MAST

Managing Sustainability Tradeoffs
Grant Agreement No.: 22035

D3.1 - Quality model and catalog of metrics per model with definitions and usage guidelines

---

Document Number	D3.1
Document Title	Quality model and catalog of metrics per model with definitions and usage guidelines
Version	0.5~~1~~
Status	Final (Subject to Some Revisions)
Work Package	WP3
Deliverable Type	Report
Contractual Date of Delivery	March 2025 – May 2026
Actual Date of Delivery	2nd June 2026
Responsible Unit	CLW (task leader)
Contributors	All partners
Keyword List	Quality model, catalog of metrics, usage guidelines
Dissemination level	Public

Amendment History

Version	Date	Author (Unit)	Description
0.1	03/2026	CLW	Initial Draft with Template
0.2	04/2026	CLW	Partners contribuition
0.3	05/2026	CLW	Model development
0.4	05/2026	CLW	Formatting issues
0.5	06/2026	CPP	.dotx → .docx format

Executive Summary

This deliverable presents the sustainability quality model and associated catalog of metrics developed within WP3 to support the assessment of sustainability across the project use cases. The work addresses the growing need for systematic approaches capable of evaluating both technical and environmental dimensions of digital systems, particularly in contexts where software-intensive solutions directly influence energy consumption, operational efficiency, and sustainability-related decision-making.

The proposed framework combines established Software Engineering quality principles with sustainability-oriented evaluation perspectives. The methodology adopted in this work is based on the Goal–Question–Metric (GQM) paradigm, ensuring traceability between project objectives, evaluation questions, and measurable indicators. The resulting model provides a harmonized structure capable of supporting heterogeneous project scenarios while preserving comparability across pilots. The quality model is organized into two main domains—Technical and Environmental—complemented by a cross-domain perspective that captures optimization trade-offs between system performance, operational efficiency, and environmental impact. Within these domains, sub-qualities and metrics were defined to evaluate aspects such as data reliability, AI performance, energy consumption, carbon emissions, waste reduction, monitoring capabilities, maintainability, reporting auditability, and sustainability-aware operational strategies.

The metric catalog was constructed through the consolidation and harmonization of partner contributions, pilot KPIs, industrial practices, and sustainability-oriented references from Green Software Engineering and Green AI initiatives. Rather than introducing entirely new indicators, the work focused on organizing existing measurement practices into a coherent and interpretable framework applicable to real pilot environments.

Three use cases were addressed within the deliverable: the Cleanwatts use case, focused on smart metering and AI-driven forecasting in energy communities; the Wirtek use case, centered on sustainable software operations and carbon-aware execution strategies; and the Canon use case, dedicated to sustainability optimization in industrial printing systems. For each use case, the document defines dedicated quality models and associated metric catalogs adapted to their technical and operational contexts.

The resulting framework establishes the foundation for subsequent WP3 activities related to sustainability monitoring, optimization, benchmarking, visualization, and trade-off analysis. By linking sustainability objectives, operational KPIs, and measurable indicators within a common structure, the deliverable contributes a practical basis for sustainability-aware assessment and decision-making throughout the project lifecycle.

Table of Contents

1 Introduction

Sustainability has become a critical dimension in the design, operation, and evaluation of digital systems, particularly in contexts where software-based services directly influence energy consumption, resource usage, and operational decision-making. In such scenarios, assessing system quality can no longer be limited to traditional performance or maintainability concerns alone; it must also account for environmental impact, efficiency trade-offs, and the capacity of systems to support more sustainable behaviours over time.

Within this context, D3.1 addresses the need for a structured and operational framework capable of evaluating sustainability across the project use cases. The work presented in this deliverable focuses on the definition of a sustainability quality model and an associated catalog of metrics that together enable the systematic assessment of both technical and environmental dimensions of the project solutions. The objective is not only to identify relevant indicators, but also to organise them in a coherent way that supports interpretation, comparison, and practical use in pilot environments.

The proposed quality model provides a common structure to capture how different aspects of system behaviour—such as data reliability, energy efficiency, waste reduction, monitoring capacity, or decision support—contribute to sustainability objectives. At the same time, the catalog of metrics translates this conceptual structure into measurable indicators that can be applied within concrete use cases and pilot activities. In this sense, the model and the metric catalog are complementary: one provides the quality perspective through which sustainability is analysed, while the other provides the quantitative basis for evaluating it in practice.

A central challenge addressed in this work is the heterogeneity of the project scenarios. The use cases involved in the project differ significantly in terms of technical context, operational constraints, and sustainability priorities. For this reason, the methodology did not aim to produce a rigid or one-size-fits-all model, but rather a harmonised framework capable of accommodating different sustainability concerns while preserving consistency across the project. This enables the comparison of results across pilots while still respecting the specificities of each solution domain.

The resulting quality model and metric catalog thus establish the foundation for subsequent activities related to sustainability monitoring, optimisation, benchmarking, and trade-off analysis. By linking project objectives, pilot KPIs, and measurable indicators within a common structure, D3.1 contributes a practical and interpretable basis for sustainability-aware assessment and decision-making throughout the project lifecycle.

2 Methodology

This section describes the methodology followed in WP3 for the definition of the sustainability quality model and the associated catalog of metrics. The adopted approach combines established Software Engineering quality practices, sustainability-oriented frameworks, scientific literature, and contributions collected from project partners and pilot activities.

The objective of this methodology is to ensure that the proposed metrics are aligned with project objectives and KPIs, applicable to real pilot environments, and suitable for supporting sustainability-aware assessment and decision-making. Rather than creating an entirely new set of metrics, the work focused on consolidating, harmonizing, and structuring existing indicators within a common framework capable of supporting comparison and analysis across different use cases.

The methodology was structured around three main activities: the identification of metrics from multiple sources, the organization of these metrics using the Goal–Question–Metric (GQM) paradigm, and the collection and harmonization of partner contributions through a structured template.

2.1 Sources of Metrics

The metrics considered in D3.1 were primarily derived from partner contributions and practical industrial experience, complemented by insights from grey literature and sustainability-oriented practices applied in real operational contexts. This approach ensured that the selected metrics reflect not only theoretical considerations but also the actual measurement needs and constraints observed in the project pilots.

In particular, a significant portion of the metric catalog originates from existing pilot KPIs, internal monitoring practices, and experimental setups defined by project partners. These contributions provided concrete indicators already used in practice, covering aspects such as system performance, energy consumption, carbon impact, and operational behavior.

Grey literature sources, including technical reports, industry guidelines, and best practices in areas such as Green Software Engineering, AI sustainability, and energy management systems, were also considered to complement partner inputs. These sources were particularly relevant for identifying metrics related to energy efficiency, carbon awareness, and resource optimization that are not always formally standardized, but are widely adopted in industrial contexts.

Rather than introducing entirely new indicators, the process focused on consolidating and harmonizing these heterogeneous sources into a consistent and structured set of metrics. This ensured both alignment with the specific objectives of each use case and comparability across different pilots, while maintaining relevance to real-world deployment scenarios.

2.2 Goal–Question–Metric (GQM) Approach

The sustainability quality model defined in WP3 follows the Goal–Question–Metric (GQM) paradigm. The GQM approach was originally proposed by Basili and Weiss as a structured methodology for defining and interpreting software metrics in Software Engineering projects (Basili & Weiss, 1984). Initially applied in NASA projects, the methodology has since been widely adopted in several domains for quality assessment and measurement activities (Basili et al., 1994).

The GQM paradigm is based on the principle that metrics should not be collected in isolation, but instead directly linked to project objectives and evaluation needs. The methodology establishes a structured relationship between goals, the questions required to evaluate those goals, and the metrics used to answer those questions.

The approach is organized into three levels. At the conceptual level, goals define the objectives to be achieved, typically associated with project KPIs, quality requirements, or sustainability objectives. At the operational level, questions are formulated to characterize and evaluate whether those goals are being achieved. Finally, at the quantitative level, metrics provide measurable indicators capable of answering the defined questions (Basili et al., 1994).

The adoption of GQM within WP3 ensures traceability between project objectives, pilot requirements, and collected metrics. It also facilitates harmonization across different use cases and supports future activities related to optimization, benchmarking, visualization, and trade-off analysis.

Within the project, pilot KPIs and sustainability objectives are interpreted as goals, which are then refined into evaluation questions and associated metrics. For example, a goal related to reducing the environmental impact of AI optimization services may lead to questions regarding energy consumption during model execution, which can then be evaluated using metrics such as AI energy consumption or carbon emissions.

The use of GQM therefore provides a coherent and interpretable structure for organizing the sustainability quality model and associated catalog of metrics, as the following image demonstrates.

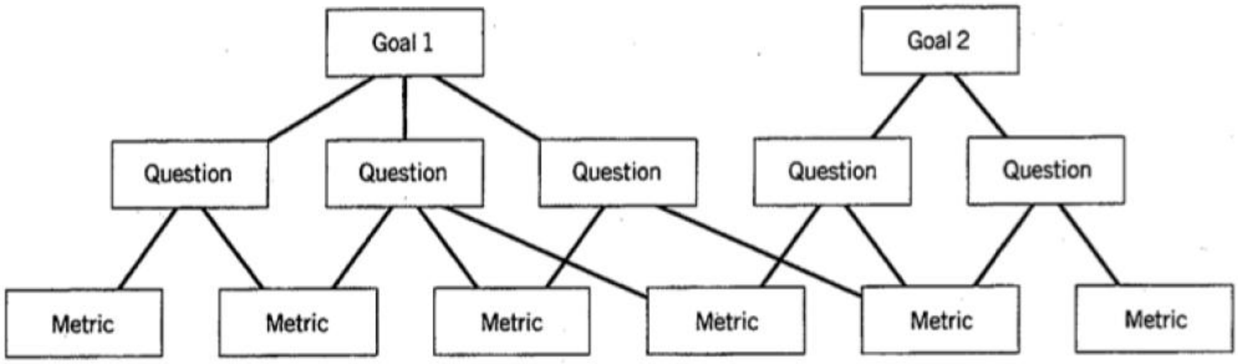


Figure 1: GQM model hierarchical structure

2.3 Relation to Existing Standards and Frameworks

The WP3 quality model builds upon recognized Software Engineering quality standards and sustainability-oriented frameworks. In particular, the work is conceptually aligned with the ISO/IEC 25010 quality model (ISO/IEC, 2011), which defines widely adopted software quality characteristics and evaluation dimensions.

The technical sustainability dimension of the proposed model is closely related to several ISO/IEC 25010 characteristics, particularly maintainability. This dimension provides the basis for evaluating the long-term quality and operational sustainability of software systems.

At the same time, the proposed model extends traditional software quality approaches by incorporating environmental sustainability dimensions not explicitly covered by ISO/IEC 25010. These extensions are particularly relevant in the context of AI-based optimization systems and energy community platforms, where computational workloads and energy usage may significantly influence the sustainability of deployed solutions.

To address these aspects, the methodology also considered concepts and practices from Green Software Engineering, Sustainable Software Engineering, and Green AI frameworks (Naumann et al., 2011). In particular, the Green Software Foundation initiatives and Software Carbon Intensity (SCI) concepts provided relevant references regarding energy-aware and carbon-aware software evaluation.

As a result, the proposed quality model combines established Software Engineering quality principles with sustainability-oriented evaluation criteria capable of supporting the assessment of both technical and environmental aspects of the project solutions.

2.4 Metric Collection Process

To support the collection and harmonization of metrics across project partners, a structured Excel-based template was developed within T3.1. The objective of this template is to standardize metric collection while maintaining sufficient flexibility to accommodate the different project use cases and pilot requirements.

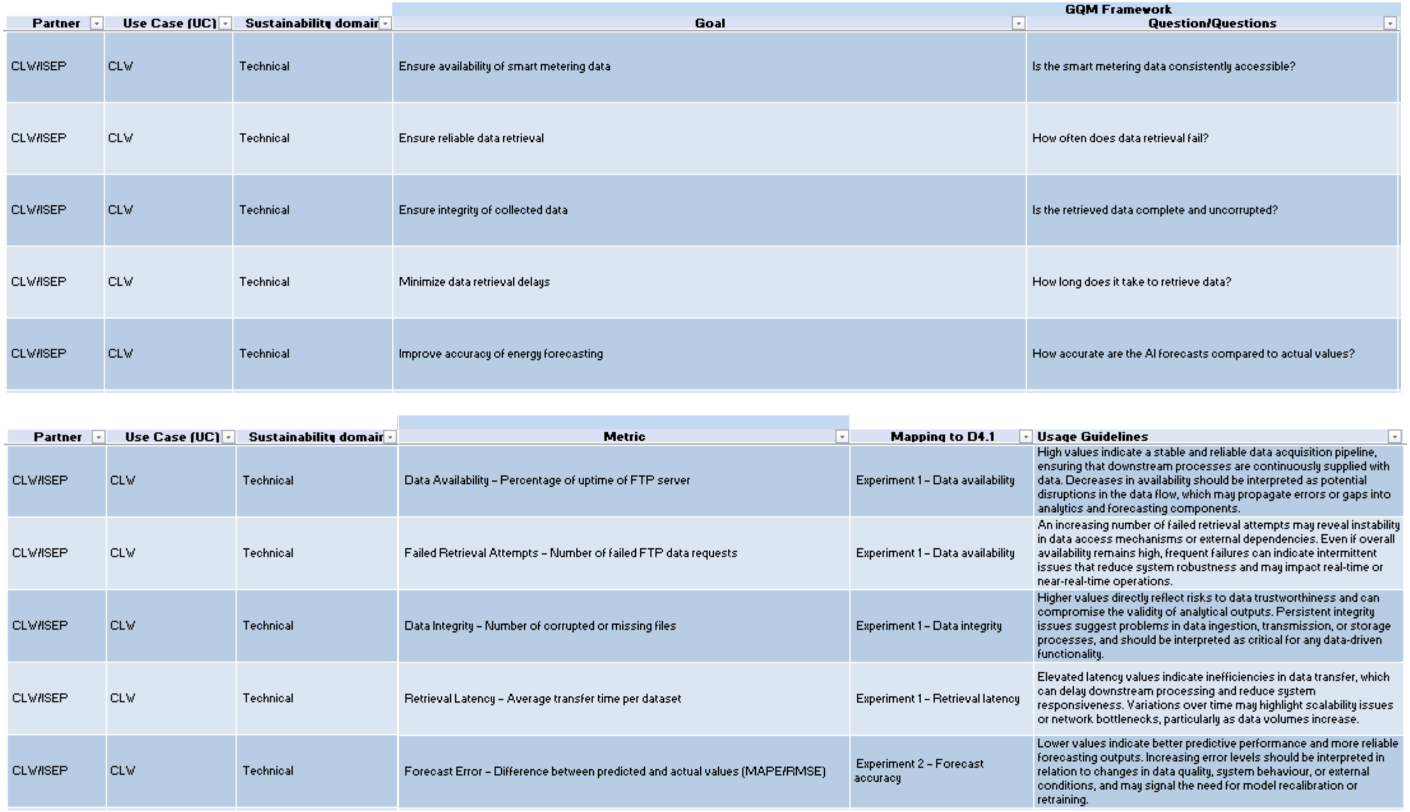
The template organizes each metric according to the Goal–Question–Metric structure and includes information regarding the associated partner, Use Case, sustainability domain, goal, evaluation question, metric definition, related KPIs, potential trade-offs, and mapping to existing pilot metrics.

The collection process involved iterative interactions with project partners, where metrics identified in pilot activities and experimental roadmaps were reviewed, refined, and aligned with the common methodology. Existing pilot metrics were reused whenever possible and reorganized according to the

D3.1 quality model structure. Additional sustainability-oriented metrics were introduced in cases where environmental aspects or AI sustainability concerns were not sufficiently covered by existing pilot documentation.

Attention was given to ensuring consistency of terminology, avoiding duplicated indicators, and identifying common sustainability dimensions across pilots. This process resulted in a harmonized catalog of metrics capable of supporting sustainability assessment and comparison across different project scenarios.

The resulting metric catalog constitutes the basis for subsequent WP3 activities related to sustainability evaluation, optimization, benchmarking, and visualization.



			GQM Framework	
Partner	**Use Case (UC)**	**Sustainability domair**	**Goal**	**Question/Questions**
CLW/ISEP	CLW	Technical	Ensure availability of smart metering data	Is the smart metering data consistently accessible?
CLW/ISEP	CLW	Technical	Ensure reliable data retrieval	How often does data retrieval fail?
CLW/ISEP	CLW	Technical	Ensure integrity of collected data	Is the retrieved data complete and uncorrupted?
CLW/ISEP	CLW	Technical	Minimize data retrieval delays	How long does it take to retrieve data?
CLW/ISEP	CLW	Technical	Improve accuracy of energy forecasting	How accurate are the AI forecasts compared to actual values?

Partner	Use Case (UC)	Sustainability domair	Metric	Mapping to D4.1	Usage Guidelines
CLW/ISEP	CLW	Technical	Data Availability – Percentage of uptime of FTP server	Experiment 1 – Data availability	High values indicate a stable and reliable data acquisition pipeline, ensuring that downstream processes are continuously supplied with data. Decreases in availability should be interpreted as potential disruptions in the data flow, which may propagate errors or gaps into analytics and forecasting components.
CLW/ISEP	CLW	Technical	Failed Retrieval Attempts – Number of failed FTP data requests	Experiment 1 – Data availability	An increasing number of failed retrieval attempts may reveal instability in data access mechanisms or external dependencies. Even if overall availability remains high, frequent failures can indicate intermittent issues that reduce system robustness and may impact real-time or near-real-time operations.
CLW/ISEP	CLW	Technical	Data Integrity – Number of corrupted or missing files	Experiment 1 – Data integrity	Higher values directly reflect risks to data trustworthiness and can compromise the validity of analytical outputs. Persistent integrity issues suggest problems in data ingestion, transmission, or storage processes, and should be interpreted as critical for any data-driven functionality.
CLW/ISEP	CLW	Technical	Retrieval Latency – Average transfer time per dataset	Experiment 1 – Retrieval latency	Elevated latency values indicate inefficiencies in data transfer, which can delay downstream processing and reduce system responsiveness. Variations over time may highlight scalability issues or network bottlenecks, particularly as data volumes increase.
CLW/ISEP	CLW	Technical	Forecast Error – Difference between predicted and actual values (MAPE/RMSE)	Experiment 2 – Forecast accuracy	Lower values indicate better predictive performance and more reliable forecasting outputs. Increasing error levels should be interpreted in relation to changes in data quality, system behaviour, or external conditions, and may signal the need for model recalibration or retraining.

Figure 2: Example of the template used in the metrics collection process

3 Quality Model

The proposed Quality Model for Sustainability is structured as a three-level hierarchical tree, following a top-down organization that aligns sustainability objectives with measurable indicators. The model is composed of:

i. domains,
ii. sub-qualities (or metric groups)
iii. metrics.

This structure ensures consistency with the GQM approach adopted in T3.1 and enables a systematic evaluation of both technical and environmental sustainability aspects.

At the highest level, the model is divided into two main domains: Technical and Environmental. The Technical domain captures the ability of the system to remain efficient, reliable, maintainable, and usable over time, while the Environmental domain focuses on minimizing the environmental impact of system operation, particularly in terms of energy consumption, carbon emissions, and resource efficiency.

Within each domain, a set of sub-qualities is defined to group related metrics according to common evaluation objectives. At the lowest level, metrics provide quantifiable measures linked to specific goals and questions, as defined in the GQM framework.

3.1 Cleanwatts Use Case (UC – CLW)

The CLW use case focuses on the monitoring and analysis of smart metering data within energy communities, with the objective of supporting data-driven decision-making through AI-based forecasting and optimization functionalities. This use case addresses key challenges related to the availability, integrity, and efficient use of energy data, as well as the performance of analytics tools and user-facing dashboards.

3.1.1 Technical domain

The Technical domain evaluates the system’s ability to ensure reliable operation, efficient performance, and effective use of data within the context of smart metering and AI-driven forecasting.

It’s important to note that while maintainability is typically considered the primary quality dimension directly associated with technical sustainability in software systems, the CLW use case requires a broader interpretation of the technical domain. In this context, several runtime qualities—such as reliability, performance, and data availability—are explicitly included, as they directly influence the effectiveness and sustainability of data-driven operations within energy communities. This extension is justified by the data-centric nature of the CLW scenario, where system value depends on the continuous availability, integrity, and timely processing of smart metering data. In such systems, runtime qualities are not only operational concerns but also determinants of sustainability performance, as failures in data access, delays in processing, or poor system responsiveness may compromise forecasting accuracy, reduce decision-making effectiveness, and ultimately lead to suboptimal energy use. For this reason, reliability- and performance-related metrics are considered integral to the technical sustainability of the CLW use case, complementing more traditional maintainability-oriented perspectives.

In line with this perspective, the sub-quality *Data Reliability and Availability* assesses the accessibility and integrity of smart metering data. It includes the following metrics: Data Availability, defined as the percentage of uptime of the FTP server; Failed Retrieval Attempts, representing the number of unsuccessful data requests; and Data Integrity, which measures the number of corrupted or missing files. Then, the sub-quality *System Performance and Efficiency* evaluates the responsiveness and execution speed of the system. This is measured through Retrieval Latency, defined as the average transfer time per dataset, and Dashboard Response Time, which captures the time required for dashboards to load or update following user interactions.

The *AI and Forecasting Quality* sub-quality focuses on the accuracy of predictive models used within the system. This is represented by the Forecast Error metric, which quantifies the deviation between predicted and actual values using indicators such as MAPE or RMSE. Also, the sub-quality *Processing Efficiency of AI* evaluates the computational performance of AI models, through the Processing Time metric, defined as the average runtime required to generate forecasts.

Finally, the sub-quality Data Utilization and Effectiveness assesses how efficiently collected data is leveraged within the system. This is captured through the Data Utilization metric, representing the percentage of usable versus missing data.

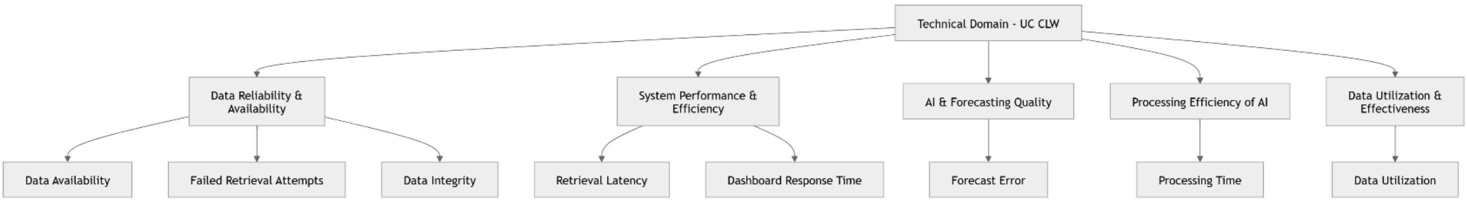


Figure 3: UC CLW – Technical Domain

3.1.2 Environmental domain

For the Environmental domain, the sustainability impact of system operation in terms of energy consumption, carbon emissions, and renewable energy integration is evaluated.

The following sub-qualities are present in the model:

- *Energy Consumption* measures the energy required to execute AI processes. It is captured through the *AI Energy Consumption* metric, which quantifies the energy used during model training and inference.

- *Carbon Emissions* evaluates the environmental footprint in terms of greenhouse gas emissions. This includes the *Carbon Emissions* metric, measuring the CO_2 emissions associated with system computation.
- *Renewable Energy Usage* assesses the extent to which locally generated renewable energy is used within the system. This is represented by the *Self-consumption Rate*, defined as the percentage of local energy consumption within the community.

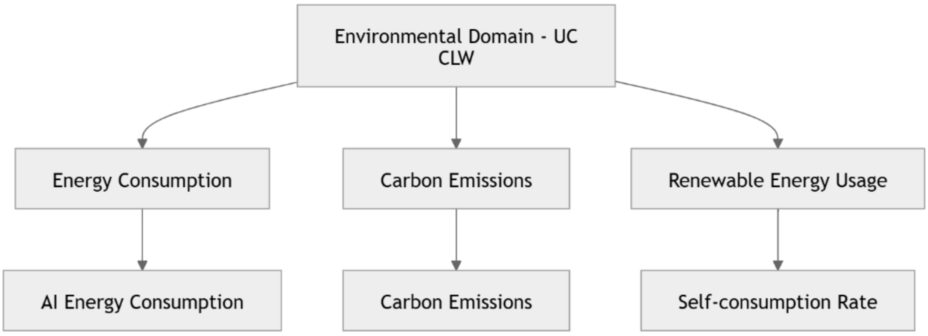


Figure 4: UC CLW – Environmental Domain

3.1.3 Cross - domain

The CLW use case also incorporates a cross-domain perspective, recognising the need to balance technical and environmental objectives.

The sub-quality *Trade-off and Optimization* evaluates how effectively the system balances competing criteria such as cost, performance, and environmental impact. This is captured through the *Trade-off Score* metric, which provides a composite indicator combining these dimensions.

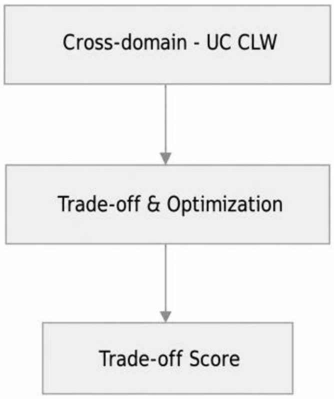


Figure 5: UC CLW – Cross-domain

3.1.4 Consolidated Quality Model

The quality model for the CLW use case integrates technical, environmental, and cross-domain perspectives into a unified framework, enabling a holistic evaluation of system performance and sustainability impact. The model is structured according to the two main domains of the project—Technical and Environmental—complemented by a cross-domain dimension that captures the interaction and trade-offs between these perspectives.

There's a deliberate prioritization of a data-centric system behaviour, where the overall quality of the solution is intrinsically linked to how effectively data is continuously accessed, processed, and transformed into actionable insights. Rather than treating data handling, analytics, and user interaction as separate layers, the model implicitly positions them as interdependent components of a single value chain.

Another important aspect is the inclusion of computational sustainability as an integral part of system evaluation. By incorporating environmental metrics alongside technical ones, the model reflects an understanding that digital solutions—particularly those relying on AI—introduce non-negligible resource consumption that must be monitored and optimized. This indicates a shift in evaluation criteria, where energy and carbon considerations are treated as inherent properties of system behaviour, rather than external constraints.

Furthermore, the presence of a cross-domain dimension highlights that optimization in this context is inherently multi-objective. The model does not assume that improvements can be made independently across dimensions; instead, it recognizes that system design and operation require continuous balancing of trade-offs.

Figure 6 and table 1 present the consolidated quality model for the CLW Use Case, integrating technical, environmental, and cross-domain perspectives.

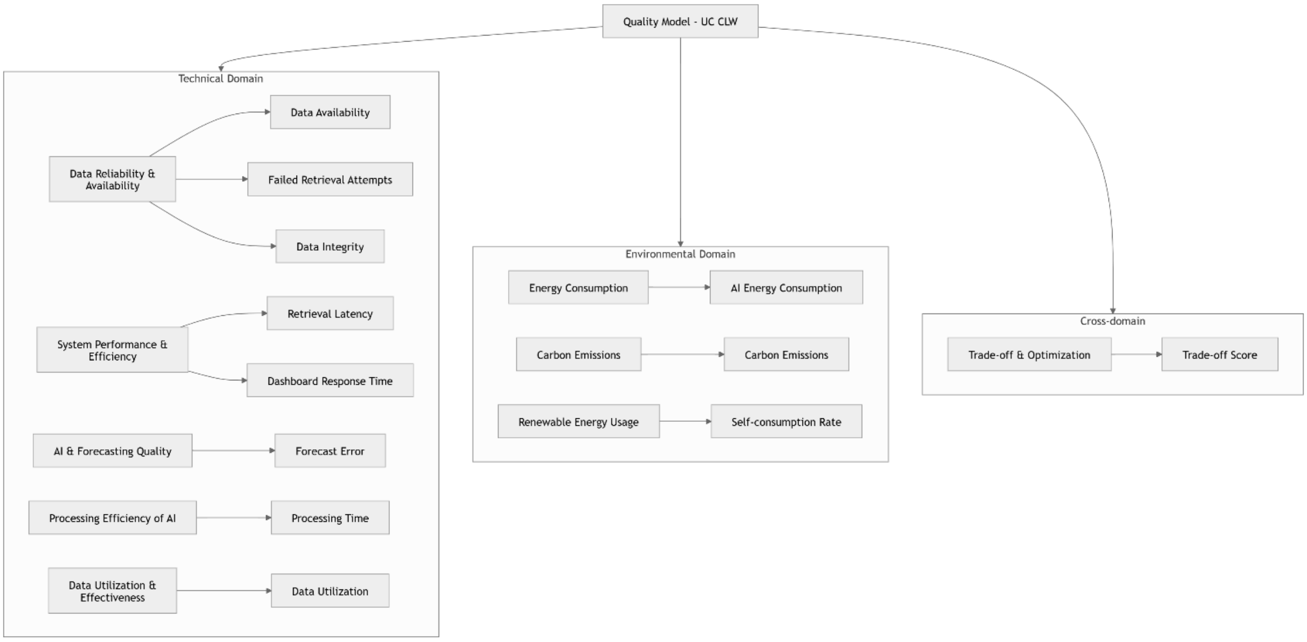


Figure 6: UC CLW – Consolidated Quality Model Diagram

Domain	**Sub-quality**	**Metric**
Technical	Data Reliability & Availability	Data Availability
Technical	Data Reliability & Availability	Failed Retrieval Attempts
Technical	Data Reliability & Availability	Data Integrity
Technical	System Performance & Efficiency	Retrieval Latency
Technical	System Performance & Efficiency	Dashboard Response Time
Technical	AI & Forecasting Quality	Forecast Error
Technical	Processing Efficiency of AI	Processing Time
Technical	Data Utilization & Effectiveness	Data Utilization
Environmental	Energy Consumption	AI Energy Consumption
Environmental	Carbon Emissions	Carbon Emissions
Environmental	Renewable Energy Usage	Self-consumption Rate
Cross-domain	Trade-off & Optimization	Trade-off Score

Table 1: UC CLW – Consolidated Quality Model Table

3.2 Wirtek Use Case (UC – WRT)

The WRT use case focuses on measuring and improving the sustainability of software operations on the *Wappsto* platform by combining code-level energy optimization, process-level regression control, and operational carbon reduction strategies. The use case targets both engineering decisions (e.g., refactoring and green-mode alternatives) and operational decisions (e.g., carbon-aware scheduling and geographical load shifting), ensuring that sustainability performance remains credible through lightweight instrumentation and auditable reporting.

3.2.1 Technical domain

For this UC, the technical domain is centered on translating sustainability from a measurement exercise into an operational capability that can be engineered, verified, and adopted. The metric set shows that the technical focus is not limited to "how much energy is used" but extends to whether the platform can continuously observe energy behaviour, *act* on it through mechanisms, and institutionalise it through engineering workflows and user uptake.

The sub-quality *Energy Measurement and Optimization* combines the baseline quantification of energy consumption with the evaluation of improvement mechanisms. It includes the metrics Module Energy Consumption, which provides the reference measurement of kWh per operation/function execution, and Green Mode Energy Reduction, which quantifies the percentage energy savings achieved when switching from performance mode to energy-efficient implementations. Together, these metrics enable the identification of high-impact modules and the assessment of whether optimisation strategies deliver meaningful improvements.

The sub-quality *Measurement Overhead Control* ensures that the act of measuring energy consumption does not invalidate the results. This is captured through the Telemetry Overhead metric, which quantifies the proportion of system resources consumed by instrumentation and monitoring processes.

The sub-quality *Energy Quality Assurance in CI/CD* evaluates how sustainability is enforced within the development workflow. It includes the CI/CD Energy Regression Rate metric, which measures the percentage of pull requests flagged for energy regressions before deployment, ensuring that efficiency improvements are preserved over time and preventing performance degradation.

The sub-quality *Trade-off Dashboard Readiness* captures whether sufficient system information is available to support decision-making. This is measured through Dashboard KPI Coverage, which reflects the extent to which dashboards expose relevant sustainability indicators for operational trade-offs.

The sub-quality *Customer Adoption of Green Features* evaluates the extent to which sustainability functionalities are used in practice. It is captured through the Green Feature Adoption Rate metric, which measures the proportion of eligible users enabling energy-efficient or carbon-aware features.

Finally, the sub-quality *ESG Reporting Auditability* evaluates the completeness and reliability of sustainability reporting. This is captured through ESG Report Completeness, which assesses whether emission reports are consistently produced with traceable and auditable data.

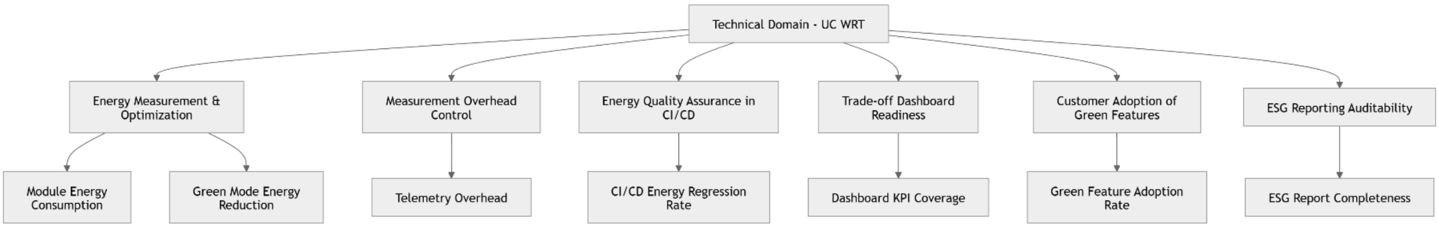


Figure 7: UC WRT – Technical Domain Diagram

3.2.2 Environmental domain

In the Environmental domain, the sustainability impact of software operations is evaluated by linking energy consumption with its carbon consequences and by assessing the effectiveness of strategies designed to reduce those impacts.

The sub-quality *Carbon Impact and Optimization* combines the attribution of carbon emissions with the mechanisms used to reduce them. It includes the metric *Operation Carbon Footprint*, which quantifies the CO_2 emissions associated with each operation by combining energy consumption with real-time grid carbon intensity, providing a direct measure of environmental impact. It also includes *Carbon Shift Savings*, which captures the emissions avoided by shifting workloads to lower-carbon periods, reflecting the effectiveness of carbon-aware scheduling strategies.

Together, these metrics provide a complete view of the environmental dimension: *Operation Carbon Footprint* identifies where emissions occur and under which conditions, while *Carbon Shift Savings* evaluates how much of that impact can be mitigated through operational decisions.

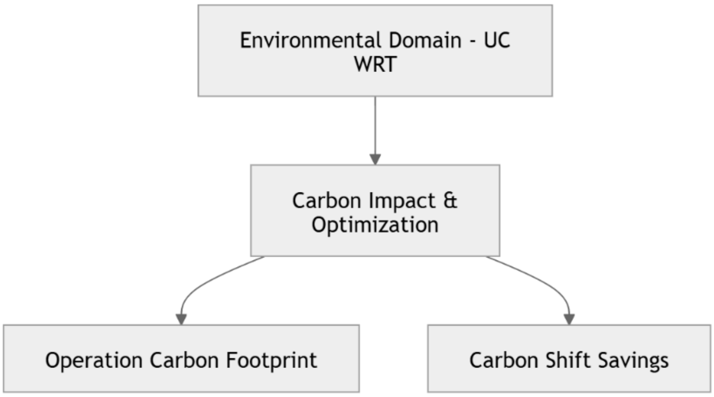


Figure 8: UC WRT – Environmental Domain Diagram

3.2.3 Cross-domain

The WRT use case also includes a cross-domain perspective to capture sustainability improvements that depend simultaneously on technical capabilities and environmental conditions.

The sub-quality *Geographical Load Shifting* evaluates the impact of redistributing workloads across locations with different grid carbon intensities. This is captured through the *Load Shift Carbon Impact* metric, which measures the percentage CO_2 reduction achieved through geographical workload redistribution based on grid intensity. This sub-quality reflects that certain optimization strategies cannot be assessed purely from a technical or environmental perspective in isolation. Instead, the effectiveness of geographical load shifting depends on both the system’s ability to route workloads across data centers and the variation in carbon intensity between regions.

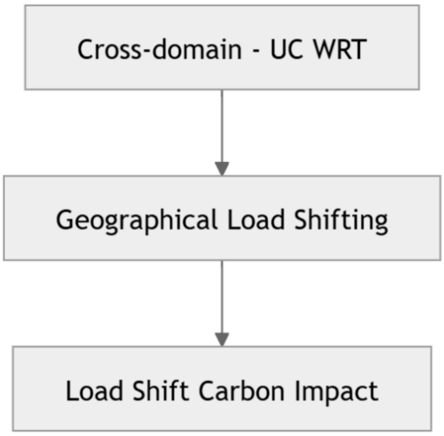


Figure 9: UC WRT – Cross Domain Diagram

3.2.4 Consolidated Quality Model

The quality model for the WRT use case integrates technical, environmental, and cross-domain perspectives into a unified framework that reflects the full lifecycle of sustainability management in software systems.

Rather than focusing on isolated performance indicators, the model captures a continuous process that starts with measurement, evolves through optimization and control, and is validated through operational adoption and reporting. At the core of this model is the combination of baseline quantification and improvement mechanisms: the ability to measure energy consumption at operation level provides the reference for identifying inefficiencies, while green-mode execution and scheduling strategies translate this knowledge into tangible reductions.

A distinguishing aspect of this model is the focus on ensuring that sustainability improvements are both credible and sustained over time, by embedding energy awareness directly into the development workflow and preventing regressions or distortions from measurement overhead. At the same time, the model recognizes that impact depends not only on technical optimization but also on decision support and user adoption, ensuring that sustainability features are both actionable and actively used. This perspective is further extended by the environmental and cross-domain dimensions, which link software execution to external conditions such as grid carbon intensity and geographical location, highlighting that sustainability performance is shaped not only by system design but also by when and where the system operates.

This integrated view can be visualized in figure 10 and table 2 below:

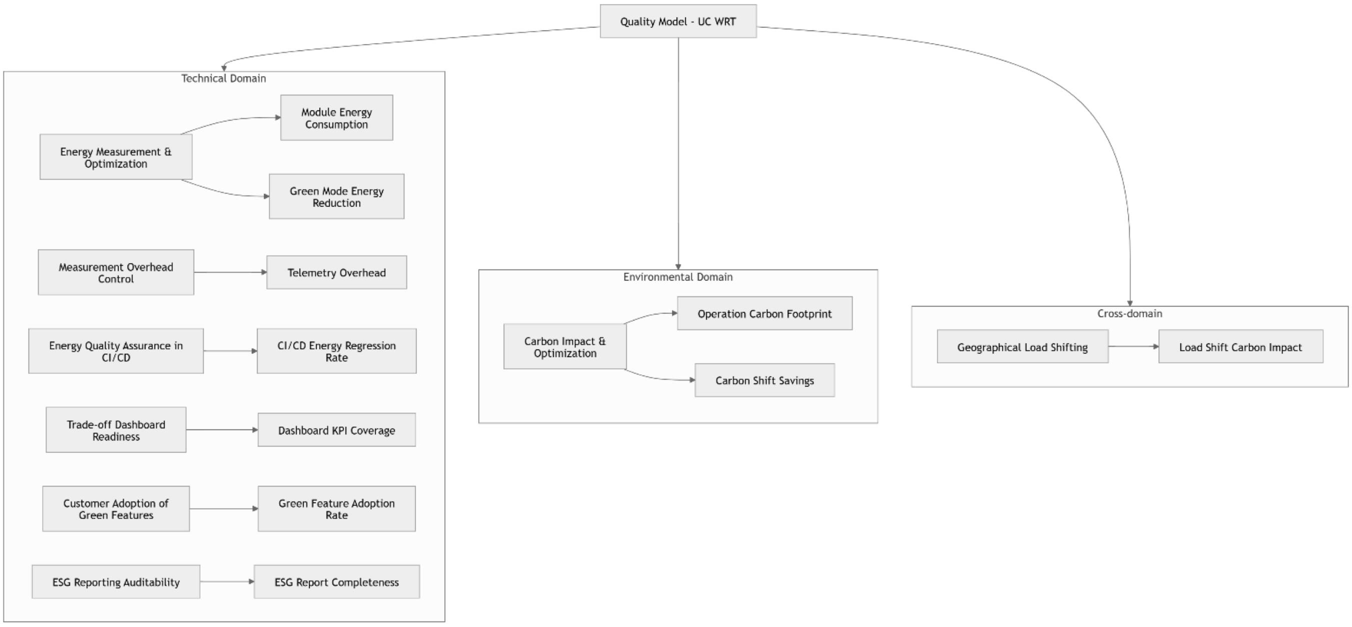


Figure 10: UC WRT – Consolidated Quality Model Diagram

Domain	Sub-quality	Metric
Technical	Energy Measurement & Optimization	Module Energy Consumption
Technical	Energy Measurement & Optimization	Green Mode Energy Reduction
Technical	Measurement Overhead Control	Telemetry Overhead
Technical	Energy Quality Assurance in CI/CD	CI/CD Energy Regression Rate
Technical	Trade-off Dashboard Readiness	Dashboard KPI Coverage
Technical	Customer Adoption of Green Features	Green Feature Adoption Rate
Technical	ESG Reporting Auditability	ESG Report Completeness
Environmental	Carbon Impact & Optimization	Operation Carbon Footprint
Environmental	Carbon Impact & Optimization	Carbon Shift Savings
Cross-domain	Geographical Load Shifting	Load Shift Carbon Impact

Table 2: UC WRT – Consolidated Quality Model Table

3.3 Canon Use Case (UC – CPP)

The CPP use case focuses on improving the sustainability and operational efficiency of industrial printing systems through a combination of energy optimization, waste reduction, and data-driven decision support. The objective is to enable adaptive system behaviors such as power state transitions and carbon-aware scheduling while providing actionable insights to both operators and end users via dashboards and diagnostics.

In this context, the use case addresses key challenges related to minimizing energy consumption and material waste, enhancing transparency through logging and reporting functionalities, and ensuring long-term system maintainability and traceability.

3.3.1 Technical domain

The Technical domain evaluates the system's ability to support sustainability-oriented decision-making, reporting, traceability, and long-term maintainability within the CPP use case.

The sub-quality *Sustainable Scheduling and Maintainability* captures the relationship between scheduling strategies and software quality. It is represented by *Scheduling mode availability; maintainability score*, which evaluates whether the implementation of sustainable scheduling modes improves sustainability outcomes without increasing long-term system complexity or technical debt.

The sub-quality *Upgrade Traceability* evaluates whether sustainability-related system changes can be tracked over time. This is measured through *% upgrades linked to metadata*, which reflects the extent to which upgrades are traceable via digital passport mechanisms and associated metadata.

The sub-quality *Role-specific Decision Support* assesses whether dashboards provide tailored and actionable sustainability insights to different user profiles. It includes the metric *Count of actionable insights per role*, which reflects the degree of personalisation and relevance of the information exposed to each role.

Finally, the sub-quality *Architectural Integrity and Technical Debt Control* evaluate the long-term maintainability and evolvability of the system through a set of architecture- and debt-related metrics. It includes *Number and severity of deviations*, which measure deviations from past architecture decisions; *Number and size of architecture smells*, which capture structural design issues; *Number and severity of self-admitted technical debt items***,** which reflect maintainability concerns explicitly recorded by developers; and *Number and severity of documentation debt items*, which assess problems in the completeness and standardisation of architectural decision documentation

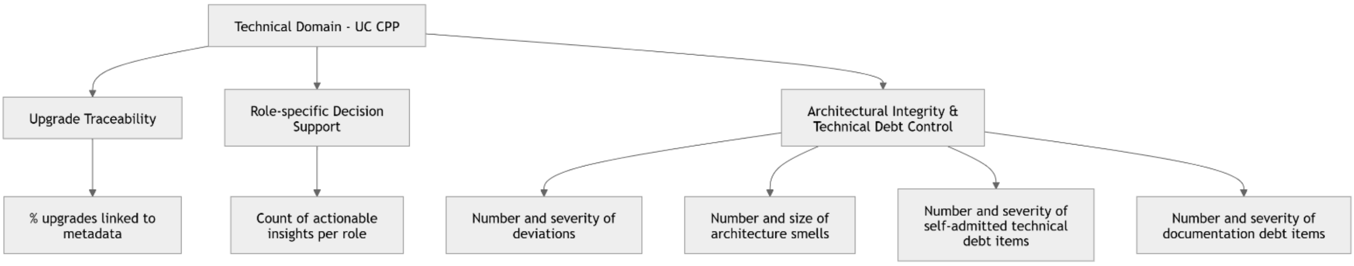


Figure 10: UC CPP – Technical Domain Diagram

3.3.2 Environmental domain

For the Environmental domain, the CPP use case evaluates the sustainability performance of printing systems in terms of energy efficiency, carbon-aware operation, material waste reduction, and energy monitoring capabilities.

The sub-quality *Power Efficiency through Adaptive Power States* assesses the impact of adaptive power-state transitions on energy consumption under realistic operating conditions. It includes the metrics *Average energy use per job; % reduction vs baseline***,** *Power/square meter printed/(scope of engines) in % versus some baseline (typical, customer, …)*, and *Maximum achievable reduction in %***,** which together provide a comprehensive view of energy performance, variability across systems, and achievable optimisation potential.

The sub-quality *Carbon-aware Scheduling* evaluates the effectiveness of scheduling strategies in reducing environmental impact. It is captured through *% of power-intensive jobs run in low-carbon windows***,** which indicates how successfully workloads are aligned with favourable carbon intensity conditions.

The sub-quality *Waste Reduction* focuses on minimising material waste through improved rejection strategies. It is represented by *% reduction in waste per job*, which reflects the effectiveness of balancing print quality and material efficiency.

The sub-quality *Fine-grained Energy Monitoring* evaluates the system's ability to provide detailed insight into how energy consumption is distributed across components over time. This is captured through *Energy consumption per system component*, supporting the identification of inefficiencies.

The sub-quality *Energy Consumption Prediction* evaluates the reliability of predictive models used for estimating power consumption. It includes *Prediction accuracy of the model*, which reflects how well predicted data approximates actual monitored energy usage.

Finally, the sub-quality *Dashboard Engagement and Reporting* assesses whether sustainability dashboards are effectively used and whether they support meaningful reporting functionalities. It includes the metrics *Dashboard sessions per customer; avg session duration*, which capture the level of customer engagement with dashboard functionalities, and *Number of reports generated; report completeness score*, which evaluate the system's capacity to support sustainability reporting in a complete and usable manner.

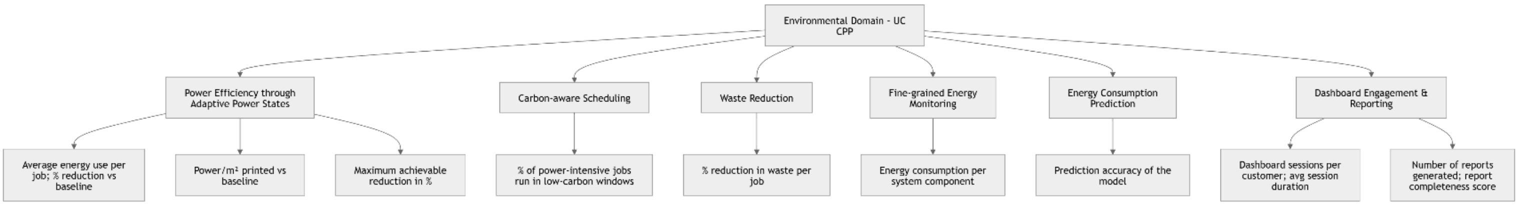


Figure 11: UC CPP – Environmental Domain Diagram

3.3.3 Cross-domain

The CPP use case includes a cross-domain perspective to capture sustainability insights that depend on the interaction between technical capabilities and environmental performance.

The sub-quality *Data Readiness and Logging Transparency* evaluates whether the information required to support sustainability analyses is available, structured, and understandable. At the same time, it reflects the complexity and transparency of the underlying data collection and logging mechanisms. It includes *Dashboard containing data readiness indicator; occupancy of tables*, which assesses the completeness and accessibility of logging information, and *Graphs that help to verify relationships between logging and system-level properties,* which support the identification of behavioural patterns and their relationship with system operation.

The sub-quality *Sustainable Scheduling and Maintainability* captures the trade-off between environmental optimisation strategies and long-term software maintainability. It is represented by *Scheduling mode availability; maintainability score*, which evaluates whether sustainability-oriented scheduling approaches can be implemented while preserving acceptable levels of maintainability and system evolvability.

The sub-quality *Waste Diagnostics and Analysis* evaluates the system's ability to identify, interpret, and benchmark waste-related behaviour through data-driven insights. It includes the metric *Waste per job trend; count of high-waste cases*, which supports the identification of high-waste patterns through dashboard diagnostics; *Data availability on environmental parameters*, which provides additional context for analysing waste behaviour under varying environmental conditions such as temperature and humidity; and *Benchmark figures for waste levels for similar jobs across the population (customer, MIF)*, which enable comparison across systems to identify inefficiencies and best practices.

Together, these metrics provide a comprehensive diagnostic view that links system operation, environmental conditions, and performance outcomes, supporting targeted interventions and optimisation strategies.

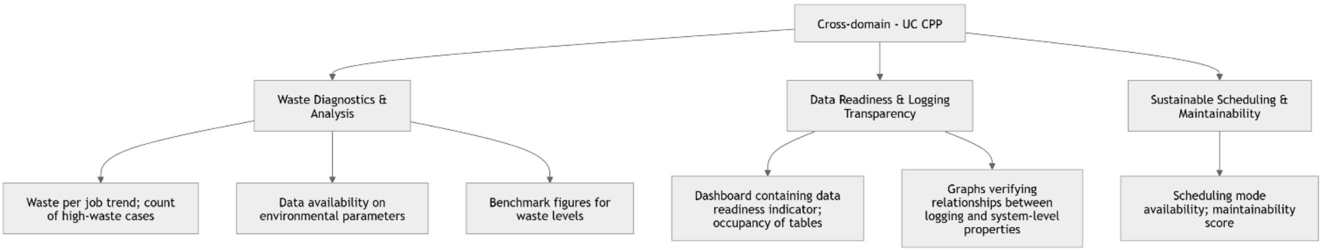


Figure 12: UC CPP – Cross-Domain Diagram

3.3.4 Consolidated Quality Model

The quality model for the CPP use case reflects the fact that sustainability in industrial printing systems emerges from the interaction between environmental optimisation mechanisms, technical maintainability concerns, and cross-domain capabilities that connect operational behaviour, monitoring infrastructures, and decision-support mechanisms. The model therefore explicitly recognises that several sustainability-related capabilities cannot be classified as purely technical or purely environmental.

The selected metrics suggest that performance in printing environments cannot be meaningfully assessed through average gains alone, since variability across jobs, devices, customer settings, and environmental conditions is itself part of what defines system quality. In this sense, the model implicitly treats sustainability as a question of controllability: it is not enough to reduce energy use or waste in isolated cases; the system must also make those improvements observable, explainable, and reproducible across contexts.

At the same time, the presence of maintainability, traceability, and debt-related metrics shows that short-term sustainability gains are being interpreted with caution, acknowledging that a solution cannot be considered genuinely sustainable if it achieves environmental improvements by increasing architectural fragility, reducing evolvability, or creating future operational burdens.

Figure 13 and Table 3 present the consolidated quality model for the CPP use case, illustrating the relationships between domains, sub-qualities, and their corresponding metrics.

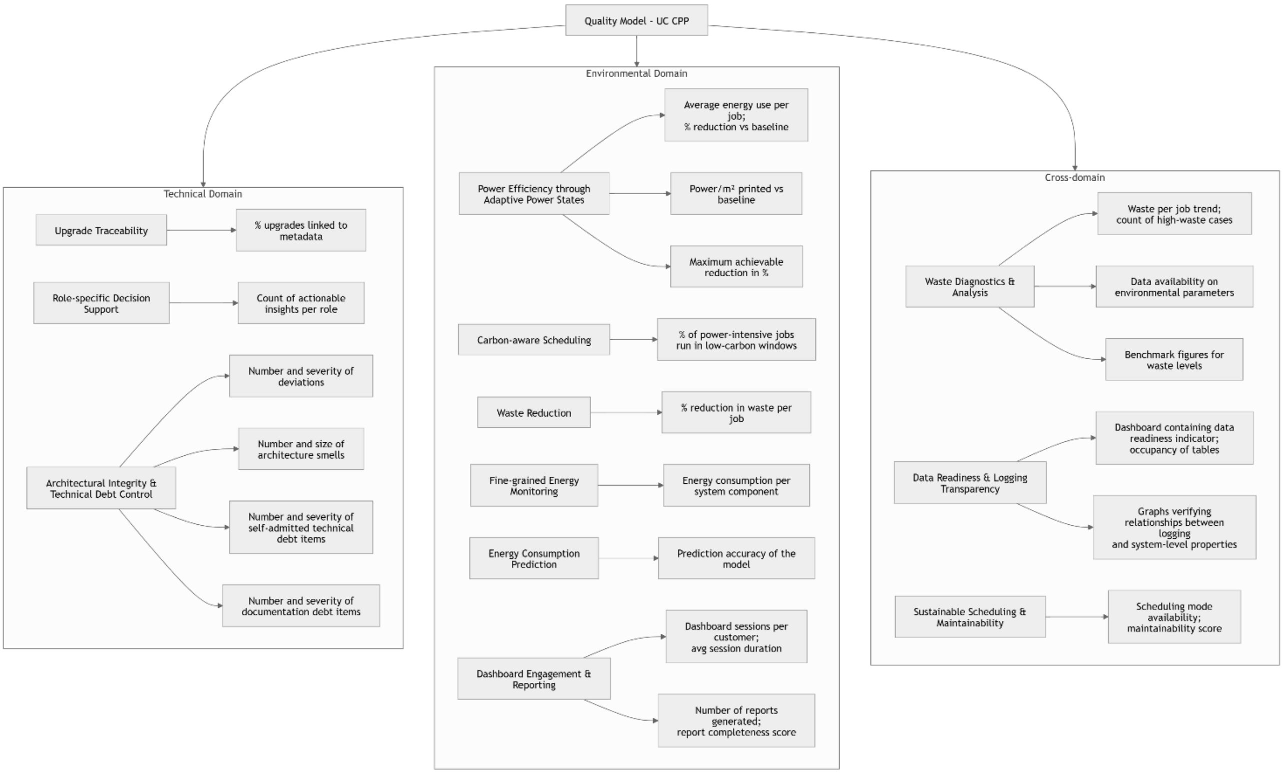


Figure 13: UC CPP – Consolidated Quality Model Diagram

Domain	**Sub-quality**	**Metric**
Technical	Sustainable Scheduling & Maintainability	Scheduling mode availability; maintainability score
Technical	Upgrade Traceability	% upgrades linked to metadata
Technical	Role-specific Decision Support	Count of actionable insights per role
Technical	Architectural Integrity & Technical Debt Control	Number and severity of deviations
Technical	Architectural Integrity & Technical Debt Control	Number and size of architecture smells
Technical	Architectural Integrity & Technical Debt Control	Number and severity of self-admitted technical debt items
Technical	Architectural Integrity & Technical Debt Control	Number and severity of documentation debt items
Environmental	Power Efficiency through Adaptive Power States	Average energy use per job; % reduction vs baseline

Environmental	Power Efficiency through Adaptive Power States	Power/square meter printed/(scope of engines) vs baseline
Environmental	Power Efficiency through Adaptive Power States	Maximum achievable reduction in %
Environmental	Carbon-aware Scheduling	% of power-intensive jobs in low-carbon windows
Environmental	Waste Reduction	% reduction in waste per job
Environmental	Fine-grained Energy Monitoring	Energy consumption per system component
Environmental	Energy Consumption Prediction	Prediction accuracy of the model
Environmental	Dashboard Engagement & Reporting	Dashboard sessions per customer; avg session duration
Environmental	Dashboard Engagement & Reporting	Number of reports generated; report completeness score
Cross-domain	Data Readiness & Logging Transparency	Dashboard containing data readiness indicator; occupancy of tables
Cross-domain	Data Readiness & Logging Transparency	Graphs that verify relationships between logging and system-level properties
Cross-domain	Sustainable Scheduling & Maintainability	Scheduling mode availability; maintainability score
Cross-domain	Waste Diagnostics & Analysis	Waste per job trend; count of high-waste cases
Cross-domain	Waste Diagnostics & Analysis	Data availability on environmental parameters
Cross-domain	Waste Diagnostics & Analysis	Benchmark figures for waste levels across the population

Figure 14: UC CPP – Consolidated Quality Model Table

4 Catalog of metrics and usage guidelines

The following section presents the full catalog of metrics for the defined use cases, structured according to the GQM approach. This allows establishing a clear link between the sustainability objectives defined for each use case, the guiding questions that frame their evaluation, and the corresponding measurable indicators.

For each use case, the metrics are organised in accordance with the sub-qualities defined in the quality model, ensuring consistency between the conceptual structure of the model and its practical implementation.

In addition, usage guidelines are provided for each metric to support consistent interpretation of results and to highlight how the measured values should be analyzed within the context of system behaviour and operational conditions.

4.1 Cleanwatts Use Case (UC – CLW)

4.1.1 Technical domain

Sub-quality: Data Reliability and Availability

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Ensure availability of smart metering data	Is smart metering data consistently accessible?	Data Availability – Percentage of uptime of FTP server	High values indicate a stable and reliable data acquisition pipeline. Decreases should be interpreted as disruptions in data flow that may impact downstream analytics.	Experiment 1 – Data availability
Ensure reliable data retrieval	How often does data retrieval fail?	Failed Retrieval Attempts – Number of failed FTP data requests	Increasing failures indicate instability in data access mechanisms. Frequent failures may affect system robustness, even if availability remains high.	Experiment 1 – Data availability
Ensure integrity of collected data	Is the retrieved data complete and uncorrupted?	Data Integrity – Number of corrupted or missing files	Higher values indicate risks to data trustworthiness and may compromise analytical outputs. Persistent issues should be treated as critical.	Experiment 1 – Data integrity

Sub-quality: System Performance and Efficiency

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Minimize data retrieval delays	How long does it take to retrieve data?	Retrieval Latency – Average transfer time per dataset	Higher latency values may indicate data transfer inefficiencies or scalability issues. Trends should be monitored rather than isolated values.	Experiment 1 – Retrieval latency
Ensure system responsiveness	How fast do dashboards respond to user interactions?	Dashboard Response Time – Time to load/update dashboard	Higher response times may reduce usability and limit the effectiveness of analytics.	NA

Sub-quality: AI and Forecasting Quality

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Improve accuracy of energy forecasting	How accurate are the AI forecasts compared to actual values?	Forecast Error – Difference between predicted and actual values (MAPE/RMSE)	Lower values indicate better predictive performance. Increasing errors may signal need for model recalibration.	Experiment 2 – Forecast accuracy

Sub-quality: Processing Efficiency of AI

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Ensure efficient execution of AI models	How long does the model take to generate forecasts?	Processing Time – Average runtime per forecast cycle	Higher processing times reduce responsiveness and may limit real-time capabilities.	Experiment 2 – Processing time

Sub-quality: Data Utilization and Effectiveness

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Maximize effective use of collected data	How much of the collected data is actually used?	Data Utilization – Percentage of	Lower utilization indicates gaps in data	Experiment 2 – Data utilization

		usable vs missing data	coverage or preprocessing issues.	

4.1.2 Environmental domain

Sub-quality: Energy Consumption

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Reduce environmental impact of AI processes	How much energy is consumed by AI algorithms?	AI Energy Consumption – Energy used during training and inference	Higher values indicate greater environmental cost. Should be interpreted relative to model benefits.	NA

Sub-quality: Carbon Emissions

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Minimize carbon emissions from system operation	What is the CO_2 impact of system computation?	Carbon Emissions – CO_2 emissions associated with computation	Higher emissions indicate lower sustainability performance. Should be analysed alongside energy consumption.	NA

Sub-quality: Renewable Energy Usage

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Increase use of locally generated renewable energy	How much locally generated energy is consumed within the community?	Self-consumption Rate – Percentage of local energy consumption	Higher values indicate better alignment with sustainability goals.	NA

4.1.3 Cross-domain

Sub-quality: Trade-off and Optimization

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping

Evaluate trade-offs between optimization objectives	How balanced are optimization results across cost, accuracy, and sustainability?	Trade-off Score – Composite indicator combining cost, accuracy, and energy impact	Should be interpreted comparatively to assess trade-offs across configurations.	NA

4.2 Wirtek Use Case (UC – WRT)

4.2.1 Technical domain

Sub-quality: Energy Measurement and Optimization

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Measure energy consumption per software module	How much energy does each software module or operation consume?	Module Energy Consumption – kWh consumed per operation/function execution on Wappsto platform	Provides a baseline for identifying high-impact modules. Increases over time typically indicate regressions, while comparisons across modes reveal optimisation potential.	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code
Quantify energy savings from green execution mode	What is the energy reduction when switching from performance mode to green mode?	Green Mode Energy Reduction – % energy reduction between performance and energy-efficient implementations	Values at or above target confirm effective optimisation. Lower values should be interpreted in context (e.g., already optimised modules or workload constraints).	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code

Sub-quality: Measurement Overhead Control

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Ensure telemetry measurement overhead remains acceptable	What is the resource overhead introduced by instrumentation?	Telemetry Overhead – % of total CPU/energy consumed by monitoring	Acts as a guardrail for all measurements. High values invalidate conclusions from other energy metrics and should trigger	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code

			optimisation of instrumentation.	

Sub-quality: Energy Quality Assurance in CI/CD

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Integrate energy regression detection into CI/CD pipeline	Do pull requests introducing energy regressions get flagged?	CI/CD Energy Regression Rate – % of PRs flagged before merge	Indicates process health rather than performance. Very low or very high values often reflect misconfigured thresholds rather than actual system behaviour.	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code

Sub-quality: Trade-off Dashboard Readiness

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Provide actionable dashboards for decision-making	Do dashboards expose sufficient KPIs for trade-offs?	Dashboard KPI Coverage – Count of sustainability KPIs in dashboard	Minimum KPI coverage is required for meaningful decision-making. Additional KPIs only add value if they are actionable.	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code

Sub-quality: Customer Adoption of Green Features

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Drive adoption of sustainability features	What percentage of customers enable green features?	Green Feature Adoption Rate – % of eligible customers using green mode or scheduling	Reflects real impact beyond technical implementation. Low adoption usually indicates usability or incentive issues rather than technical limitations.	WRT Pilot_Exp_2 – Goal 2: Smart Schedule of energy

Sub-quality: ESG Reporting Auditability

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Support ESG reporting with	Are emission reports complete and traceable?	ESG Report Completeness – % of months with	Measures reliability of reporting. Missing data points should be treated	No direct D4.1 mapping (D2.3 FR6 – not carried

auditable data		complete, auditable reports	as system issues rather than reporting gaps.	into pilot definition)

4.2.2 Environmental domain

Sub-quality: Carbon Impact and Optimization

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Track carbon footprint of software operations	What is the carbon footprint of each operation?	Operation Carbon Footprint – gCO$_2$eq per operation	Should be analysed in time series. Spikes often reflect grid intensity, while trends indicate software or execution inefficiencies.	WRT Pilot_Exp_1 – Goal 1: Green code vs Clean Code
Reduce emissions through scheduling strategies	How much CO$_2$ is avoided through carbon-aware scheduling?	Carbon Shift Savings – gCO$_2$eq avoided per scheduling cycle	High values indicate effective scheduling. Low or negative values reveal limitations in workload flexibility or forecasting accuracy.	WRT Pilot_Exp_2 – Goal 2: Smart Schedule of energy

4.2.3 Cross-domain

Sub-quality: Geographical Load Shifting

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Enable geographical load shifting to minimise carbon impact	How does moving workloads affect carbon emissions?	Load Shift Carbon Impact – % CO$_2$ reduction from geographical redistribution	Must be interpreted alongside constraints (latency, cost, data residency). High reductions are only valid if operational requirements are preserved.	WRT Pilot_Exp_2 – Goal 2: Smart Schedule of energy

4.3 Canon Use Case (UC – CPP)

4.3.1 Technical domain

Sub-quality: Upgrade Traceability

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Ensure traceability of sustainability upgrades	Are upgrades traceable via digital passport?	% upgrades linked to metadata	High traceability supports lifecycle management and reporting. Low values indicate gaps in tracking mechanisms and metadata consistency.	CPP Pilot Demo 1

Sub-quality: Role-specific Decision Support

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Provide role-specific actionable sustainability insights	Are dashboards tailored per user role?	Count of actionable insights per role	Higher values indicate effective personalisation and usability. Low values reduce users' ability to make informed sustainability decisions.	CPP Pilot Exp 5

Sub-quality: Architectural Integrity and Technical Debt Control

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Track deviations from architecture decisions	How does the current system implementation deviate from past architecture decisions and what are the consequences?	Number and severity of deviations	Higher values indicate increasing technical debt and potential risks to maintainability. Monitoring these deviations supports proactive architectural governance.	CPP Pilot Exp 4
Detect architecture and design smells	Does the software architecture suffer from dependency-based architecture smells, such as cyclic dependencies, hub-like dependencies, unstable dependencies and god components?	Number and size of architecture smells	Higher values indicate structural issues requiring refactoring, while lower values suggest a healthier architecture.	CPP Pilot Exp 4

Detect self-admitted technical debt items	What kind of problems with system maintainability/evolvability are self-admitted by members of the development team in natural language artifacts such as source code comments, issues, pull requests, commit messages?	Number and severity of self-admitted technical debt items	Higher numbers indicate awareness but also the presence of maintainability issues. Tracking trends supports prioritisation of improvements.	CPP Pilot Exp 5
Detect documentation debt items	What kind of problems are present in decision documentation? How complete are they? How standardised are they?	Number and severity of documentation debt items	Lower quality or completeness of documentation increases risks in knowledge transfer and long-term system evolution.	CPP Pilot Exp 5

4.3.2 Environmental domain

Sub-quality: Power Efficiency through Adaptive Power States

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Reduce energy consumption through adaptive power state transitions (dependent on learned/guessed customer behaviour)	How much energy is used per print job after adaptive power states?	Average energy use per job; % reduction vs baseline	This metric provides an estimate of the average energy reduction achieved relative to a defined baseline. It should be interpreted in the context of system constraints and realistic operational scenarios.	KPI Goal 1: Power Efficiency
	How much variation is typical in the field population?	Power/meter printed/(scope of engines) in % versus some baseline (typical, customer, …)	Variation across field systems should be interpreted as an indicator of differences in customer behaviour, configuration, or system conditions. High variability may reveal inconsistent optimisation performance, while low variability suggests more uniform and controlled	KPI Goal 1: Power Efficiency

			system behaviour. Benchmarking against typical or baseline systems supports identification of best practices.	
		Maximum achievable reduction in %	This metric provides an estimate of the maximum achievable energy reduction relative to a defined baseline. Higher achievable reductions indicate significant optimisation potential, while lower values suggest the system is already well-optimised. It should be interpreted in the context of system constraints and realistic operational scenarios.	KPI Goal 1: Power Efficiency

Sub-quality: Carbon-aware Scheduling

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Schedule power-intensive jobs in low-carbon periods	What % of power-intensive jobs run in low-carbon windows?	% of power-intensive jobs in low-carbon windows	Higher percentages indicate successful scheduling of energy-intensive jobs during low-carbon periods, contributing to reduced environmental impact. Lower values suggest limited use of scheduling optimisation or constraints in shifting workloads. Results should be interpreted in relation to workload flexibility and accuracy of carbon intensity forecasts.	KPI Goal 2: Scheduling Intelligence

Sub-quality: Waste Reduction

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Reduce material waste via smarter	How much waste per	% reduction	Lower waste levels indicate effective use of adaptive rejection strategies that balance quality and material efficiency. Higher	KPI Goal 5: Waste Reduction

rejection algorithms	job is reduced?	in waste per job	waste levels suggest overly conservative thresholds or suboptimal algorithms. Trends and comparisons across jobs can help identify opportunities to refine rejection strategies.	

Sub-quality: Fine-grained Energy Monitoring

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Establish fine-grained energy monitoring	How is energy consumption allocated to system components over time?	Energy consumption per system component.	This metric provides insight into how power consumption is distributed across individual components. Higher energy consumption in specific components may indicate inefficiencies or opportunities for optimisation. Balanced energy distribution indicates healthier energy allocation.	CPP Pilot Exp 2

Sub-quality: Energy Monitoring and Prediction

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Develop energy consumption prediction model	How well can modelled data substitute monitored energy consumption data?	Prediction accuracy of the model.	This metric reflects how closely the modelled power consumption aligns with actual monitored data. Higher accuracy indicates that the model is reliable for predictive analysis. Lower accuracy suggests the model requires improvement and that monitored data is still needed.	CPP Pilot Exp 5

Sub-quality: Dashboard Engagement and Reporting

Goal	**Question**	**Metric**	**Usage Guidelines**	**D4.1 Mapping**
Increase customer engagement with	How often are dashboards used?	Dashboard sessions per	Higher usage levels indicate strong customer engagement and perceived value of	KPI Goal 3: Dashboard Usage

sustainability dashboards		customer; avg session duration	sustainability insights. Low usage may suggest usability issues or limited relevance of the presented information.	
Enable sustainability reporting from dashboards	Can users generate complete sustainability reports?	Number of reports generated; report completeness score	Higher numbers and completeness of reports indicate effective support for sustainability reporting. Lower values suggest gaps in data availability, integration, or usability of reporting functionality.	KPI Goal 4: Reporting Capability

4.3.3 Cross-domain

Sub-quality: Sustainable Scheduling and Maintainability

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Balance sustainable scheduling with technical debt reduction	Do scheduling modes improve sustainability and maintainability?	Scheduling mode availability; maintainability score	Balanced improvements indicate successful integration of sustainability strategies without increasing technical debt. Trade-offs should be assessed carefully.	CPP Pilot Exp 4

Sub-quality: Data Readiness and Logging Transparency

Goal	Question	Metric	Usage Guidelines	D4.1 Mapping
Prepare functional logging for simulation and dashboards	What structured functional logging is available and accessible?	Dashboard containing data readiness indicator; occupancy of tables	Well-structured and complete logging enables reliable data analysis and dashboarding. Missing or inconsistent logging reduces the ability to generate insights and limits overall system transparency.	CPP Pilot Exp 1
	What data do we need for	Graphs that help to verify	Clear relationships between system logs and customer	CPP Pilot Exp 1

	determining customer behavioural patterns linked to system states?	relationships between logging and system-level properties	behaviour indicate strong analytical capability. Weak or unclear relationships suggest the need for improved data collection or modelling approaches.	

Sub-quality: Waste Diagnostics and Analysis

Goal	Question	Metric	Usage Guidelines
Provide actionable waste diagnostics	Can high-waste jobs/media be identified via dashboards?	Waste per job trend; count of high-waste cases	Increasing visibility of high-waste patterns supports targeted interventions. Lack of identifiable patterns may indicate insufficient data granularity.
	What is the influence of environmental parameters?	Data availability on environmental parameters	Provides contextual information for analysing waste behaviour. Strong correlations enable targeted optimisation; weak correlations suggest other dominant factors.
		Benchmark figures for waste levels across the population (customer, MIF)	Supports benchmarking across jobs and systems. Large deviations indicate inefficiencies or specific operating conditions requiring further analysis.

5 Conclusion

This deliverable presented the sustainability quality model and associated catalog of metrics developed within WP3 to support the systematic evaluation of sustainability across the project use cases. The work addressed the challenge of defining a common framework capable of integrating heterogeneous pilot requirements while maintaining consistency, interpretability, and practical applicability.

The proposed model combines traditional Software Engineering quality perspectives with sustainability-oriented evaluation criteria, enabling the assessment of both technical and environmental dimensions of digital systems. Through the adoption of the Goal–Question–Metric paradigm, the framework establishes traceability between project objectives, evaluation questions, and measurable indicators, supporting a structured and operational approach to sustainability assessment.

The resulting quality models demonstrate that sustainability in software-intensive systems cannot be evaluated exclusively through isolated environmental indicators or traditional software quality attributes alone. Instead, sustainability emerges from the interaction between system reliability, maintainability, operational efficiency, energy consumption, carbon impact, resource optimization, and decision-support capabilities. This is particularly evident in data-centric and AI-driven systems, where runtime behavior and computational efficiency directly influence sustainability outcomes.

The metric catalog developed in this deliverable provides a harmonized basis for monitoring and comparing sustainability performance across the project use cases. At the same time, the framework preserves sufficient flexibility to accommodate the specific operational characteristics and priorities of each pilot environment. The inclusion of cross-domain dimensions further highlights the importance of trade-off analysis and multi-objective optimization when evaluating sustainability-aware systems.

The outcomes of D3.1 establish the foundation for subsequent WP3 activities related to sustainability monitoring, benchmarking, optimization, visualization, and decision support. Future work will focus on the operationalization of the proposed metrics within pilot environments, the collection and analysis of measurement data, and the exploration of optimization strategies capable of improving sustainability performance while preserving system quality and operational effectiveness.

Bibliography

[1] Basili, V. R., & Weiss, D. M. (1984). *A Methodology for Collecting Valid Software Engineering Data*. IEEE Transactions on Software Engineering, SE-10(6), 728–738.

[2] Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). *The Goal Question Metric Approach*. Encyclopedia of Software Engineering.

[3] Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). *The GREENSOFT Model: A Reference Model for Green and Sustainable Software and its Engineering*. Sustainable Computing: Informatics and Systems, 1(4), 294–304.