O.R.I.O.N - Operational Response and Incident Observation Network

Principal Investigator: Diego Quijada (UD|CS)
diegoqui@udel.edu
(240) 935-4843

Subject Matter Expert: Chien-Chung Shen
Professor of Computer and Information Sciences, University of Delaware
cshen@udel.edu
(302) 831-1951

Project Manager: Gurneet Kaur (UCB|CS)

Engineering:
Ava Guevara (UCLA|ME) - AI Engineer
Kevin Santoush (UH|CS) - Software Engineer
Shahmeer Saud (RU|CS, FIN, MATH) - AI Engineer

Science:
Christian Cortez (MSJC|ME) - AI Research Scientist
Dustin Nguyen (CMU|CS) - AI Research Scientist

Programmatics:
Maverick Huynh (USC|ISE) - Logistics
Natalia Dominguez (UTD|FIN) - Financial Analyst

Applicable NASA Technology Taxonomies:

- TX04 - Robotics Systems
- TX05 - Communications, Navigation, and Orbital Debris Tracking and Characterization Systems
- TX06 - Human Health, Life Support, and Habitation Systems
- TX10 - Autonomous Systems
- TX11 - Software, Modeling, Simulation, and Information Processing
- TX13 - Ground, Test, and Surface Systems
- TX17 - Guidance, Navigation and Control

Figure 1: O.R.I.O.N. Operational Response and Incident Observation Network — Lunar Base Command Center. Source: AI-generated using Lovable by Team 8, 2026.

1. Abstract

Establishing a sustainable, long-term presence on the Moon through the development of a Moon Base on the South Pole of the Moon introduces new operational challenges that will require new systems to overcome those challenges. Moreover, in order for the mission to be successful, there are numerous systems that need to be operated and observed in order to diagnose problems when they happen and prevent future incidents. Current lunar surface concepts depend on separate systems for crew monitoring, rover tracking, emergency response, and system maintenance. The problem is that because some of these work independently from each other, incidents can arise that can potentially cause problems for other systems that end up taking longer to address and solve these issues. **O.R.I.O.N.** is an integrated lunar base network that connects astronaut safety systems, rover tracking, habitat sensors, life-support status, maintenance alerts, cybersecurity monitoring, and emergency response assets into one coordinated command system. It would monitor crew and rover locations, observe abnormal events in real time, detect maintenance or security issues, and recommend response actions within seconds. Unlike existing subsystem-specific monitoring approaches, **O.R.I.O.N** correlates cyber and physical events across multiple operational domains to produce a single prioritized incident and evidence-based response recommendation. The final deliverable, a 12-month project, will be a working software prototype with a simulated lunar base network, crew and rover tracking interface, incident alert dashboard, maintenance monitoring system, cybersecurity alert layer, and validation report.

2. Technology Merit and Work Plan

2.1 Need, Application, and Objective

Current lunar surface concepts inherit the ISS operational model: subsystem-specific monitoring with heavy reliance on ground controllers to notice, diagnose, and coordinate responses. On the ISS, this works because communication with mission control is continuous and near-instant, while dozens of flight controllers watch telemetry around the clock. A lunar base breaks each of those assumptions. Round-trip light delay is roughly 2.6 seconds; far-side or occluded operations may lose Earth contact entirely for extended periods, and local operations must also remain functional during a communications outage [7]. NASA studies have shown that communication delays increase operator workload and that onboard tools can improve coordination [1]. Artemis-era staffing plans also do not include an ISS-scale ground team per base. A sustained lunar surface outpost will depend on habitat, power, astronaut health, rovers, communications, and cybersecurity systems operating together. Existing technologies can monitor or manage individual vehicles and subsystems, but one underlying fault can still appear as several unrelated warnings. Our proposed Operational Response and Incident Observation Network (O.R.I.O.N.) addresses this gap by:

- **Collecting:** Status data from five operating domains: crew/spacesuits, habitat/life support, power, rovers, and communications/network security.
- **Detecting:** Hard safety-limit violations as well as slower degradation that may occur before a fixed limit is crossed.
- **Connecting:** Related alerts into one prioritized incident with a likely cause instead of several separate warnings.
- **Guiding:** Crew and mission operators to an approved response while keeping the human operator in control.

The 12-month, $10,000 effort will deliver and validate a working software prototype in a simulated lunar-base environment. Its purpose is to reduce the technical risk of future development by providing the integration architecture, measurable performance, and operator workflow before higher-cost hardware testing.

2.2 Technology Concept and Innovation

O.R.I.O.N. is an integrated lunar operations platform that unifies lunar base monitoring functions that today are handled by separate, independent systems: crew biometrics and location, rover telemetry, habitat environmental sensors, life-support status, power infrastructure, maintenance alerts, and network security events. O.R.I.O.N.'s main innovation is the integration between systems. The platform converts diverse data into a common format, connects related physical and cybersecurity events, and presents a single, understandable incident with supporting evidence elaborating on what caused the incident and what can be done to prevent and detect future incidents. The prototype is organized as five layers connected by a publish-subscribe telemetry bus, implemented in Python with containerized modules:

- **Simulation and fault injection:** Generate normal and faulty data for the five operating domains. Tests will include single faults, gradual degradation, cascading faults, missing data, cybersecurity events, and Earth-link loss to measure detection accuracy and timing.
- **Common data layer:** Translate every reading into the same record format: source, time, value, units, and data-quality flag. Modular Python containers will allow each function to be developed and tested independently.
- **Hybrid detection:** Use transparent engineering rules for hard safety limits and compare one statistical method with one lightweight machine-learning method for slower, less obvious changes.
- **Incident correlation and decision support:** Group alerts that are linked in time and by the base dependency map, then rank actions from a human-authored response playbook. Each recommendation will show its evidence and procedure.
- **Operational digital twin and cybersecurity:** Display the live status, relationships, locations, and incidents of base assets. Network events will enter the same pipeline as physical data, while the network and link outages will be modeled rather than physically built.

2.3 State of the Art and Differentiation

NASA already has advanced onboard automation, fault management, habitat autonomy, and telemetry analysis. **O.R.I.O.N.** is designed to complement these efforts rather than claim that they do not exist. Table 1 identifies the closest work, its measured evidence, and the remaining gap addressed by this proposal.

Relevant State of the Art	Measured Evidence / Present Capability	Remaining Gap	O.R.I.O.N. Relationship
Limit alarms and crew/ground procedures	Limit checking is widely used because it is fast and understandable [6]. One ISS rack operation required 3 control positions, 47 steps, 57 commands, and 276 telemetry checks before AMO automated the defined procedure [2].	Fixed limits can miss gradual or context-dependent faults, and separate alerts still require human synthesis.	Retains hard-limit rules but adds early-warning detection, cross-system grouping, and one incident view.
JPL telemetry anomaly detection	A published SMAP/MSL study evaluated 82 channels and 105 anomaly sequences, achieving 87.5% precision and 80.0% recall [6].	Models were created per channel; automated relationships and dependencies between channel anomalies were identified as future work.	Uses anomaly detection as one input, then adds dependency-based correlation and response ranking across operating domains.

Gateway Vehicle System Manager (VSM)	VSM covers 4 functions: mission/timeline, resource, fault, and vehicle control/operation, with automation ranging from advisory to autonomous [3].	It is the top-level manager for Gateway's vehicle architecture, not an open lunar-surface test framework spanning crew, rovers, and cyber events.	O.R.I.O.N. is a surface-operations integration and decision layer that could consume VSM-like subsystem outputs.
ISAAC and HOME habitat autonomy	ISAAC combined robot inspection with vehicle telemetry and was demonstrated in simulation, ground tests, and 16 ISS activities[4]. HOME develops resilient, autonomous habitats and multi-system simulators [5].	These efforts emphasize vehicle/habitat caretaking, robotics, and deep subsystem resilience.	O.R.I.O.N. focuses on base-wide cyber-physical incident correlation, crew-facing response guidance, and a reusable validation method.

Table 1: State of the Art Comparison and O.R.I.O.N. Differentiation

2.4 Technical Challenges and Risk Reduction

The principal risks, with mitigations, are listed below:

1. **Simulation fidelity.** Metrics are meaningful if the test cases are credible. Each scenario will be traced to a public NASA fault class (ISS/Artemis) or operating concept; its assumptions will be recorded, SME-reviewed, and the catalog will be published with the results. All findings will be labeled as simulation-only.
2. **False alerts and missed faults.** O.R.I.O.N. will retain deterministic limits as a safety floor, measure precision and recall on held-out cases, tune thresholds only on training cases, and compare the integrated result with an independent-alarm baseline.
3. **Limited lunar training data.** Due to the lack of real lunar-base telemetry, the learned model will train on simulated nominal operation, and a simpler statistical detector will remain the required fallback. Results for both methods will be reported.
4. **Trust and safety.** Free-form AI responses and automatic safety-critical commands are excluded. Human-authored playbooks will show the triggering evidence, confidence, and affected assets, and an operator will approve any action.
5. **Integration and scope.** The common data format will be frozen after Month 2, and one complete cascading-fault demonstration will be required by Month 6. Delay-tolerant networking, automatic device isolation, flight-grade encryption, and a high-fidelity physics twin are follow-on work.

2.5 Key Performance Parameters

Table 2 carries out our proposed Key Performance Parameters (KPPs). Minimum performance defines supplanting the SOA in the demonstration context; targets define success. All are measured against the published fault-scenario set in the validation report. Supporting engineering results such as alert correctness, false-alert rate, predictive warning, and operation during a simulated Earth-link outage will be reported in the validation package.

KPP	Reference Baseline	Minimum Performance	Target and Verification
Incident Detection Time	Manual reporting or separate subsystem alerts; no base-wide correlated incident time.	Display an automated incident alert within 5 min in at least 80% of scripted fault trials.	Display the alert within 1-5 simulated min in at least 90% of held-out trials; measure from fault injection to dashboard alert.
Emergency Response Speed	Crew-dependent manual coordination after an	Display a relevant response from the	Display a playbook-matched recommendation within 60 s in at

	alert is received.	approved playbook within 60 s in at least 80% of confirmed incidents.	least 90% of held-out incidents; measure from incident confirmation to dashboard recommendation.
Alarm Consolidation	Separate subsystem alarms require the operator to determine whether they share one cause.	Correctly group related alerts into one incident in at least 70% of scripted multi-subsystem fault trials.	Correctly consolidate at least 80% of held-out multi-subsystem trials; score against scenario ground truth, with unrelated faults kept separate.

Table 2: O.R.I.O.N. Key Performance Parameters and Verification Criteria

2.6 Development Work Plan and Final Products

The technical work is organized into four phases. Each phase ends with an observable product or test, and each directly addresses a risk identified above (Table 3).

Phase	Technical Work	Exit Criterion / Product	Risk Addressed
1. Foundation *Months 1-2*	Freeze requirements, five data domains, common record format, dependency map, scenario catalog, and independent-alarm baseline. Build nominal simulator and dashboard frame.	Architecture and requirements review; approved interface specification; nominal data visible end to end.	Scope, interface mismatch, simulation credibility.
2. Core Integration *Months 3-6*	Implement fault injection, hard-limit detection, incident grouping, initial playbooks, data logging, and live digital-twin display.	Month 6 minimum viable prototype: one cascading fault passes through every layer to a displayed recommendation.	Late integration, unclear interfaces, schedule growth.
3. Intelligence and Security *Months 7-9*	Compare statistical and lightweight ML detection; add gradual-fault prediction, cyber events, communications delay, and outage cases; tune only on training scenarios.	Interim KPP report on held-out cases; selected detector and documented fallback.	Data scarcity, false alerts, model complexity, cyber overclaim.
4. Validation and Delivery *Months 10-12*	Run the full nominal and fault test campaign, complete sensitivity and failure analysis, fix defects, package code and documentation, and conduct final demonstration.	Validated prototype, source package, dashboard, scenario library, interface specification, and final report.	Unverified claims, poor repeatability, incomplete handoff.

Table 3: O.R.I.O.N.'s Technical Work Plan, Exit Criteria, and Risk Reduction

At the end of the base period, NASA will receive a repeatable proof of concept: runnable source code, a simulated lunar-base test environment, documented adapters and data format, a cyber-physical dependency map, response playbooks, an operator dashboard, the full scenario catalog, and a validation report tied to the KPPs. The common data schema, dependency model, correlation logic, and auditable playbook-ranking approach will be documented in

“NASA Form 1679” for NASA's determination of reportability. The near-term infusion path is a ground or analog-habitat demonstration; the longer-term path is integration as an advisory layer that accepts outputs from qualified subsystem managers rather than replacing them.

3. Project Management Approach

3.1 Management Strategy and Controls

O.R.I.O.N. will be executed over 12 months (Months 1–12) using a hybrid Systems Engineering and Agile approach, combining NASA's Systems Engineering lifecycle with iterative development sprints across four phases: Foundation, Core Integration, Intelligence and Security, and Validation and Delivery.

Roles and Responsibilities: Each phase has a designated technical owner. The Principal Investigator owns system architecture and cross-domain integration decisions, serving as the final technical authority for the project. The Project Manager owns Validation, Documentation, Deployment, and overall project oversight/budget control, responsible for keeping deliverables on schedule and within budget. The Engineering team owns Core Integration and Intelligence & Security development, responsible for building the data pipeline, detection modules, dashboard, and system integration. The Science team owns validation methodology, evidence review, and state-of-the-art benchmarking, ensuring test design and results are technically defensible. The Programmatics team supports deployment planning, logistics, and financial tracking across all phases. Full role descriptions, hourly commitments, and team backgrounds are detailed in *Section 4.1*.

Progress is evaluated through biweekly technical meetings, monthly management/mentoring reviews, and four formal gate reviews at Months 2, 6, 9, and 12 (*Section 3.2*). Technical performance is measured against the KPPs defined in *Section 2.5*; schedule performance is tracked via milestone completion in shared project-management software.

Risk Management: A risk register is reviewed monthly, with mitigations built directly into the schedule: network-reliability risk is addressed through dedicated interface work in Foundation (Months 1–2); AI-integration and false-alert risk is buffered by two protected margin weeks each in Core Integration (Months 3–6) and Intelligence and Security (Months 7–9) — the two highest-effort phases; and data-scarcity/model-trust risk is mitigated by retaining a deterministic rules-based fallback throughout, per *Section 2.4*.

3.2 Schedule, Milestones, and Deliverables

The schedule moves from definition to integration, advanced testing, and final validation. Each phase ends with an observable gate that must be passed before the next phase becomes the working baseline. The 12-month plan provides 48 active work weeks plus four protected margin weeks. Figure 2 summarizes the plan. Appendix B expands it into weekly tasks, owners, dependencies, and outputs (*See Appendix B for detailed Gantt chart*).

- **Foundation (Months 1–2):** Define requirements, interfaces, subsystem dependencies, test scenarios, and the independent-alarm baseline. The Month 2 gate approves the architecture and confirms nominal data flow from source to dashboard.
- **Core Integration (Months 3–6):** Build the data pipeline, fault injection, rules-based detection, alarm consolidation, response playbooks, logs, and dashboard. The Month 6 gate demonstrates one cascading fault through a displayed recommendation. We allocate two margin weeks to protect integration.
- **Intelligence and Security (Months 7–9):** Compare statistical and lightweight machine-learning detection, add predictive degradation, cyber events, communications outages, and held-out KPP tests. The Month 9 gate selects the detector and documents a reliable rules-based fallback; two margin weeks protect testing.

- **Validation and Delivery (Months 10–12):** Run nominal and fault campaigns, verify KPPs, correct defects, and package the prototype, documentation, and final demonstration. The Month 12 gate accepts the validated system and delivery package.

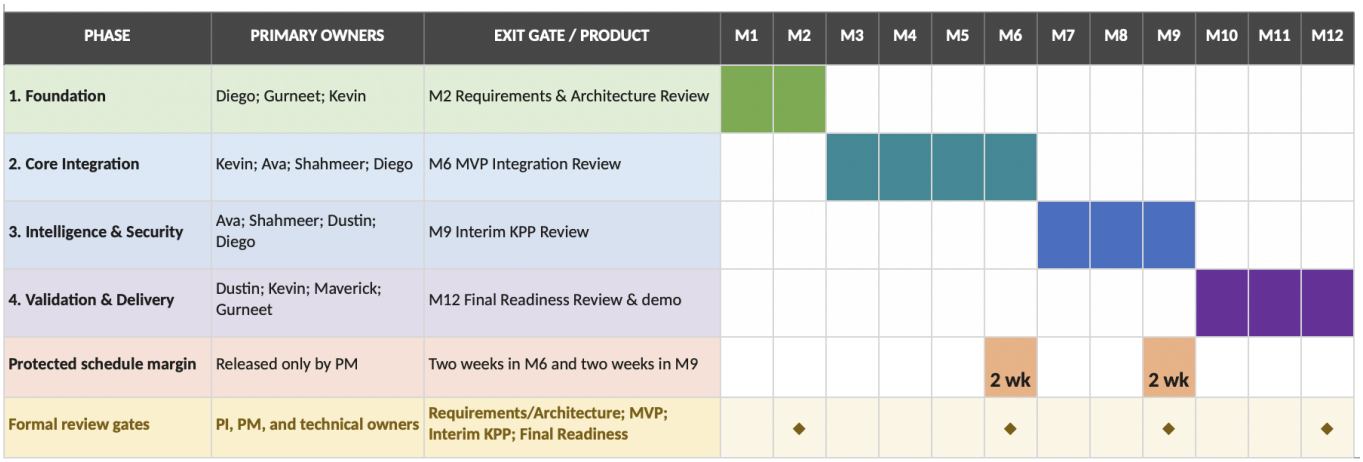


Figure 2: 12-Month Gantt Chart, Simplified

3.3 Resources, Labor, and Basis of Estimate

Project Element	Required Skills	Est. Hours	Est. Labor Cost*
Requirements & Project Planning	Project Management, Systems Engineering	40	$800
System Design	Systems Engineering, Network Architecture	60	$1,200
Prototype Development	Software Engineering, AI Development	120	$2,400
Integration & Testing	Software Engineering, Test & Verification	100	$2,000
Documentation & Deployment	Technical Writing, Project Management	40	$800
Project Oversight	Project Management	40	$800
Total Labor		**400**	**$8,000**

Table 4: Project Elements, Hourly Estimates, and Skills/Labor Costs
Estimated at a stipend-level blended rate of $20/hr; update with actual team/institutional cost basis.

Non-labor costs: Cloud computing infrastructure, software licensing, and a minimal simulation-platform footprint are estimated at **$2,000** for the 12-month period, using free-tier/academic-tier cloud credits and open-source tooling wherever possible to hold costs down.

Total estimated project cost: $10,000 ($8,000 labor + $2,000 non-labor), reflecting a scoped-down prototype effort appropriate for the available funding.

4. Teaming and Workforce Development

4.1 Core Team, Roles, Experience, and Commitment

O.R.I.O.N.'s proposed core-team staffing totals approximately 1,584 hours across 48 active weeks. Our team and respective roles are described below:

- **Diego Quijada – Principal Investigator (UD|CS); 5 h/wk.** Diego will lead cyber-physical integration. Responsibilities include system architecture, cybersecurity, integration, and final technical decisions.
- **Gurneet Kaur – Project Manager (UCB|CS); 4 h/wk.** Responsible for O.R.I.O.N. planning, scheduling, and risk management. Responsibilities include adherence to the proposed schedule, milestones, work breakdown structure, and coordination.
- **Kevin Santhosh – Software Engineer (UH|CS); 5 h/wk.** Responsible for the software architecture, distributed systems, communications, and integration work. Responsibilities include developing the data layer, interfaces, dashboard, communications, and end-to-end integration.
- **Ava Guevara – AI Engineer (UCLA|ME); 4 h/wk.** Responsible for O.R.I.O.N. machine learning, sensor-fusion, and predictive-analysis research. Responsibilities include sensor behavior, anomaly methods, incident detection, and predictive testing.
- **Shahmeer Saud – AI Engineer (RU|CS, FIN, MATH); 4 h/wk.** Responsible for the O.R.I.O.N. decision-system, threat-ranking, and KPP work. Responsibilities include playbooks, incident prioritization, performance measures, and KPP evaluation.
- **Dustin Nguyen – AI Research Scientist (CMU|CS); 3 h/wk.** Responsible for O.R.I.O.N. NASA-roadmap, literature, technical-reference, and validation-method research. Responsibilities include test design, evidence review, technical references and validation report.
- **Christian Cortez – AI Research Scientist (MSJC|ME); 3 h/wk.** Responsible for state-of-the-art and technology comparison research. Responsibilities include NASA comparisons and remaining technology gaps.
- **Maverick Huynh – Logistics (USC|ISE); 3 h/wk.** Responsible for O.R.I.O.N. deployment, logistics, and maintainability planning. Responsibilities include maintenance, logistics, and deployment phases.
- **Natalia Dominguez – Financial Analyst (UTD|FIN); 2 h/wk.** Responsible for budget estimation, resource planning, and labor justification. Responsibilities include budget and labor estimate and cost tracking.
- **Chien-Chung Shen – Subject-Matter-Expert (UD|CS Prof.); ~18 h over 12 months.** Provides faculty guidance for the software prototype and reviews architecture, AI/cyber/network integration, test design, KPP evidence, and technical feasibility.

4.2 Workforce Development and Resources

Biweekly technical meetings, monthly mentoring reviews, and Months 2, 6, 9, and 12 gates provide recurring feedback. During Months 1-2, Diego pairs with Ava and Kevin on Python, Git, and tests. Dustin pairs with Christian and Shahmeer on reproducible notebooks. Gurneet pairs with Maverick and Natalia on issue/deployment and repository/cost workflows. Each completes one reviewed artifact within planned hours. To note, Natalia has no production-code duty. Kevin covers prototype front-/back-end work, Diego covers cyber/network architecture, and the SME reviews implementation and test methods. Lunar operations and system safety remain separate gaps mitigated through simulation-only scope, NASA source traceability, and an additional reviewer if available. Our work uses standard computers, Python, a shared repository, and virtual tools.

- **Appendix**

A. References

[1] Frank, J., Spirkovska, L., McCann, R., Wang, L., Pohlkamp, K., and Morin, L. Autonomous Mission Operations. 2012 IEEE Aerospace Conference. NASA NTRS 20170011337. https://ntrs.nasa.gov/citations/20170011337

[2] Stetson, H. K., et al. AMO EXPRESS: A Command and Control Experiment for Crew Autonomy Onboard the International Space Station. 2015. NASA NTRS 20150016490. https://ntrs.nasa.gov/citations/20150016490

[3] Dabney, J., Badger, J., and Rajagopal, P. Trustworthy Autonomy for Gateway Vehicle System Manager. 2023. NASA NTRS 20230008770. https://ntrs.nasa.gov/citations/20230008770

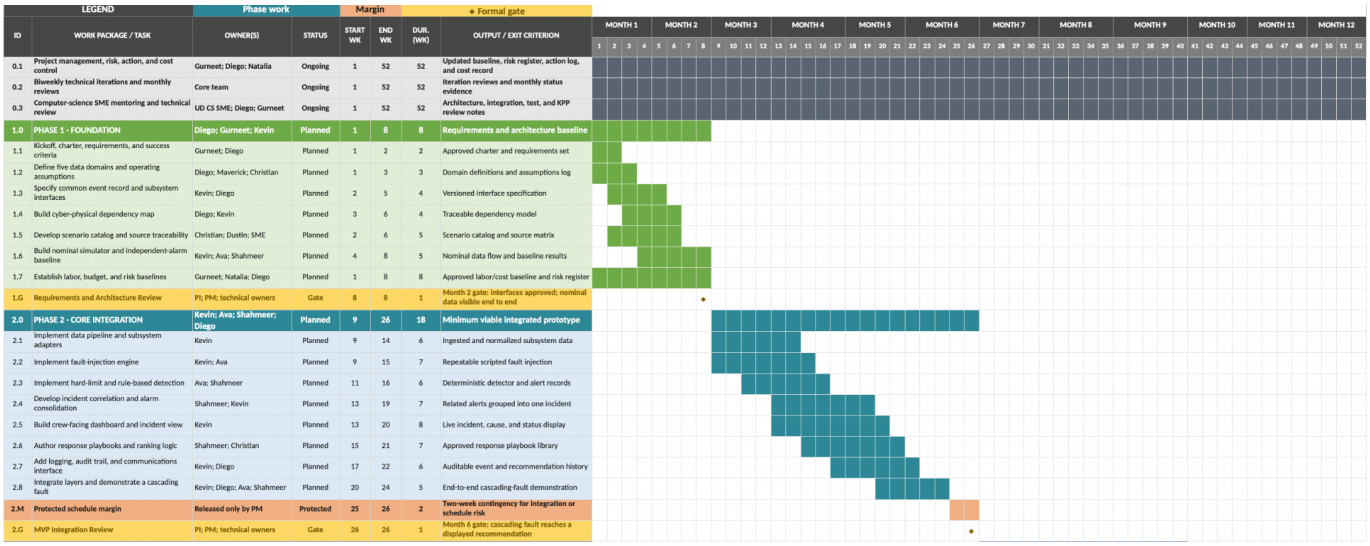
[4] National Aeronautics and Space Administration. Integrated System for Autonomous and Adaptive Caretaking (ISAAC). Updated June 30, 2025. https://www.nasa.gov/integrated-system-for-autonomous-and-adaptive-caretaking-isaac/

[5] National Aeronautics and Space Administration. Habitats Optimized for Missions of Exploration (HOME). Updated June 22, 2026. https://www.nasa.gov/directorates/stmd/space-tech-research-grants/habitats-optimized-for-missions-of-exploration-home/

[6] Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. KDD 2018. https://doi.org/10.1145/3219819.3219845

[7] Lester, D., and Thronson, H. Low-Latency Lunar Surface Telerobotics from Earth-Moon Libration Points. 2011. NASA NTRS 20110022582. https://ntrs.nasa.gov/citations/20110022582

B. Detailed Gantt Chart



LEGEND: Phase work | Margin | ◆ Formal gate

ID	WORK PACKAGE / TASK	OWNER(S)	STATUS	START WK	END WK	DUR. (WK)	OUTPUT / EXIT CRITERION
0.1	Project management, risk, action, and cost control	Gurneet; Diego; Natalia	Ongoing	1	52	52	Updated baseline, risk register, action log, and cost record
0.2	Biweekly technical iterations and monthly reviews	Core team	Ongoing	1	52	52	Iteration reviews and monthly status evidence
0.3	Computer-science SME mentoring and technical review	UD CS SME; Diego; Gurneet	Ongoing	1	52	52	Architecture, integration, test, and KPP review notes
1.0	PHASE 1 - FOUNDATION	Diego; Gurneet; Kevin	Planned	1	8	8	Requirements and architecture baseline
1.1	Kickoff, charter, requirements, and success criteria	Gurneet; Diego	Planned	1	2	2	Approved charter and requirements set
1.2	Define five data domains and operating assumptions	Diego; Maverick; Christian	Planned	1	3	3	Domain definitions and assumptions log
1.3	Specify common event record and subsystem interfaces	Kevin; Diego	Planned	2	5	4	Versioned interface specification
1.4	Build cyber-physical dependency map	Diego; Kevin	Planned	3	6	4	Traceable dependency model
1.5	Develop scenario catalog and source traceability	Christian; Dustin; SME	Planned	2	6	5	Scenario catalog and source matrix
1.6	Build nominal simulator and independent-alarm baseline	Kevin; Ava; Shahmeer	Planned	4	8	5	Nominal data flow and baseline results
1.7	Establish labor, budget, and risk baselines	Gurneet; Natalia; Diego	Planned	1	8	8	Approved labor/cost baseline and risk register
1.G	Requirements and Architecture Review	PI; PM; technical owners	Gate	8	8	1	Month 2 gate: interfaces approved; nominal data visible end to end
2.0	PHASE 2 - CORE INTEGRATION	Kevin; Ava; Shahmeer; Diego	Planned	9	26	18	Minimum viable integrated prototype
2.1	Implement data pipeline and subsystem adapters	Kevin	Planned	9	14	6	Ingested and normalized subsystem data
2.2	Implement fault-injection engine	Kevin; Ava	Planned	9	15	7	Repeatable scripted fault injection
2.3	Implement hard-limit and rule-based detection	Ava; Shahmeer	Planned	11	16	6	Deterministic detector and alert records
2.4	Develop incident correlation and alarm consolidation	Shahmeer; Kevin	Planned	13	19	7	Related alerts grouped into one incident
2.5	Build crew-facing dashboard and incident view	Kevin	Planned	13	20	8	Live incident, cause, and status display
2.6	Author response playbooks and ranking logic	Shahmeer; Christian	Planned	15	21	7	Approved response playbook library
2.7	Add logging, audit trail, and communications interface	Kevin; Diego	Planned	17	22	6	Auditable event and recommendation history
2.8	Integrate layers and demonstrate a cascading fault	Kevin; Diego; Ava; Shahmeer	Planned	20	24	5	End-to-end cascading-fault demonstration
2.M	Protected schedule margin	Released only by PM	Protected	25	26	2	Two-week contingency for integration or schedule risk
2.G	MVP Integration Review	PI; PM; technical owners	Gate	26	26	1	Month 6 gate: cascading fault reaches a displayed recommendation

3.0	PHASE 3 - INTELLIGENCE AND SECURITY	Ava; Shahmeer; Dustin; Diego	Planned	27	40	14	Selected detector, fallback, and held-out KPP evidence
3.1	Develop statistical anomaly detector	Ava; Dustin	Planned	27	31	5	Statistical detector benchmark
3.2	Develop lightweight ML detector and comparison	Ava; Shahmeer	Planned	29	34	6	ML-versus-statistical comparison
3.3	Add predictive degradation scenario	Ava; Christian	Planned	30	35	6	Predictive-warning demonstration
3.4	Add cyber-event monitoring scenarios	Diego; Kevin	Planned	27	34	8	Cyber-physical incident scenarios
3.5	Add communications delay and link-outage cases	Diego; Kevin	Planned	32	36	5	Delay and Earth-link outage tests
3.6	Build held-out test set and KPP harness	Dustin; Christian; Shahmeer	Planned	30	38	9	Repeatable KPP test harness and held-out cases
3.7	Select detector and document deterministic fallback	Ava; Diego; SME	Planned	35	38	4	Selection rationale and fallback configuration
3.M	Protected schedule margin	Released only by PM	Protected	39	40	2	Two-week contingency for data, model, cyber, or test risk
3.G	Interim KPP Review	PI; PM; technical owners	Gate	40	40	1	Month 9 gate: detector and deterministic fallback approved
4.0	PHASE 4 - VALIDATION AND DELIVERY	Dustin; Kevin; Maverick; Gurneet	Planned	41	52	12	Validated prototype and delivery package
4.1	Run nominal and scripted fault campaign	Dustin; Ava; Shahmeer	Planned	41	45	5	Complete nominal and fault results set
4.2	Verify detection, response, and consolidation KPPs	Dustin; Shahmeer	Planned	42	46	5	KPP results with scripted and held-out evidence
4.3	Perform false-alert, sensitivity, and failure analysis	Ava; Dustin	Planned	43	47	5	Limits, false-alert rate, and failure analysis
4.4	Correct defects and rerun affected tests	Kevin; Ava; Diego	Planned	45	49	5	Closed defects and regression results
4.5	Package code, dashboard, interface specification, and user documentation	Kevin; Dustin; Maverick	Planned	44	50	7	Versioned source and documentation package
4.6	Complete deployment, infusion, budget closeout, and NTR draft	Maverick; Gurneet; Natalia	Planned	46	51	6	Infusion plan, cost closeout, and NASA Form 1679 draft
4.7	Prepare final report and demonstration	All core members	Planned	49	52	4	Final report, presentation, and demonstration package
4.G	Final Readiness Review and Demonstration	PI; PM; full team	Gate	52	52	1	Month 12 gate: validated prototype and all final products accepted

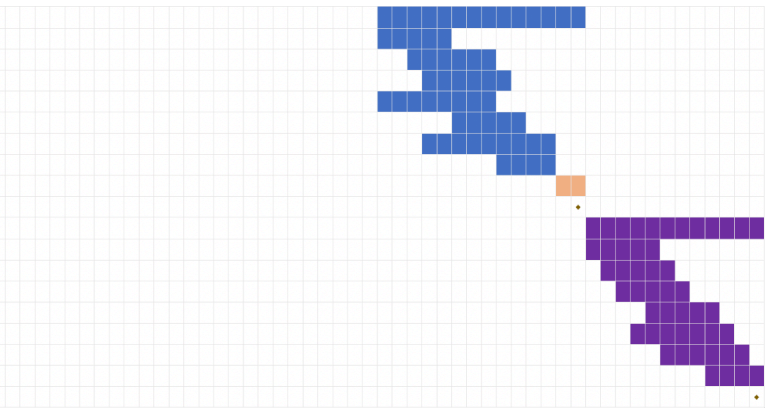


Figure B-1: Detailed Gantt Chart: Tasks, Owners, Dependencies, and Deliverables

C. NTR – NASA Form 1679

National Aeronautics and Space Administration

Disclosure of Invention and New Technology (Including Software)

Form Approved

O.M.B. NO.

2700-0009

7/22/26

INT. DOCKET NO./ CONTRACTOR TRACKING NO.

NASA CENTER NTR BELONGS TO:

NASA CASE NO. (OFFICIAL USE ONLY)

e-NTR Number

(OFFICIAL USE ONLY)

This is an important legal document. Carefully complete and forward to the Patent Representative (NASA in-house innovation) or New Technology Representative (contractor/grantee innovation) at NASA. Use of this report form by contractor/grantee is optional; however, an alternative format must

at a minimum contain the information required herein. NASA in-house disclosures should be read, understood and signed by a technically competent witness in the witness signature block at the end of this form. In completing each section, use whatever detail deemed appropriate for a "full and complete disclosure." Contractors/Grantees please refer to the New Technology or Patent Rights – Retention by the Contractor clauses. When necessary, attach additional documentation to provide a full, detailed description.

1. NEW TECHNOLOGY TITLE

O.R.I.O.N - Operational Response & Incident Observation Network

2. INNOVATOR(S) *(For each innovator provide: Name, Title, Work Phone Number, Org Code, and Work E-mail Address. If multiple innovators, number each to match Box 5.)*

Diego Quijada — Principal Investigator, Kevin Santhosh — Software Engineer, Ava Guevara — AI Engineer, Shahmeer Saud — AI Engineer, Dustin Nguyen — AI Research Scientist, Christian Cortez — AI Research Scientist, Gurneet Kaur — Project Manager, Maverick Huynh — Logistics, Natalia Dominguez — Financial Analyst

3.BRIEF ABSTRACT *(Describe your technology.)*

- The Operational Response & Incident Observation Network (O.R.I.O.N.) is a software framework that helps monitor and manage incidents during long-term lunar surface missions. O.R.I.O.N. combines data from five key areas: crew and spacesuits, habitat and life support, power, rovers, and communications/cybersecurity. Instead of showing many separate alarms caused by the same problem, O.R.I.O.N. connects related warnings into a single incident. It can detect immediate safety issues and slower system degradation, then recommend responses while keep human operators in control of critical decisions.

4. DESCRIPTION OF THE PROBLEM OR OBJECTIVE THAT MOTIVATED THE INNOVATION'S DEVELOPMENT (*General description of problem/objective or unique problem characteristics.)*

- O.R.I.O.N. was created to help manage a complex lunar base when communication with Earth is delayed or unavailable. A single system failure could trigger multiple alarms across different systems, making it difficult for operators to quickly understand what is happening. O.R.I.O.N. reduces this workload by: Collecting data → Detecting problems → Connecting related alerts → Prioritizing incidents → Showing evidence → Recommending approved responses

O.R.I.O.N. supports human decision-making rather than replacing it. Safety-critical actions remain under human control, and the system does not automatically execute critical commands.

5. TECHNICALLY COMPLETE DESCRIPTION OF INNOVATION *(Purpose and description of innovation/software and explanation of mode of operation referring to drawings, sketches, photographs, graphs, flow charts, and/or parts or ingredient lists illustrating the components; functional operation; alternate embodiments of the innovation/software and supportive theory.)*

- The **Operational Response & Incident Observation Network (O.R.I.O.N.)** is a software system designed to help astronauts and mission operators monitor a future lunar base. It collects information from crew and spacesuits, habitats and life-support systems, power systems, rovers, and communications. O.R.I.O.N. puts this information into one common system, checks for problems or unusual changes, and groups related warnings into one clear incident instead of showing many separate alerts. It then shows the problem on a dashboard and provides a recommended response based on a human-written procedure. The operator still makes the final decision. O.R.I.O.N.'s main innovation is bringing different monitoring systems, fault detection, alerts, and response guidance together.

6.UNIQUE OR NOVEL FEATURES *(Provide brief details focused on what component(s) or method step(s) differentiate(s) the new technology from other similar technologies (aka the "secret sauce"). Include as attachments any presentations, images, flowcharts, etc. that help identify the unique component(s) or method step(s). If there are no unique component(s) and method step(s) (e.g. NTR submitted only for software release), state "None.")*

- *It connects the dots across systems. Other tools watch one system each and each sends its own alarm. O.R.I.O.N. takes alarms from crew, habitat, power, rovers, and cybersecurity and combines related ones into a single incident with a likely cause. Published NASA anomaly-detection work checks one data channel at a time and lists this kind of cross-system connection as something still to be built — O.R.I.O.N. build it.*

- *It keeps humans in control. Every recommendation comes from a human-written playbook and shows its evidence and reasoning. O.R.I.O.N. never fires off automatic safety-critical commands or AI answers.*
- *It has a reliable backup. O.R.I.O.N. uses a smarter AI detector but always keeps simple rule-based limit checks as a safety net, since real lunar data doesn't exist yet to fully train the AI.*

7 COMMERCIALIZATION POTENTIAL (*Identify other applications for this technology beyond the specific NASA use. What type of industries would be most applicable for this technology? Are there related commercial products that you're aware of that would benefit from this technology? List any companies that you've contacted, or think may be interested in using this technology.)*

- Industries that could use it:
  Critical infrastructure — power plants, water systems, and utilities.
  Remote sites — offshore platforms, mining, ships, and research stations that can't rely on constant outside help.
  Large facilities — data centers, hospitals, and smart-building control rooms.

8. DEGREE OF TECHNOLOGY SIGNIFICANCE (Which best expresses the degree of technological significance of this innovation?)

Modification to Existing Technology Substantial Advancement in the Art Major Breakthrough

9. QUESTIONS FOR SOFTWARE ONLY

a. Does this technology include custom software, developed wholly or in part under NASA funding? YES NO

b. Is this technology primarily a software product or computer program technology (versus primarily a hardware technology)? YES NO

c. Could the software be used or adapted for other applications outside of your project? YES NO

d. Does the software contain any embedded, third-party code? YES NO

If yes, list each third-party code by title and version, under what license they were obtained, and either cut and paste the license below or provide the URL for the license to the downloaded version of the third-party code:

e. Does the software call any third-party code when it runs? YES NO

If yes, list each third-party code by title and version, under what license they were obtained, and either cut and paste the license below or provide the URL for the license to the downloaded version of the third-party code:

f. Can the software be distributed without third-party code? YES NO

g. Copyright registered? YES NO UNKNOWN

If yes, then by whom?

h. Are there any programmatic restrictions or other sensitivities that impact release/distribution of the software (e.g., contains Government sensitive information/command and control/spaceflight software, etc.)? YES NO UNKNOWN

If yes, explain

i. State of Development (for software only)

Concept Only Requirement Phase Design Phase Code Completed Code Testing Complete Used in Current Work

10. STATE OF DEVELOPMENT (*For software only, complete State of Development in question 9i.)*

Concept Only Design Prototype Modification Production Model Used in Current Work

D. Quad Chart

O.R.I.O.N (Operation Response and Incident Observation Network)

PI: Diego Quijada, Team 8

Goal / Objective

Phase 3 lunar bases will require continuous monitoring of astronaut safety, habitat systems, rover activity, power infrastructure, life support, communications, and maintenance operations. Current lunar surface concepts depend on separate systems for crew monitoring, rover tracking, emergency response, and system maintenance, which can delay incident detection and corrective action. **O.R.I.O.N.** is an integrated lunar base network that connects astronaut safety systems, rover tracking, habitat sensors, life-support status, maintenance alerts, cybersecurity monitoring, and emergency response assets into one coordinated command system.

The final deliverable from the $10,000, 12-month project will be a working software prototype with a simulated lunar base network, crew and rover tracking interface, incident alert dashboard, maintenance monitoring system, cybersecurity alert layer, and validation report.

This concept qualifies for an NTR.

Figure 1. O.R.I.O.N. Operational Response and Incident Observation Network — Lunar Base Command Center. Source: AI-generated using Lovable by Team 8, 2026.

Team Overview

Team Members:

- *Shahmeer Saud* — RU — CS, FIN, MATH
- *Kevin Santhosh* — UH — CS
- *Maverick Huynh* — USC — ISE
- *Ava Guevara* — UCLA — ME
- *Diego Quijada* — UD — CS
- *Dustin Nguyen* — CMU — CS
- *Gurneet Kaur* — UCB — CS
- *Natalia Domínguez* - UTD — FIN
- *Christian Cortez* - MSJC — ME

Team Capabilities:

Team 8 has relevant experience in computer science, artificial intelligence, machine learning, software development, autonomous systems, cybersecurity, systems engineering, mechanical design, data analysis, simulation, and mission planning. The team will use Python-based tools, software simulation environments, network modeling, sensor data processing, and AI-assisted alert logic to develop a prototype lunar base operations network.

SMEs:

- **Confirmed:** Chien-Chung Shen, University of Delaware Department of Computer and Information Sciences
- **Potential:** Weisong Shi, University of Delaware Department of Computer and Information Sciences

NASA Technology Taxonomies: TX04, TX05, TX06, TX10, TX11, TX13, TX17

Metrics and Key Performance Parameters

O.R.I.O.N. improves lunar base safety and reliability by connecting crew monitoring, rover tracking, habitat systems, maintenance alerts, cybersecurity monitoring, and emergency response into one integrated network. In the future state, O.R.I.O.N. would reduce cost, schedule, and risk by improving incident detection, response coordination, base awareness, and preventive maintenance during sustained lunar operations.

KPP	SOA	Target
Incident Detection Time	Manual reporting or separate system alerts	Automated alerts within 1–5 min
Emergency Response Speed	Crew-dependent manual coordination	Response recommendation within ≤60 sec

Figure D-1: O.R.I.O.N. Project Quad Chart