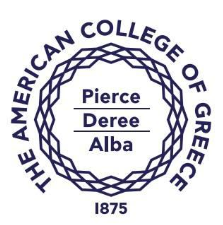
**ITC6045 – Project Management in Data Science**

Final Project Report



**Instructor:** K. Bloutsos

**Submitted by:**

Kapsalis C.

Stavrogiannis C.

Tzoras K.  
 Proiskos G.

**Table of Contents**

[Chapter 1: Introduction 3](#_Toc1478473532)

[Chapter 2: Project Management Approach 4](#_Toc822751064)

[2.1 Traditional - 'Waterfall' Approach 5](#_Toc1912519050)

[2.2 Agile Approach 5](#_Toc1099758210)

[2.3 The Case for a Hybrid Approach 5](#_Toc725230475)

[2.4 The Helios Hybrid Approach 5](#_Toc2122829410)

[Chapter 3. Project Phases and Timeline 6](#_Toc416636568)

[Chapter 4: Team Structure and Rationale 12](#_Toc887530662)

[4.1 Team Structure 13](#_Toc1705003)

[4.2 Alignment of FTEs with Project Requirements 13](#_Toc1827583759)

[Chapter 5: Cost Analysis and FTE Usage 14](#_Toc861177920)

[Chapter 6: Phase Gates and Risk Assessment 16](#_Toc1097003954)

[6.1 Technical and Operational Risks 17](#_Toc420037705)

[6.2 Phase Gates 17](#_Toc59512227)

[6.3 Governance and Compliance 18](#_Toc607651897)

[Chapter 7: Progress Monitoring & Success Metrics 18](#_Toc1687797730)

[7.2 Direct Metrics 19](#_Toc890365978)

[7.3 Indirect Metrics 20](#_Toc1707328281)

[7.4 Tracking Agile Cadence 20](#_Toc1302634531)

[Chapter 8: Conclusion 20](#_Toc161746790)

[References 22](#_Toc965420035)

**Table of Tables**

[Table 1: Summary showcasing No material difference in timeline or execution method for Phase 1. 5](#_Toc203911369)

[Table 2: Aggressive scenario prioritizes speed via overlapping activities and upfront cloud provisioning. 6](#_Toc203911370)

[Table 3: Aggressive scenario uses offshore capacity to accelerate AI development and data engineering concurrently. 7](#_Toc203911371)

[Table 4: Maintenance and Calibration Summary 9](#_Toc203911372)

[Table 5: Summary Table of FTEs 12](#_Toc203911373)

[Table 6: Cost Breakdown Table 13](#_Toc203911374)

# **Chapter 1: Introduction**

The Helios 'Predictive Maintenance' project is a prime example of the 'AI projects' becoming increasingly prevalent in corporate backlogs. Its apparent business value is significant, aiming at the production of a supervised learning model that predicts failures of the revenue-generating assets of the company.

In recent years, this type of projects has emerged as a new challenge for project management practitioners. As stressed by Vial, Cameron, Giannelia, and Jiang (2023), the reason is that they differ from common software engineering projects in two important aspects. First, the nature and technical requirements of their implementation are unique to this type of project. Second, they bring together people with unique and niche expertise. When combined with the timeline, deliverables, and cost predictability requirements often stressed in corporate environments, it becomes clear that such a project necessitates special handling.

Our discussion highlights the need for a hybrid approach, building on top of the advantages of the most popular PM approaches, adjusted to the particular needs of AI workflows.

# **Chapter 2: Project Management Approach**

## **2.1 Traditional - 'Waterfall' Approach**

The 'waterfall' approach has been the dominant PM method for at least the past five decades. It was initially designed by Royce (1970) for managing large-scale, costly projects, where the prime indicator of **success** is balancing the 'iron triangle' of time, cost, and scope. In projects like those that inspired the development of this approach, **requirements** are exhaustively defined beforehand. This is also what legitimizes the waterfall approach's main **assumption** that project risk decreases over time (Vial et al., 2023).

## **2.2 Agile Approach**

At the turn of the century, technological advancements led to improvements in the stacks of developers, with the profession itself also becoming more widely adopted. This led to a significantly increasing number of new software projects around the world. It was then that software practitioners started noticing the flaws of Waterfall as a one-size-fits-all approach in software development.

In 2001, Beck et al. published the ‘Agile Manifesto’, stressing the need for the prioritization of iterative improvements and stakeholder engagement over contract negotiation and rigid planning. The main assumption informing this view had to do with the fact that technical requirements often change rapidly, and the technological viability of proposed pathways is not always guaranteed. Thus, success is defined as iteratively delivering software solutions that provide value to the customer.

## **2.3 The Case for a Hybrid Approach**

The emergence of Machine Learning (ML) techniques as core components of commercial applications has rendered the strict adherence to both previous models irrelevant. The development of such components often comes with sequential dependencies that often stall progress or even cause regressions back to previous stages, with feedback loops being constrained only among the Data Science team. During the AI development process, stakeholders becomes increasingly disengaged, as the expertise involved is very niche and advanced. This also results in the inability of the Data Science team to provide tangible, business-focused results to customers in regular loops.

## **2.4 The Helios Hybrid Approach**

The previous discussion highlights the case for a hybrid approach, drawing characteristics from both traditional and agile techniques, while catering to the intricate needs and characteristics of AI projects.

The framework we propose, following the arguments of Vial et al. (2023), stretches across five linear phases. These phases are distinguished between them by waterfall-inspired gate-keeping, yet allow for the adoption of agile approaches for the underlying tasks. Such approaches have been appraised by researchers like Gemino et al. (2021) to “significantly increase stakeholder success over traditional approaches while achieving the same budget, time, scope, and quality outcomes”.

The proposed phases have as follows:

1. The Ideation Phase
2. The Blueprint Phase
3. The Proof of Concept Phase
4. The Minimum Viable Product Phase
5. The Maintenance & Calibration Phase

# **Chapter 3. Project Phases and Timeline**

This section outlines the phased structure of the Predictive Maintenance project. The Predictive Maintenance project follows a hybrid methodology combining traditional stage-gated management, agile execution where applicable, and an AI-centric workflow characterized by iterative experimentation. The execution of this plan can follow one of two proposed scenarios:

* **Aggressive Scenario**: Optimized for speed, leverages offshore resources and parallel execution to reduce delivery time.
* **Conservative Scenario**: Optimized for cost-efficiency, relies solely on internal resources with staggered development to reduce budget impact.

**Phase 1: Ideation (Duration: 1–2 Weeks. Same in both scenarios)**

The primary objective of the Ideation phase is to define the business problem in concrete terms, assess the technical feasibility of the proposed solution, and ensure stakeholder alignment.

During this phase, workshops will be organized with key stakeholders across the operations, engineering, and maintenance teams to clarify business objectives and pain points. In parallel, technical leaders will evaluate whether existing infrastructure, data availability, and team capabilities are adequate for AI development.

Key activities include:

* Conducting an **AI readiness assessment**, covering data maturity, cloud infrastructure, and governance.
* Scoping for the predictive maintenance problem in alignment with measurable KPIs, such as reducing unplanned downtime or maintenance cost.
* Identifying relevant datasets (e.g., SCADA sensor data, historical fault logs, maintenance schedules).
* Facilitating stakeholder interviews to ensure cross-functional support and commitment.

The output of this phase is a **Go/No-Go recommendation**, supported by a preliminary business case and technical viability report.

Table : Summary showcasing No material difference in timeline or execution method for Phase 1.

|  |  |  |
| --- | --- | --- |
| Component | Aggressive Scenario | Conservative Scenario |
| Objective | Define business problem, assess feasibility | Same |
| Workshops | Scheduled rapidly with full stakeholder commitment | Scheduled based on availability; may extend slightly |
| Data Access | Fast-tracked agreements and assessments | Conducted in sequence with limited resource bandwidth |
| Output | Go/No-Go decision based on technical and business readiness | Same |

**Phase 2: Blueprint**

**Aggressive: 2 weeks**

**Conservative: 3 weeks**

The Blueprint phase is dedicated to developing a comprehensive technical design for data architecture, AI model, and system integration environment. This ensures all necessary foundations are laid before significant resource investment begins.

Activities will focus on:

* Designing **end-to-end data pipelines** including ingestion, cleaning, transformation, and feature engineering flows.
* Defining the **AI model architecture**, based on preliminary research and known failure patterns in wind turbine equipment.
* Selecting appropriate **cloud technology** and development environments (e.g., AWS, Azure, GCP), and provisioning cloud storage and computational resources.
* Conducting initial **data profiling and exploratory analysis** to assess data quality, volume, and patterns.
* Developing system architecture documentation, including proposed integration points with Helios’ existing IT systems.

Deliverables at this stage include a full **technical design document**, an initial **data readiness assessment**, and a resource allocation plan for the upcoming Proof-of-Concept.

Table : Aggressive scenario prioritizes speed via overlapping activities and upfront cloud provisioning.

|  |  |  |
| --- | --- | --- |
| Component | Aggressive Scenario | Conservative Scenario |
| Data Design | Performed in parallel with infrastructure setup | Staged and sequential; slower due to internal constraints |
| Cloud Provisioning | Proactive and fully provisioned upfront | Provisioned in phases to reduce cloud cost |
| Team Involvement | High availability of senior roles; support from India if needed | Limited bandwidth from senior roles; no offshore support |
| Output | Full technical architecture and resource plan | Same deliverables; may take longer to finalize |

**Phase 3: Proof of Concept (PoC)**

**Aggressive: 8 weeks (parallel tasks with India-based support)  
Conservative: 8–10 weeks (sequential internal development only)**

The PoC phase focuses on hands-on development of an initial predictive maintenance model using historical data. This phase incorporates AI workflow logic—allowing for experimental loops, hypothesis testing, and performance benchmarking—while using agile methods for weekly iteration and visibility.

Key tasks include:

* **Data Engineering**: Constructing the initial ETL pipelines and preprocessing workflows to prepare data for AI consumption.
* **AI Model Development**: Testing multiple machine learning algorithms (e.g., Random Forest, Gradient Boosting, Deep Learning) to identify the most promising candidates for fault prediction.
* **Experiment Tracking**: Implementing tools like MLflow to log model versions, hyperparameters, and evaluation metrics.
* **Team Collaboration**: Coordinating between internal teams and offshore partners (if applicable under the aggressive scenario), including 2 India-based AI Engineers for a 4-week burst of parallel experimentation.
* **Progress Monitoring**: Holding sprint reviews to assess whether performance benchmarks and business-aligned metrics (e.g., prediction accuracy, recall on fault events) are being met.

The final deliverable for this phase is a **functional prototype**, which demonstrates the feasibility of applying machine learning to predict component failures with acceptable accuracy.

Table : Aggressive scenario uses offshore capacity to accelerate AI development and data engineering concurrently.

|  |  |  |
| --- | --- | --- |
| Component | Aggressive Scenario | Conservative Scenario |
| Data Engineering | Conducted by 2 Sr. Engineers + offshore (India) Junior Engineers | Conducted only by internal Sr. Engineer (half-time) |
| AI Development | Full AI team supported by 2 offshore AI Engineers for parallel prototyping | AI team only; no offshore help, slower iteration cycle |
| Feedback Loops | Weekly sprints with fast iteration, supported by robust tracking tools | Biweekly sprints, fewer parallel experiments |
| Oversight | AI Lead and Data Lead dedicate more time (~30%) | Reduced oversight due to workload on internal leads |
| Output | Functional prototype with metrics & performance benchmarks | Same prototype; potential delays due to internal resource limits |

**Phase 4: Minimum Viable Product (MVP) Development**

**Aggressive: 10 weeks**

**Conservative: 10–12 weeks**

Following a successful PoC, the MVP phase transitions the project from experimental prototyping to a usable, production-grade application.

This includes:

* **Model Integration**: Wrapping the validated AI model into a microservice or API that can be accessed by external systems.
* **Application Development**: Building the user interface and back-end systems necessary to interact with the prediction service. This may include dashboards for operations staff or automated alerts for maintenance teams.
* **Infrastructure Setup**: Establishing CI/CD pipelines to automate testing, deployment, and rollback. This includes containerization with Docker and orchestration with Kubernetes where appropriate.
* **Knowledge Transfer**: Conducting internal training, documentation sessions, and transition planning with Helios’ IT and operations teams.

The outcome of this phase is a **deployable, tested MVP** application integrated into Helios’ cloud infrastructure and accessible to designated operational staff.

**Phase 5: Maintenance and Calibration (Ongoing)**

After deployment, the application enters an ongoing maintenance phase to ensure long-term performance, reliability, and business alignment. AI models naturally degrade over time due to changing data patterns (model drift), making continuous monitoring essential.

Key components of this phase include:

* **Monitoring Infrastructure**: Implementing systems to track model accuracy, data drift, and application uptime (e.g., using Prometheus, Grafana).
* **Scheduled Retraining**: Setting up retraining workflows (e.g., via Apache Airflow or cron jobs) that periodically update the model using the latest data.
* **Customer Support**: Establishing a support process and ticketing system (e.g., Jira or Zendesk) for users to report issues or request enhancements.
* **Documentation and Handover**: Creating retraining guides, model cards, and access control policies to ensure governance and auditability.

This phase is critical to **sustaining business value** over time and aligning predictive insights with Helios’ evolving operational needs.

Table : Maintenance and Calibration Summary

|  |  |  |
| --- | --- | --- |
| Component | Aggressive Scenario | Conservative Scenario |
| Monitoring Setup | Full-stack monitoring implemented at MVP handoff | Phased setup post-MVP deployment |
| Retraining Workflow | Automated from the outset using Airflow or similar tools | Initially manual; automation added later |
| Support Infrastructure | Jira/Zendesk integration from go-live | Manual support handling at start, systemized later |
| Documentation | Delivered as part of MVP handover | Completed in parallel with first months of maintenance |

**Visual Representation**

A Gantt chart will be prepared to illustrate the following for each scenario:

* Task parallelism in the aggressive plan vs. sequential execution in the conservative scenario
* Use of offshore resources during the PoC in the aggressive plan
* Phase gate checkpoints for review and decision-making

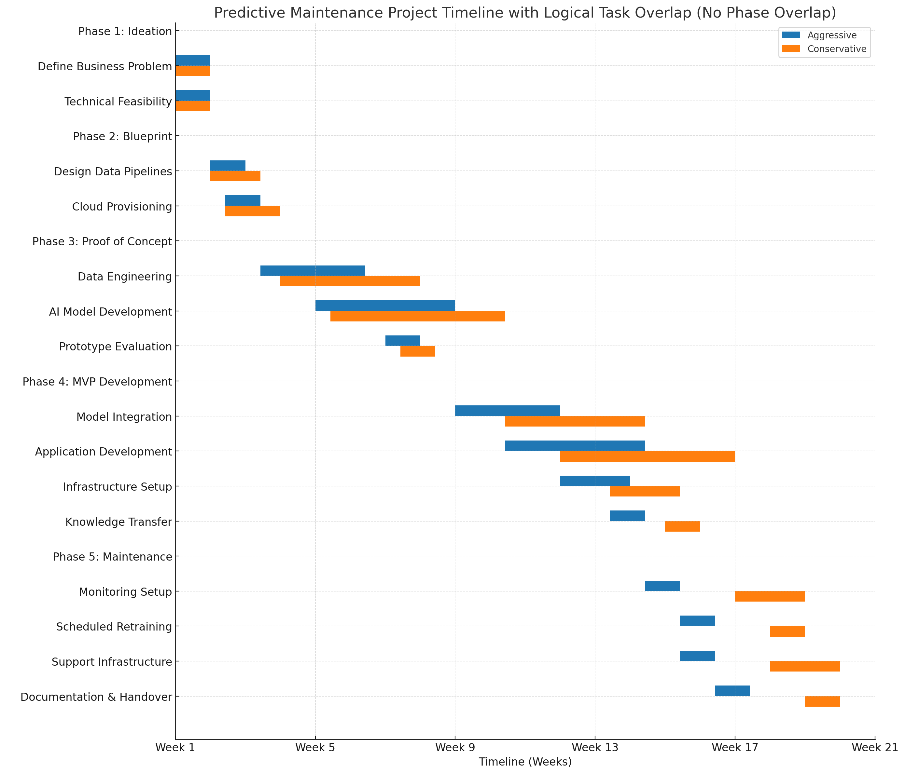


Figure : Gantt chart of the Predictive maintenance project concerning both Aggressive and Conservative approaches.

# **Chapter 4: Team Structure and Rationale**

## **4.1 Team Structure**

The team structure for the predictive maintenance project is designed to ensure technical excellence, operational efficiency, and business alignment.

**Data Team:**

* **Data Lead (10% FTE):** Oversees data governance and ties development to business needs.
* **Senior Data Engineer (50% FTE):** Builds and maintains pipelines and data workflows.
* **Junior Data Engineers:** Unfilled due to internal limitations; workload absorbed by senior staff.

**AI Team:**

* **AI Lead (20–30% FTE):** Leads model strategy.
* **Senior Data Scientist:** Develops and evaluates core ML algorithms.
* **Junior AI Engineers:** Assist in processing and model training.
* **AI/MLOps Engineer (80% FTE):** Ensures scalable deployment and automates workflows.
* **Offshore AI Engineers:** Deployed for 4 weeks in the aggressive scenario for parallel prototyping.

**Systems Team:**

* **Two IT Developers (100% FTE):** Implement UI/backend systems and full integration

Each AI specialist is paired with a domain expert in a dual-role “AI Power Couple,” ensuring technical solutions stay aligned with business realities.

## **4.2 Alignment of FTEs with Project Requirements**

For the Data Team, unavailable juniors mean the Senior Data Engineer and Data Lead handle all work, 6 weeks (conservative) or 4 weeks (aggressive) with higher intensity; cloud costs are included. For the AI Team, FTEs match requirements. The conservative scenario uses only internal staff (AI Lead 20%), while the aggressive scenario adds 2 offshore juniors (AI Lead 30%) and the AI/MLOps stays at 80% FTE and cloud costs are covered. Systems Team has 2 IT Developers full-time for 10 weeks in both cases, meeting requirements.

Table : Summary Table of FTEs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Duration (weeks) | Key Roles & FTEs (as per info) | Reported FTEs & Timeline | Alignment |
| Data Prep | 4–6 | Data Lead (10%), Sr. Data Eng. (50%), 2 Jr. Data Eng. (n/a) | 6 wks (conservative), 4 wks (agg.) | Yes (w/ constraints) |
| AI/Modeling | 8 | AI Lead (20–30%), Sr. Data Scientist, Jr. AI Eng., MLOps (80%), 2 offshore (4 wks, if needed) | 8 wks, matching FTEs and offshore use | Yes |
| Systems Build | 10 | 2 IT Developers (100%) | 10 wks, 2 FTEs | Yes |

The team, FTEs, and timelines align with lead inputs, with internal gaps and offshore support clearly addressed in both scenarios.

# **Chapter 5: Cost Analysis and FTE Usage**

A detailed cost analysis has been performed for both conservative and aggressive scenarios. The focus is on how Full-Time Equivalent (FTE) allocations drive resource usage and cost efficiency.

Table : Cost Breakdown Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phase | Scenario | Duration (weeks) | Internal Cost (€) | Offshore Cost ($) | Cloud ($) | Total ($) |
| Data Prep | Conservative | 6 | 5,700 | 0 | 2,400 | 8,670 |
| Data Prep | Aggressive | 4 | 3,800 | 0 | 1,600 | 5,780 |
| AI/Modeling | Conservative | 8 | 38,800 | 0 | 3,200 | 45,880 |
| AI/Modeling | Aggressive | 8 | 40,400 | 4,000 | 3,200 | 51,640 |
| Systems Build | Conservative | 10 | 25,000 | 0 | 4,000 | 31,500 |
| Systems Build | Aggressive | 10 | 25,000 | 0 | 4,000 | 31,500 |

**Note:**  
Total ($) = (Internal Cost (€) × 1.1 [assumed EUR/USD exchange rate]) + Offshore Cost ($) + Cloud ($).

**Total cost from the aggressive approach:** $ 88,920

**Total cost from the conservative approach:** $ 86,050

**FTE and Resource Utilization**

* **Data Phase:**
  + Conservative: Senior Data Engineer at 50% FTE, Data Lead at 10% FTE, over 6 weeks. The absence of Junior Data Engineers increases the senior team’s workload, reflected in higher internal costs.
  + Aggressive: Compressed to 4 weeks with the same FTE allocation, reducing costs but increasing intensity.
* **AI/Modeling Phase:**
  + Conservative: 2 Junior AI Engineers, Senior Data Scientist, AI/MLOps Engineer (80% FTE), AI Lead (20% FTE) over 8 weeks. No offshore resources, maximizing internal utilization.
  + Aggressive: AI Lead increases to 30% FTE, and 2 offshore Junior AI Engineers are added for 4 weeks, raising costs but enabling faster delivery and reduced risk of bottlenecks.
* **Systems Build Phase:**
  + Both scenarios: 2 IT Developers full time for 10 weeks, ensuring the application is robust and delivered within the planned timeframe.

**Cloud cost** of$400 per week in all phases, making efficient scheduling essential for cost control.

**Scenario Comparison**

* **Conservative Scenario:** Focuses on cost efficiency by using internal resources and longer timelines. FTEs are managed to minimize spend, trading off speed.
* **Aggressive Scenario:** Prioritizes faster delivery through higher FTE involvement and offshore support, increasing costs but enabling earlier business impact.

The team structure and FTE allocations are tailored for each project phase, balancing expertise, cost, and timelines. The cost analysis highlights trade-offs between conservative and aggressive approaches, enabling Helios to choose based on its priorities.

# **Chapter 6: Phase Gates and Risk Assessment**

Effective risk management is paramount to ensuring the Predictive Maintenance project delivers its promised benefits on schedule, within budget, and in compliance with Helios's quality standards. This chapter identifies key project risks and outlines targeted mitigation strategies, while establishing formal phase gates to control uncertainty and guarantee ongoing alignment with strategic objectives.

## **6.1 Technical and Operational Risks**

**Data Quality and Acquisition**: The burden on the senior data engineer (given the absence of junior data engineers in the team) may lead to delays in extracting and transforming heterogeneous data sources. Mitigation includes implementing an automated data validation framework within the initial two weeks and engaging external ETL consultants if error rates (as discussed in section 7.2) exceed 10%.

**Model Performance**: Algorithmic prototypes may underperform on edge cases or fail to generalize across turbine types. Mitigation involves adopting modular experimentation pipelines using containerized environments and reserving 15% of the AI Lead's capacity for independent model validation audits.

**Deployment Scalability**: Unexpected latency or resource contention in cloud environments could impair real-time predictions. Load testing at 1.5× expected production throughput will be conducted, with latency beyond 200ms triggering architecture reviews.

**Resource Constraints**: Overreliance on offshore junior engineers introduces communication delays and quality variability. Daily overlapping stand-ups and shared coding standards will maintain consistency across geographies.

## **6.2 Phase Gates**

At the conclusion of each phase, formal decision gates will be used to control uncertainty and guarantee ongoing alignment with strategic objectives:

**1. Ideation Phase**: Evaluate data access, problem framing, and business alignment.  
**2. Blueprint Phase**: Verify data preparedness, architecture viability, and technical design.  
**3. PoC Phase**: Assess the viability of the prototype and model performance.  
**4. MVP Phase**: Verify organizational and production readiness for deployment.

According to the practice of the 'waterfall' approach (Chapter 1), the project's scope will be reviewed or abandoned if any of the phase gates does not meet any of the acceptance criteria. Phase results will be assessed on three critical dimensions:

* **Quality of Execution**: Work meets Helios's technical standards and generates high-quality outcomes
* **Business Rationale**: Deliverables align with strategic objectives and demonstrate ongoing value
* **Resource Availability**: Adequate financial and human resources exist to complete subsequent phases

Gate decisions result in one of four outcomes: **Go** (proceed to next phase), **Kill** (terminate project), **Hold** (pause for resource reallocation), or **Recycle** (return to current phase for improvements).

## **6.3 Governance and Compliance**

**Stakeholder Adoption**: Resistance to new predictive maintenance tools will be addressed through "AI Power Couple" liaisons in each business unit and biweekly demonstrative updates.

**Regulatory Compliance**: GDPR and data privacy exposure will be mitigated through Data Protection Impact Assessments and compliance team reviews of all data workflows.

By proactively identifying risks and implementing structured phase gates, the Predictive Maintenance project is positioned to meet its objectives within defined timeframes and budget. The outlined mitigation strategies and gate criteria ensure Helios realizes operational efficiencies while maintaining quality control throughout the project lifecycle.

# **Chapter 7: Progress Monitoring & Success Metrics**

To track the progress of the Helios ‘Predictive Maintenance’ project, we propose a monitoring system based on Key Performance Indicators (KPIs) inspired by the traditional "iron triangle" of time, cost, and scope, expanded by AI-centric metrics.

The need for this was established by the proposition of Vial et al. (2023), that AI project monitoring must not be constrained to typical software parameters. The reason is that they rely on data quality and specialized expertise, and that the business relevance of their output is often less transparent. In the case of the hybrid approach we suggest for Helios, this is manifested in terms of:

* Direct measures of data and model integrity
* Indirect indicators of downstream business impact
* Adherence to a disciplined Scrum cadence

All of these can be surfaced in a unified, real-time dashboard, as proposed by multiple authorities in the field (Google Cloud, 2024).

## **7.2 Direct Metrics**

Based on Multimodal.dev (2024), data quality trends need to be quantified based on:

* completeness: the percentage of required data fields populated,
* integrity: estimated in terms of schema-violation rates,
* uniqueness: duplicate record proportion, and
* accessibility: API latency and uptime

Model performance in the Helios case is gauged using classification scores (accuracy, precision, recall, and F1-score) alongside operational measures such as:

* response time: the time it takes the AI model to deliver results after receiving an input
* throughput under load: the number of tasks an AI system can process in a specific time frame (e.g., 5 minutes)
* error-rate monitoring: the proportion of errors out of the total number of classification/prediction attempts

Privacy/security compliance needs also to be ensured through automated vulnerability scans (Google Cloud, 2024).

## **7.3 Indirect Metrics**

I**ndirect metrics** complement direct metrics by translating technical outputs into business value. Chief among these is:

* The percentage reduction in unplanned maintenance incidents post-deployment, which directly reflects the model’s impact on operational uptime,
* Technician-hour savings: aggregate technician hours saved via predictive alerts, and
* An end-user effort score derived from stakeholder surveys.

Latency from anomaly detection to actionable alert needs to also be measured, ensuring that our predictive insights reach the relevant teams in time to intervene (Murphy, 2024).

## **7.4 Tracking Agile Cadence**

Even within a hybrid framework, maintaining an Agile rhythm during the Proof of Concept and MVP phases is vital. Daily stand-ups surface data-preparation and training blockers, while sprint reviews track incremental improvements against benchmarks (Google Cloud, 2024; Murphy, 2024). Live updates from these 'agile rituals’ will feed a real-time dashboard combining:

* Sprint burndown (time): remaining story points vs. days,
* Budget burn-rate (cost): percentage overrides based on budget forecasts, and
* Feature completion (scope).

We suggest that following this progress monitoring and tracking approach will provide Helios with a transparent path to attaining both its technical goals as well as retrieve the desired measurable business return (Vial et al., 2023; Multimodal.dev, 2024).

# **Chapter 8: Conclusion**

The Helios Predictive Maintenance project demonstrates effective project management by combining waterfall-style phase gates with agile practices. This hybrid approach ensures structured progress while allowing flexibility during complex phases, such as Proof of Concept and MVP development. Phase gates provide clear checkpoints for alignment and risk control, while agile methods allow rapid response to challenges and continuous improvement. By modeling both conservative and aggressive implementation scenarios, Helios can strategically balance speed, cost, and resources. Rigorous KPI tracking and a well-defined team structure further support accountability, transparency, and adaptability. These techniques together reduce project risks and support successful AI deployment within a dynamic industrial environment.

# **References**

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development.<http://agilemanifesto.org/>

Gemino, A., Reich, B. H., & Serrador, P. M. (2021). Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? *Project Management Journal, 52*(2), 161–175. doi: <https://doi.org/10.1177/8756972820973082>

Google Cloud. (2024). *KPIs for gen AI: Measuring your AI success – deep dive*. Google Cloud Blog. Retrieved from<https://cloud.google.com/transform/gen-ai-kpis-measuring-ai-success-deep-dive>

Multimodal.dev. (2024, August 5). *34 AI KPIs: The Most Comprehensive List of Success Metrics*. Retrieved from<https://www.multimodal.dev/post/ai-kpis>

Murphy, J. (2024, October 16). *How businesses can measure AI success with KPIs*. Search Enterprise AI. Retrieved from<https://www.techtarget.com/searchenterpriseai/tip/How-businesses-can-measure-AI-success-with-KPIs>

Royce, W. W. (1970, August). *Managing the development of large software systems*. In *Proceedings of the IEEE WESCON* (Vol. 26, pp. 1–9). Retrieved on July 13, 2025 from<https://www.praxisframework.org/files/royce1970.pdf>

Vial, G., Cameron, A.-F., Giannelia, T., & Jiang, J. (2023). *Managing artificial intelligence projects: Key insights from an AI consulting firm*. Information Systems Journal, 33(3), 669–691.<https://doi.org/10.1111/isj.12420>