents and making a real difference for thousands of people. At Danfoss, we believe data - driven safety can save lives - and we need your creativity and skills to make it happen!

Step into the mission of helping Danfoss identify risks *before* accidents occur, thereby making workplaces safer by using real hazard reports from across the globe. Today, this data is multilingual, unstructured, and under-utilized, limiting its potential to drive proactive safety improvements. The challenge is to translate, process, and analyse this data to identify risk patterns, predict high-risk areas, and generate actionable safety plans – all visualised through dashboards.

Your mission is to turn this complex and messy data into smart safety insights and compelling dashboards that support the creation of concrete safety roadmaps.

Challenge Area

- Translating and cleaning up multilingual safety reports
- Using AI and data science to spot patterns and predict risks
- Building interactive dashboards and translate this into actionable safety roadmaps

Danfoss Context

Danfoss aims to leverage its hazard report data to proactively improve workplace safety. Reports are currently siloed, multilingual, and unstructured, making it difficult to extract insights and drive safety improvements. The project supports Danfoss's commitment to data-driven safety management and continuous improvement.

The Challenge

How can we systematically translate, classify, and analyse hazard reports to predict risks and recommend targeted safety improvements, while ensuring all outputs are ready for visualization and even further to actionable roadmaps?

Your Mission

Design and implement a end-to-end data solution that:

- Translate all safety reports into English
- Process & categorize hazards (e.g., electrical, chemical, slips/trips)

- Calculates risk scores and trends
- Build predictive models to identify high-risk areas and prevent future accidents
- Recommend prioritized safety improvements at Segment, Division, or site level
- Prepare everything for easy visualization (e.g. Power BI or similar dashboards)
- Create prioritized specific actions and roadmaps

Potential Tools

- Python (for coding and data analysis)
- NLP libraries (spaCy, NLTK, transformers)
- Translation APIs (Google Translate, DeepL, etc.)
- scikit-learn (machine learning)
- Pandas (data manipulation)
- SQL (databases)
- Power BI or similar (dashboarding & advanced features)

Potential Skills

- Data preprocessing, cleaning and translation
- Text analysis and classification
- Machine learning and predictive modelling
- Data visualization and dashboard design

Provided Data

- Structured fields: case number, date, location, entity, reported risk potential/severity if available, case classification, case activity, root cause (in excel file)
- Unstructured fields: free-text titles, descriptions & actions (in multiple languages)
- Danfoss Risk model (formula) & other relevant Danfoss standards
- Possible hazards categories

Deliverables

1. Clean, translated dataset (CSV or database)
2. Hazard classification model and performance report
3. Risk scores calculations and trend analysis
4. Predictive model(s) with explanation
5. Safety improvement recommendations (report)
6. Power BI-ready dataset and dashboard or alternative visualization
7. Data dictionary and schema
8. Specific actions and roadmap, linked to risk insights, predictions, and relevant safety standards

Success Looks Like

- All reports are translated and categorized accurately and method used is available for future analysis (new set of data yearly)
- Robust risk scoring and trend identification
- Predictive models highlighting risk areas to be mitigated to avoid future accidents
- Actionable, prioritized safety recommendations
- Well-structured dashboards, Power BI-compatible datasets or similar
- Insights are not only visualized, but also translated into prioritized, specific actions and roadmap
- Clear documentation and reproducible code.

Ready to make a real impact? Join the Danfoss Safety Dashboard Hackathon and help us build a safer future with AI!

Case Owner

Mirela Suzana Oprea, EHS Director, DCS

Case 2: AI Agent for Converting ROS2 Drivers to MISRA C Compliant Embedded C

Brief description

Sensor manufacturers typically provide ROS2 drivers written in C++ for their hardware (LiDAR, cameras, IMUs, etc.). These drivers rely heavily on ROS2 middleware, C++ standard library features, dynamic memory allocation, and other constructs that are unsuitable for safety-critical embedded systems. When integrating these sensors into embedded platforms, engineers must manually rewrite the driver logic in plain C — stripping out ROS2 dependencies, replacing C++ constructs, and ensuring compliance with MISRA C coding standards. This manual translation is time-consuming, error-prone, and must be repeated for every new sensor. The goal is to build an AI agent that can automatically convert a manufacturer-provided ROS2 driver into MISRA C compliant embedded C code.

Challenge Area

AI/LLM-based Code Generation, Embedded Systems, ROS2, MISRA C Compliance, Safety-Critical Software

Danfoss Context

Danfoss develops autonomous machines that integrate sensors from various manufacturers. These manufacturers provide ROS2 drivers written in C++ that handle sensor communication, data parsing, and configuration. However, Danfoss embedded platforms require plain C code that is MISRA C compliant — no C++ features, no ROS2 dependencies, and strict adherence to safety coding standards. Currently, engineers manually study each ROS2 driver to extract the core communication protocol and data parsing logic, then rewrite everything in embedded C. This is a repeatable, pattern-based task that is a strong candidate for AI automation. The agent must be sensor-agnostic: it should work with LiDARs, cameras, IMUs, radar, ultrasonic sensors, or any other hardware whose manufacturer supplies a ROS2 C++ driver.

Your Mission

Build an AI agent (using LLMs or other AI techniques) that can analyze any manufacturer-provided ROS2 sensor driver source code and automatically generate MISRA C compliant embedded C code:

- Analyze any ROS2 driver package to automatically detect the sensor type, communication protocol (TCP, UDP, SPI, I^2C, serial, USB, etc.), data parsing logic, and configuration parameters

- Identify and remove all ROS2 dependencies (publishers, subscribers, services, parameters, launch system)
- Convert C++ constructs (classes, templates, STL containers, exceptions, RAII) to plain C equivalents
- Ensure the generated C code is MISRA C compliant (no undefined behavior, no prohibited constructs, proper type safety)
- Produce well-structured, modular embedded C source and header files with clear interfaces between modules
- Generate a sensor descriptor header (sensor_config.h) that captures sensor-specific constants so the same generic driver framework can be parameterized for different sensors

Potential Data & Tools

- LLM APIs (OpenAI GPT-4, Anthropic Claude, open-source models)
- Python for agent orchestration and code generation pipelines
- Code analysis tools (tree-sitter, AST parsers, clang-tidy)
- MISRA C static analysis tools (Cppcheck with MISRA addon, PC-lint, Polyspace)
- C compiler with strict warnings (GCC -Wall -Wextra -pedantic, or ARM GCC for cross-compilation)
- ROS2 development environment (for understanding driver structure)

Potential Skills

- LLM prompt engineering and agentic AI workflows
- C and C++ programming
- ROS2 architecture (topics, messages, launch files, parameters)
- MISRA C rules and safety-critical embedded coding standards
- Embedded systems programming (static memory, bare-metal or RTOS patterns)
- Code generation and automated refactoring techniques.

Deliverable(s)

1. A working, sensor-agnostic AI agent that accepts any ROS2 driver package as input and produces MISRA C compliant embedded C code as output
2. Generated embedded C source and header files for at least two different sensor types (e.g., a LiDAR and an IMU, or a LiDAR and a camera) demonstrating the agent's generality — all without ROS2 or C++ dependencies
3. A MISRA C compliance report (from a static analysis tool) showing the compliance level of the generated code for each sensor

4. A concise presentation covering:
    1. Your AI agent architecture and sensor-agnostic approach
    2. How the agent detects sensor type and adapts its conversion strategy
    3. How ROS2/C++ constructs were mapped to embedded C equivalents
    4. MISRA C compliance strategy and results
    5. Demonstration of the generated code for multiple sensor types
    6. Limitations and potential improvements

Success Looks Like

- The AI agent generates syntactically correct C code that compiles without errors or warnings under strict compiler settings
- Generated code has zero ROS2 or C++ dependencies — pure embedded C
- Generated code passes MISRA C static analysis with minimal or no violations
- The core sensor communication logic (protocol handling, data parsing) is functionally preserved from the original ROS2 driver
- Generated code outputs sensor data in the Danfoss coordinate system (X forward, Y left, Z up) where applicable — verified by comparing against known reference data
- Generated code is modular with clean interfaces between protocol, parsing, coordinate transformation, and configuration modules
- The agent demonstrates generalizability — it successfully converts ROS2 drivers from multiple sensor manufacturers and sensor types without sensor-specific hardcoding

Case Owner:

Mohammad Kapadiya
Danfoss Power Solutions – Platform Software

Case 3: Predictive Manufacturing - Optimizing Production Capacity & Supply Chain Sourcing

Case description

Our global manufacturing network faces complex scheduling and sourcing challenges across multiple factories, work centers, and tools. We need to accurately predict production capacity and raw material requirements (e.g., metal coils, specific rubber compounds for gaskets) based on future project pipelines.

Currently, it is difficult to map unapproved sales projects with varying probability ratios against our actual production constraints. Cycle times for pressing components vary across different factories depending on machine pressure and tool characteristics, and one tool can produce multiple component types.

Additionally, we must factor in mandatory tool maintenance and downtime.

The challenge is to build an AI model that synthesizes these variables—sales pipelines, material master data, factory-specific processing times, and tool downtime—to simulate production scenarios, predict capacity bottlenecks, and optimize the timely sourcing of correct raw materials.

Challenge Area

AI modelling, production planning, supply chain optimization, scenario simulation, and predictive analytics.

Danfoss Context

Danfoss operates a global manufacturing network with multiple factories, tools, and varying production conditions. Planning production capacity and sourcing is complex due to differences in machine performance, tooling capabilities, maintenance requirements, and uncertainty in sales pipelines.

The Challenge

The challenge is to build an AI model that synthesizes these variables—sales pipelines, material master data, factory-specific processing times, and tool downtime - to simulate production scenarios, predict capacity bottlenecks, and optimize the timely sourcing of correct raw materials.

Your Mission

Design a solution that integrates sales pipeline probabilities, production parameters, and tooling

constraints to forecast capacity, simulate scenarios, and support sourcing decisions.

Potential Tools

- Salesforce (sales pipeline data)
- Anaplan (planning parameters)
- BI reporting tools
- Data modelling and machine learning tools

Potential Skills

- Data modelling and analysis
- Machine learning / predictive modelling
- Supply chain and production planning understanding
- Scenario simulation and optimization

Provided Data

- Sales Pipeline Data: Unapproved projects, expected timelines, and probability ratios (e.g., 50% probability) sourced from Salesforce
- Material Master Data: Base times needed to press specific components (e.g., one plate or gasket)
- Tooling & Maintenance Data: Tool capabilities and scheduled downtime/maintenance requirements
- Factory Parameters: Variations in pressing times across different factories
- Anaplan – AS-IS Planning Parameters (H/PCS)
- BI Report – Realised Production (PCS/H)
- Inventory & Transit Data

Deliverable(s)

1. Predictive Capacity Model: A prototype that forecasts work center load and identifies potential capacity bottlenecks across different factories
2. Scenario Simulation: Ability to run different production scenarios based on project probability ratios
3. Smart Sourcing Forecast: Automated predictions for timely sourcing of raw materials
4. Downtime Integration: Scheduling logic incorporating tool maintenance and varied processing speeds

Success Looks Like

A solution capable of forecasting production capacity, simulating scenarios, identifying bottlenecks, and enabling timely sourcing decisions while accounting for uncertainty, downtime,

and varying factory conditions.

Case Owner
Daniel Parapunov
Danfoss Climate Solutions

Case 4: Cross-Sales Opportunity Detection

Case description
We aim to identify customers with untapped cross-sales potential by analyzing historical sales data. Many products are typically sold together, but some customers only purchase one of them. By applying machine learning and statistical analysis, we can detect these gaps and recommend additional products to sales teams, increasing revenue and customer satisfaction.

Challenge Area

- Sales analytics
- Machine learning / statistical modelling
- Data engineering & visualization

Danfoss Context
Danfoss has a broad portfolio of products, and certain combinations are frequently purchased together. However, some customers only buy part of these combinations. Leveraging internal sales data and advanced analytics can help sales teams proactively identify and act on these opportunities.

The Challenge
Sales teams currently rely on manual insights or experience to spot cross-sales opportunities. This approach is time-consuming and inconsistent. The challenge is to automate detection of customers who are likely to buy complementary products, based on purchasing patterns across the customer base.

Your Mission
Develop a data-driven solution that uses machine learning or statistical association analysis to:

1. Identify product combinations frequently sold together.
2. Detect customers who have purchased only part of these combinations.
3. Provide actionable recommendations to sales teams via dashboards or reports.
4. Estimate the possible annual sales growth according to your analysis.

Potential Tools:
Databricks platform, including:

- Databricks Notebooks for collaborative development and analysis
- Delta Lake for data storage and versioning
- Databricks Dashboard for reporting

- Databricks Jobs for automation and scheduling

Potential Skills

- Data engineering (ETL pipelines)
- Machine learning / statistical modelling
- Data visualization and dashboard creation

Provided Data

- Historical sales transaction data (customer ID, product ID, date, quantity, price)
- Product metadata (category, segment, hierarchy)
- Customer metadata (region, segment, type)

Deliverable(s)

1. ML/statistical model to detect cross-sales opportunities.
2. Dashboard/report showing customers with high cross-sales potential.
3. Documentation of methodology and recommendations for integration into sales processes.

Success Looks Like

The solution outputs a ranked list of customers with specific product recommendations for cross-sales, based on historical purchasing patterns. Sales teams can use this to target customers with tailored offers, leading to measurable increases in revenue and improved customer relationships.

Case Owner

Rasmus Juul Jørgensen
Danfoss Climate Solutions – Sales Excellence & Analytics

Case 5: AI Use Case Intake - Powered by a Multi-Agent Orchestration System

Case description

At Danfoss, employees across all business units regularly identify opportunities where AI could improve their work. However, translating a raw idea into a well-structured, actionable AI use case - one that is technically feasible, business-justified, data-ready, and compliant - is a complex journey that today relies on a static intake form and significant back-and-forth with the AI team. This case challenges students to build a conversational, multi-agent AI system that guides employees dynamically through the full intake funnel: from initial idea articulation, through feasibility, data, ROI, and compliance assessment, all the way to a polished user story ready for the AI team's Product Owner. A central orchestrator agent serves as the entry point and dynamically invokes specialized sub-agents depending on the maturity and completeness of the use case being described. The goal is a smarter, faster, and more consistent AI intake process that scales across the organization.

Challenge Area

Conversational AI, Multi-Agent Systems, AI Governance, Process Automation, Natural Language Processing, Agentic Workflows

Danfoss Context

Danfoss has an active AI Incubator program that receives AI use case submissions from employees across all segments and business units. Today, a basic intake form exists, but it is static - it does not guide the submitter through quality assessment, feasibility thinking, data readiness, or business case formulation. As a result, many submissions arrive incomplete or poorly articulated, requiring the AI team's Product Owner to spend significant time in follow-up clarification loops before a case can be properly evaluated and prioritized. This creates bottlenecks in the intake pipeline, slows AI adoption, and reduces the quality and speed of innovation across the organization. A dynamic, intelligent intake assistant would dramatically improve both the submitter experience and the quality of cases reaching the AI team's backlog.

The Challenge

The core challenge is to replace a static, one-size-fits-all form with a dynamic, intelligent, conversational intake experience. The system must understand where a user is in their idea-development journey and route them to the right specialist agent at the right time. Early-stage ideas need help with clarity and problem framing. More developed ideas need feasibility checks, data availability assessments, ROI estimations, and compliance reviews. The orchestrator must handle users who are uncertain what they need, gracefully progress them through each maturity

stage, and consistently produce a structured, high-quality user story as the final deliverable — without requiring expert AI knowledge from the submitter.

Your Mission

Build a multi-agent orchestration system that:

- Provides a conversational entry point (Main Orchestrator Agent) where users describe their AI idea in plain language
- Idea Clarity Agent - helps the user articulate the problem, goal, and intended AI approach
- Feasibility Agent - assesses technical feasibility and AI-readiness of the use case
- Data Agent - evaluates data availability, quality, ownership, and accessibility
- ROI / Business Value Agent - helps estimate business impact, efficiency gains, and priority
- Compliance Agent - flags GDPR, data privacy, security, and AI governance considerations
- User Story Writer Agent - synthesizes all inputs into a structured, PO-ready AI use case user story
- Tracks maturity progression through the funnel (Idea → Scoping → Data Assessment → ROI → Compliance → User Story), only advancing the user when sufficient quality information has been gathered at each stage
- Produces a final, structured AI use case user story in a standard format that can be submitted directly to the AI team's Product Owner backlog

Potential Tools

- Microsoft Copilot Studio (agent orchestration and enterprise-ready conversational UI)
- Azure OpenAI Service / GPT-4o (LLM backbone for agent reasoning and generation)
- Python (custom agent logic, APIs, and orchestration scripting)
- Google Gemini Agents / Vertex AI Agent Builder (alternative agent framework)
- N8N (workflow automation and visual agent routing)
- Claude API / Claude Code (LLM reasoning and agentic coding assistance)
- LangGraph or CrewAI (multi-agent orchestration frameworks)
- Azure Logic Apps / Power Automate (integration and routing — optional)

Potential Skills

- Large Language Model (LLM) prompting and agent design
- Multi-agent system architecture and orchestration
- Conversational UX design
- Python programming

- API integration and system design
- Understanding of the AI/ML project lifecycle and use case evaluation criteria
- Basic knowledge of GDPR and enterprise data governance
- Familiarity with agile user story writing (As a... I want to... So that...)

Provided Data

The as-is use case intake process will be presented at the hackathon. Case examples will be provided. Team PO will be available for questioning.

Deliverable(s)

1. A working multi-agent prototype demonstrating the full intake funnel — from raw idea input to a structured user story output
2. A demo walkthrough showing at least 3 specialized sub-agents being invoked during a realistic intake conversation
3. A concise presentation (slides) covering: agent architecture design, orchestration logic and routing decisions, maturity funnel stages, an example of a generated user story, and reflections on scalability

Success Looks Like

- A user with only a rough, unstructured AI idea can engage the system through natural conversation and be guided through the full intake process
- The system correctly identifies gaps in the use case (missing data strategy, unclear ROI, compliance risks) and invokes the appropriate specialist agent at the right moment
- The final output is a well-structured, PO-ready AI use case user story that requires minimal human revision before entering the AI team's backlog
- The architecture is demonstrably extensible: new specialist agents can be added as the intake process evolves without redesigning the core system

Case Owner

Anton Benkestok
Danfoss Group IT – AI & Digital Accelerator

Case 6: Now You See Me - Unveiling Production Insights

Case description
This case challenges participants to connect fragmented datasets related to production orders and bills of materials to create a cohesive view of the production process of Danfoss Power Solutions production in Nordborg.

Challenge Area
AI, Pattern Recognition, Data Analysis

Danfoss Context
The Danfoss Power Solutions Production in Nordborg has a high vertical integration. This case is relevant as it aims to enhance the transparency of production planning and its management, leveraging existing data within SAP to improve decision-making and operational performance.

The Challenge
Participants are tasked with creating transparency in production by integrating various datasets, including production orders, materials, quantities, scrapped parts, and throughput times over a year, focusing on internal deliveries towards final assembly.

Your Mission
Develop a solution that visualizes the connections from semi-finished goods to final assembled products, enhancing understanding of production dynamics and allowing users to extract specific information based on material numbers or specific production orders.

Potential Tools
AI Models, data visualization tools (e.g., Power BI, Tableau), and programming languages (e.g., Python, R) for data manipulation

Potential Skills
Data analysis, visualization

Provided Data
Data about Productions orders, Bill of Material (recipes for Products), Plant Layout (xlsx, csv files)

Deliverables with increasing difficulty
1. Visualization of production data (e.g. Sankey-Diagram, Use of Plant Layout, etc)
2. Search functionality for single final products

3. Search functionality for specific timeframes and production areas.

Success Looks Like

Being able to give information about selected groups or areas.

Case Owner

Eike Hachmann
Danfoss Power Solutions – Orbital & Gear Solutions