

A Validated Visual Analytics Prototype for Monitoring and Alerting of Gantry 2 and OPTIS 2 Parameters at PSI CPT

Christoph Elmiger
FHNW — School of Computer Science
Program: Data Science and AI
Project 6 (P6) Bachelor's Thesis Report

7 August 2026

Abstract

Operational data from Gantry 2 and the ocular-treatment system OPTIS 2 at the Paul Scherrer Institute Center for Proton Therapy are distributed across heterogeneous historical exports. This work extends an existing prototype into a traceable, read-only monitoring system and examines three linked questions: reproducible processing of those exports, task-oriented visual analysis, and practical usefulness for representative monitoring work.

An applied, artefact-centred case study was used. Versioned source contracts, staged processing, validation before publication, and recorded provenance were used to prepare historical data from both treatment systems for monitoring. Ten dashboard products covering data from 2014 to March 2026 were produced within the reviewed source scope. The dashboard was reorganized around recurring monitoring tasks, including long-term trend review, comparison of related parameters and operating contexts, inspection of warnings and safety-interlock records, and follow-up of unusual observations. An exploratory anomaly-oriented workflow combines robust statistical indicators with Isolation Forest scoring to produce a review queue for expert inspection rather than a validated alarm.

All three prospective local users identified for the project took part in a formative evaluation of dashboard version 0.3.0. Task ratings were generally positive and the mean System Usability Scale score was 86.7. These findings informed the redesign, but they do not establish the usability of the final interface because no release-equivalent follow-up evaluation was completed. The fixed release was also built and exercised on an AlmaLinux test virtual machine as a RHEL-compatible proxy for the intended production environment. The evidence supports reproducible processing within the reviewed source scope, technical realization of the intended visual-analysis workflows, and technical readiness for handover in the tested environment. Final PSI deployment and operational acceptance, user evaluation of the final interface, and domain validation of anomaly candidates remain open.

Keywords: proton therapy; monitoring dashboard; data pipeline; reproducibility; visual analytics; provenance; usability evaluation.

Acknowledgements

I am grateful to Zema Chowdhuri and Santiago Perez Carmona at the PSI Center for Proton Therapy for explaining the operational context, clarifying the source data and infrastructure requirements, and providing feedback throughout the project. I also thank my academic supervisors, Prof. Dr. Arzu Çöltekin and Cédric Merz, at the FHNW School of Computer Science for their guidance on the research framing, evaluation, and report. Finally, I thank my wife, Viktoria, for her patience and support throughout the project, and our dog, Minnie, for providing a reliable reason to step away from the screen occasionally.

Contents

List of Figures	v
List of Tables	v
List of Abbreviations	vii
Glossary	viii
1 Introduction	1
1.1 Context and Motivation	1
1.2 Problem Statement	1
1.3 Aim and Scope	2
1.4 Research Questions	2
1.5 Research Approach and Evidence	2
1.6 Main Contributions	3
1.7 Structure of the Thesis	3
2 Related Work	3
2.1 Operational Monitoring in Proton Therapy	3
2.2 RQ1: Reproducible Data Processing and Provenance	4
2.3 RQ2: Visual Analytics and Interaction Design	5
2.4 RQ3: Usability and Practical Usefulness	6
2.5 Exploratory Robust and Anomaly-Oriented Methods	7
2.6 Synthesis and Implications for the Research Questions	8
3 Methods	8
3.1 Research Design	8
3.2 Case Context, Scope, and Data Governance	8
3.3 Baseline Assessment and Revision Process	9
3.4 Data Basis and Source Contracts	9
3.5 RQ1: Pipeline Extension and Validation	9
3.6 RQ2: Dashboard Revision Method	13
3.7 Exploratory Anomaly-Oriented Method	14
3.8 RQ3: Formative Evaluation	14
3.9 Analysis and Reporting Procedure	15

4	Results	15
4.1	Overall Evidence Status	15
4.2	RQ1: Data Pipeline and Data Basis	17
4.3	RQ2: Dashboard and Interaction Revision	20
4.4	RQ3: Formative Evaluation of Practical Usefulness	20
4.5	Exploratory Anomaly-Oriented Results	21
5	Discussion	22
5.1	Overall Interpretation	22
5.2	RQ1: Data Pipeline and Data Basis	22
5.3	RQ2: Visual Analytics and Interaction Design	23
5.4	RQ3: Evaluation of Practical Usefulness	23
5.5	Exploratory Anomaly-Oriented Extension	23
5.6	Threats to Validity	24
5.7	Contribution of the Thesis	25
5.8	Practical Implications and Handover	25
6	Conclusion	26
	References	27
	Declaration of Independent Authorship	30
	Use of AI-Assisted Tools	31
A	Interface Evidence	32
B	Supplementary Material and Evidence	36

List of Figures

3.1	Logical architecture and responsibility boundaries of the monitoring artefact. The diagram separates synchronized input and source interpretation from staged processing, exploratory anomaly scoring, dashboard consumption, and cross-cutting metadata and lineage recording.	11
3.2	Stage-local reuse, rebuild, validation, and publication logic, illustrated for Bronze. A matching reusable published artefact is reused; otherwise a candidate is built and may become current only after the publication criteria are satisfied. Blocking findings quarantine the candidate without replacing the published state. Silver and Gold apply the same mechanism with their respective upstream artefacts and transformation definitions.	12
4.1	Data-quality page showing the latest published run, prepared products, validation findings, and execution history.	18
4.2	Revised overview with data-readiness status and principal review paths.	20
A.1	Monitoring overview with data-readiness status, the shared observation window, and the three principal review paths.	33
A.2	Gantry 2 BTVS page with a daily trend, selected parameter, and energy-band comparison.	33
A.3	Gantry 2 anomaly-candidate page with candidate volume, method classes, and grouped review periods.	34
A.4	Hover information in the Gantry 2 BTVS view, exposing the observation date, energy level, and the displayed shift relative to the reference baseline.	34
A.5	Warnings and interlocks page with source-system records, filters, and available operational context.	35
A.6	Data-quality and coverage page showing the latest published pipeline execution, prepared products, validation findings, and execution history.	35

List of Tables

3.1	Operationalization of the research questions.	9
3.2	Integrated source families and intended analytical use.	10
3.3	Task-to-view mapping used for the dashboard revision.	13
4.1	Evidence status by research question and additional project scope.	15
4.2	Planned scope and achieved state at hand-in.	16
4.3	Implemented source scope by treatment system.	17
4.4	Summary of the published dashboard-product state.	17
4.5	Production-like verification of the final release.	19
4.6	Task-level ratings for dashboard version 0.3.0, reported as median [range].	21
4.7	System Usability Scale results for dashboard version 0.3.0.	21
4.8	Trace from formative baseline evidence to the final dashboard response.	21

4.9	Exploratory candidate counts in the recorded dashboard review window.	22
B.1	Supplementary artefacts included with the hand-in.	36
B.2	Technical claim-to-evidence index for the sanitized archive.	37

List of Abbreviations

Abbreviations are expanded at first use where practical; this list provides a compact reference for recurring project, domain, and technical terms.

BTVS	Beam Tune Verification System
CI	Continuous Integration
CPT	Center for Proton Therapy
CSV	Comma-Separated Values
FHNW	University of Applied Sciences and Arts Northwestern Switzerland
ILK	Interlock source-family identifier used in the PSI exports
ML	Machine Learning
NBVS	OPTIS 2 nozzle beam-verification source-family identifier
NPVS	OPTIS 2 nozzle position-verification source-family identifier
OPTIS	PSI ocular proton-therapy treatment facility
P5	Project 5, the student project preceding this bachelor's-thesis project
P6	Project 6, the current bachelor's-thesis project
PSI	Paul Scherrer Institute
RHEL	Red Hat Enterprise Linux
RQ	Research Question
SUS	System Usability Scale
VM	Virtual Machine

Glossary

The glossary defines proton-therapy and project-specific data/deployment terms used in the report. General visual-analytics terminology is introduced in Chapter 2 and is not repeated here.

Domain and monitoring terms

Anomaly candidate. An observation flagged as statistically unusual and placed in a review queue. A candidate is not, by itself, evidence of a fault or an alarm condition.

Beam tune / tune energy. A configured set of beam-line settings associated with a proton-beam energy. Changing proton energy changes penetration depth, so comparisons at the same or different tune energies can provide important operating context.

Energy band. A grouping of observations by beam/tune energy for comparison. The dashboard uses energy-band views to compare how a parameter behaves at different operating energies.

Fraction. One treatment session within a course of radiotherapy. Patient and fraction identifiers appear in some source data because they help relate measurements to the treatment context in which they were recorded.

Gantry. In this report, Gantry 2 is PSI’s scanned-beam treatment station for deep-seated tumours. Its recorded machine orientation is available as gantry-angle context in some data.

Gantry angle. The recorded mechanical rotation angle of Gantry 2. The project uses it as contextual information when measurements may depend on treatment geometry.

Interlock. A machine-protection or safety condition that can inhibit or interrupt operation when a monitored condition violates defined limits. Interlock records are treated as source-system evidence, not as dashboard-derived anomaly classifications.

Nozzle. The final beam-line assembly close to the patient, containing components used to shape, monitor, or otherwise prepare the treatment beam. OPTIS 2 nozzle records therefore provide system-specific operating context.

Paired channels. Two corresponding measurement channels that are compared with each other, for example left/right channels. Their difference can reveal relative drift even when both channels change together.

Quality assurance. Planned checks used to verify that treatment equipment and its relevant measurements remain suitable for safe operation. The dashboard may support review of such information but does not replace formal quality-assurance procedures.

Treatment field. A planned beam-delivery setup within a treatment. In this report, “field” in a clinical context refers to the treatment field, not a database column.

Data and deployment terms

Bronze, Silver, Gold. Names for the three processing stages used in the project. Bronze preserves source-oriented data and ingestion history; Silver standardizes supported variants into a common structure; Gold prepares joins and summaries used by monitoring views.

Container image. A packaged software runtime containing the application and its required dependencies. Using a fixed container image helps reproduce the same software environment on another compatible host.

Data artefact. A stored output produced by a processing step, such as a Bronze, Silver, or Gold data file. Artefacts are recorded so that the system can distinguish newly built, reused, rejected, and previously published versions.

Data lineage / provenance. The recorded relationship between a displayed or published data product and the source files, processing steps, configuration, and validation decisions from which it was derived.

Operational acceptance. Confirmation on the final target infrastructure that the system works with the required storage, security, synchronization, service-management, and ownership arrangements. It is stronger than a successful technical test in a similar environment.

Production-like environment. A test environment that deliberately reproduces important characteristics of the intended production setup, but is not the final operational system. In this project, AlmaLinux was used as a RHEL-compatible proxy for the intended PSI host.

Publication. In the data pipeline, making a validated product the current version available to downstream consumers such as the dashboard. This meaning is unrelated to academic publishing.

Release-equivalent evaluation. A user evaluation performed with the same software state, or an explicitly equivalent state, as the version intended for handover. Results from an earlier prototype can inform redesign but cannot validate the final release.

Service. A long-running or controlled software process managed by the operating system. In the deployment model, batch data processing and the continuously available dashboard run as separate services.

Source contract. Versioned rules that state how a source file is recognized and interpreted, including expected filenames, layouts, fields, types, known corrections, and validation requirements.

Source family / source variant. A source family is a recurring category of operational export with a common purpose. A source variant is a historically different file layout or representation within that family that requires explicit handling.

1 Introduction

1.1 Context and Motivation

The Center for Proton Therapy (CPT) at the Paul Scherrer Institute (PSI) operates treatment systems with different technical and clinical roles. At PSI, accelerated protons are guided through beam lines to treatment stations, where the beam is shaped, steered, monitored, and delivered. Gantry 2 is used for deep-seated tumours, whereas OPTIS 2 is dedicated to ocular treatment and uses a fixed horizontal beam line (Paul Scherrer Institute, Center for Proton Therapy, n.d. b, n.d. c). Both systems produce operational measurements, warnings, safety-interlock records, and evaluation data. Reviewing these records over time can support routine monitoring and the investigation of changes that require expert follow-up (Elmiger, 2026b).

Before the preceding project, much of this work relied on comma-separated values (CSV) exports that were plotted manually for individual questions. The preceding Project 5 (P5) student project introduced a Python-based data pipeline—a repeatable sequence for reading, transforming, validating, and preparing those exports—and an interactive dashboard for Gantry 2. It demonstrated reproducible processing of the available 2025 data and established the technical baseline for the present project (Elmiger, 2026a).

Important gaps remained after P5. Historical data had not been integrated across the full available period, OPTIS 2 was not included, the dashboard still reflected an early prototype, and the planned user evaluation had not been completed. The system also had to fit an on-premises operating model, meaning that it would run on PSI-managed infrastructure rather than a public cloud. In that model, synchronized source files are processed as a controlled batch while the dashboard reads validated, published products without modifying source systems.

This project extends the prototype as a single monitoring artefact: the pipeline prepares the data used by the dashboard, and the dashboard supports the monitoring tasks evaluated in the study. Technical implementation alone cannot establish usability or operational acceptance.

1.2 Problem Statement

The data basis is heterogeneous. File names, delimiters, headers, columns, parameter sets, timestamps, and record structures vary between systems and historical periods, and some files contain known defects or mixed layouts. Processing therefore has to recognize supported variants explicitly, reject unsupported structures conservatively, and publish only validated outputs.

The prepared products must also retain the context needed for monitoring. Depending on the task, this includes trends, beam-energy settings, paired measurement channels, gantry angles (the machine rotation angle), patients, treatment fractions (individual treatment sessions), warnings, and interlocks. The dashboard must expose these relationships through views and interactions that match representative monitoring tasks rather than simply presenting a collection of charts.

The project agreement also calls for task-based assessment of usability, interpretability, and practical usefulness (Elmiger, 2026b). A formative evaluation of dashboard version 0.3.0 informed the redesign; no final-version user study was completed. The anomaly-oriented analysis is likewise limited in scope: without a complete labelled set of known faults, it can produce traceable review candidates but cannot validate an automatic alarm system.

1.3 Aim and Scope

The aim is to extend and critically assess the prototype as a traceable, read-only monitoring and visual-analytics system for operational data from Gantry 2 and OPTIS 2. The core scope comprises:

1. integration of historical Gantry 2 and agreed OPTIS 2 source variants;
2. reproducible transformation, validation, publication, and lineage;
3. preparation of monitoring-oriented data products;
4. revision of selected dashboard views and interactions;
5. baseline and, where feasible, follow-up task-based evaluation of usability and practical usefulness; and
6. production-like technical validation of the deployment model.

The formative baseline evaluation was completed. A release-equivalent user study and the final on-premises deployment could not be scheduled before the early-August hand-in because prospective users and infrastructure staff were unavailable during parts of the final project month (Elmiger, 2026b).

The exploratory component uses robust statistical indicators and Isolation Forest scoring to prioritize observations for expert review. It does not modify treatment or control systems, replace formal quality-assurance procedures, make clinical decisions, or generate operational alarms. Technical deployment was exercised on an AlmaLinux test virtual machine (VM) as a proxy for the target Red Hat Enterprise Linux (RHEL) environment; final PSI deployment remained outside the completed work.

1.4 Research Questions

The project is structured around three research questions (RQs) defined in the project agreement:

- RQ1 — Data pipeline and data basis:** How can the existing traceable data pipeline be extended so that operational data from Gantry 2 and OPTIS 2 can be processed robustly, reproducibly, and in a form suitable for visual monitoring, including previously unseen data variants?
- RQ2 — Visual analytics and interaction design:** How should selected dashboard views and interactions be revised so that domain users can identify and interpret trends, deviations, and unusual patterns in operational parameters more clearly and consistently during representative monitoring tasks?
- RQ3 — Evaluation of practical usefulness:** To what extent does the revised prototype support representative monitoring and analysis tasks in the PSI/CPT context, as evaluated through task-based assessment of usability, interpretability, and practical usefulness?

The questions are sequential: RQ1 establishes the data basis used in RQ2, while RQ3 evaluates the interaction concept. Because the final interface was not included in the user study, the available evidence is stronger for the technical work than for final-interface usability.

1.5 Research Approach and Evidence

An applied, artefact-centred research approach is used. Data-pipeline claims are assessed through source coverage, executed transformations, validation, publication behaviour, lineage, and reruns; dashboard claims through task abstraction, design rationale, and implemented interaction paths; and usability through the formative evaluation of version 0.3.0, including task ratings, qualitative feedback, and the System Usability Scale.

The submitted source repository and sanitized technical evidence package accompany the report as supplementary material (Appendix B) and preserve detailed configuration, release, test, and execution records. The anomaly-oriented component is assessed separately as a feasibility workflow; fault-detection performance is not evaluated.

1.6 Main Contributions

The thesis applies established data-engineering, provenance, visual-analytics, and usability methods to the historical Gantry 2 and OPTIS 2 monitoring case. It does not introduce a new algorithm or visual encoding. The main contributions are:

- **Data engineering:** source contracts for the reviewed historical variants, staged transformation into monitoring-ready products, validation before publication, and recorded lineage and reuse;
- **Visual analytics:** a task-oriented dashboard that connects data readiness, trends, comparisons, recorded events, anomaly candidates, and contextual follow-up while retaining system-specific context;
- **Evaluation and reporting:** explicit separation between what was technically verified, what was observed in the formative user study, and what still requires operational validation; and
- **Handover:** a fixed release, production-like deployment material, and a sanitized evidence package for technical inspection and continuation of the work.

Separately, the anomaly workflow demonstrates traceable candidate generation and contextual review without treating statistical unusualness as a validated alarm.

1.7 Structure of the Thesis

Chapter 2 reviews the relevant literature, Chapter 3 describes the study and implementation methods, Chapter 4 reports the findings, and Chapters 5–6 interpret and summarize them.

2 Related Work

The review covers the literature needed for the project: proton-therapy monitoring, reproducible data processing and provenance, visual-analytics design, usability evaluation, and exploratory anomaly analysis. These areas provide the basis for the design and evaluation choices described in Chapter 3.

2.1 Operational Monitoring in Proton Therapy

Proton therapy is a form of external-beam radiotherapy in which accelerated protons are transported to a treatment station. The penetration depth of the beam depends on its energy, so beam-energy settings and the hardware used to transport, shape, and monitor the beam are directly relevant to treatment delivery. At PSI, the accelerator feeds several treatment stations through dedicated beam lines. Gantry 2 is used for deep-seated tumours and scans the beam across the target, while OPTIS 2 is dedicated to ocular tumours and uses a fixed horizontal beam line (Paul Scherrer Institute, Center for Proton Therapy, n.d. a, n.d. b, n.d. c). The monitoring views therefore retain the system-specific context implied by these different treatment configurations.

For a non-clinical reader, the operational chain is more important here than the full treatment physics: a beam tune sets the beam energy and associated beam-line configuration; treatment

hardware delivers and monitors the beam; and the control and safety systems record measurements, warnings, and interlocks. Some exports also retain patient identifiers, treatment-field context, and fraction identifiers (one fraction is one treatment session within a radiotherapy course) so that technical measurements can be interpreted in the context in which they were acquired.

Proton-therapy delivery depends on coordinated behaviour across several technical systems. Machine quality assurance includes recurring measurements of beam position, range, output, and other system-specific quantities. Procedures vary between facilities and treatment systems, but the common aim is to identify relevant deviations before they affect treatment delivery or system availability (Actis et al., 2017; Arjomandy et al., 2019; Ding et al., 2021).

Quality-assurance decisions are not based on one data source or one threshold. Comprehensive proton-machine quality assurance uses different checks for different machine characteristics and failure modes (Arjomandy et al., 2019). Risk-based radiotherapy quality-assurance work likewise shows that priorities depend on the failure mode and its consequences; Taylor et al. (2023) demonstrate this specifically for clinical-trial quality assurance using failure modes and effects analysis. The PSI data used here likewise combine measurements, warnings, interlocks, and evaluation records from two systems. These records can support review, but their meaning still requires domain interpretation.

Daily checks and long-term trend review answer related but different questions. A value may remain within an operational tolerance while its centre, variability, or frequency changes over a longer period. Statistical process-control literature treats observations as part of a process over time rather than as independent pass/fail events (Montgomery, 2019; Xiao & Li, 2024). For the dashboard, the practical implication is to support temporal comparison and investigation without presenting descriptive references as clinical limits.

Warnings and interlocks require similar restraint. An interlock is a machine-protection or safety condition that can inhibit or interrupt operation when defined limits are violated. Warning and interlock records are direct source-system evidence, but their presence alone does not establish the cause or clinical importance of an event. The prototype keeps these records available for expert follow-up and separate from inferred anomaly candidates, without introducing a new alarm classification.

For this project, monitoring therefore has to preserve system and operating context, treat long-term trends as a complement to formal quality-assurance procedures, and leave interpretation and action to qualified PSI staff.

2.2 RQ1: Reproducible Data Processing and Provenance

For heterogeneous operational data, reproducibility requires more than retaining the source code. The relevant inputs, configuration, processing steps, software state, and outputs must be identifiable so that a result can be reproduced and interpreted (Freire & Chirigati, 2018; Kanterakis et al., 2018; Leipzig et al., 2021).

Provenance describes the relationships between data products and the entities or activities from which they were derived. The World Wide Web Consortium (W3C) PROV model distinguishes entities, activities, and agents and provides a general vocabulary for representing derivation and responsibility (Moreau & Missier, 2013). A project-specific pipeline can follow the same principle without implementing the full standard, provided that it records enough information to identify the source files, transformations, and configuration behind a published result.

Metadata—records about inputs, processing state, and outputs—are particularly important for incremental processing. New files are added, existing files may change, and only part of a dataset

may need rebuilding. A final file path alone does not describe such a run: the result also depends on the observed input state, transformation definition, validation outcomes, and publication decision. Provenance-driven workflow approaches likewise emphasize relationships among inputs, processing, and outputs, with the latest output alone being an incomplete record (Leipzig et al., 2021; Mitchell et al., 2022).

Data-quality assessment is context-dependent. Batini et al. (2009) distinguish multiple data-quality dimensions and methodological phases for assessment and improvement. For this project, that motivates source-specific validation rules and severity levels tied to the consequences of defects for downstream use and publication safety.

These concepts lead to four requirements for the pipeline.

1. **Documented source interpretation.** Historical files should be processed under stated assumptions about their names, layouts, columns, types, and known irregularities. Unsupported structures should be rejected for review instead of guessed silently.
2. **Separation of processing responsibilities.** Raw ingestion, canonical transformation, and task-specific analytical preparation should remain distinguishable. This reduces the risk of mixing source repair, semantic normalization, and dashboard aggregation in one opaque step.
3. **Publication after validation.** A newly generated file should replace the current output only after the relevant validation checks. Validation outcomes must be recorded, and a publication-blocking failure should leave the previous valid state available.
4. **Auditable reruns.** Repeated execution should distinguish rebuilt, reused, rejected, and unchanged products. The metadata should preserve the relationship between source observations, derived artefacts, and published state.

The Bronze–Silver–Gold structure used in the project is one implementation of the second requirement. Bronze retains source-oriented structure and ingestion provenance. Silver converts supported historical variants into canonical (standardized) datasets, applies validation, and removes defined duplicates. Gold combines related records across prepared data and produces summaries and other products required by monitoring tasks. The terminology is implementation-specific; the relevant research principle is the separation of source preservation, semantic standardization, and analytical use.

Configuration-driven source contracts move assumptions about filenames, layouts, aliases, types, required values, and known corrections out of conditional file-reading code and into reviewable definitions. The supported scope is explicit, and an unknown structure requires analysis, a documented contract, and corresponding tests before it is accepted.

The same reproducibility argument extends to operational delivery. Configuration, container images (packaged software runtime environments), filesystem paths used at run time, and operating-system service definitions influence whether the same processing behaviour can be obtained outside the development environment. The deployment model is therefore included in the implementation evidence for RQ1. Validation on the final PSI environment is considered separately.

2.3 RQ2: Visual Analytics and Interaction Design

Visual analytics combines computational data analysis with interactive visual representations that support human reasoning (Cui, 2019). In this project, the pipeline and dashboard are coupled because the available visual tasks depend on the analytical products prepared upstream.

Shneiderman’s overview-first principle provides a basic interaction sequence: present an overview, allow filtering or zooming, and provide details on demand (Shneiderman, 1996). This pattern is suitable for operational monitoring because an investigation often begins with a broad time range or summary and then moves toward a parameter, period, patient, fraction, warning, or source record. It does not prescribe a specific chart type, but it provides a structure for connecting views.

Munzner’s nested model separates four levels of visualization design: the domain problem, the abstraction of data and tasks, the choice of visual encoding and interaction, and the algorithmic implementation (Munzner, 2009). Errors at an outer level cannot be solved only by improving an inner level. A technically correct chart may still fail if it addresses the wrong task, and an appropriate design may still fail if the underlying data product is unsuitable. This separation is directly relevant to the thesis:

- the domain level concerns the monitoring questions of PSI/CPT staff;
- the abstraction level concerns tasks such as reviewing trends, comparing conditions, and investigating unusual observations;
- the encoding and interaction level concerns line charts, heatmaps, tables, filters, and drill-down;
- the implementation level concerns the interactive web dashboard, implemented with Dash and Plotly, and its data-access logic.

Conventional line charts, scatter plots, heatmaps, summary values, and tables are used because they are familiar and sufficient for the identified tasks. The design problem is how to select, arrange, and connect them. Dashboard-design literature emphasizes repeated patterns, consistency, and a workable balance between overview and detail (Bach et al., 2023); interaction taxonomies distinguish intents such as selecting, exploring, filtering, reconfiguring, and connecting data (Yi et al., 2007). Controls are included when they advance a monitoring task or preserve relevant context.

Applied to the revised dashboard, these principles favour pages organized around monitoring questions and treatment systems rather than implementation modules. Repeated controls should carry the same meaning across pages, while task-specific context remains accessible. Each main view should also offer a useful next step, such as moving from a daily overview to energy detail, gantry-angle context, patient or fraction records, or contributing source records.

2.4 RQ3: Usability and Practical Usefulness

Under ISO 9241-11, usability is defined for specified users, goals, and a context of use and includes effectiveness, efficiency, and satisfaction; it is broader than the appearance of an interface (International Organization for Standardization, 2018). Evaluation should therefore use tasks that resemble the intended monitoring work.

Task-based evaluation is particularly relevant for multidimensional visualizations because different tasks expose different strengths and weaknesses. Evaluation should examine what users are asked to identify, compare, or interpret instead of relying on a single general judgement (Valiati et al., 2006). Dashboard usability studies use a range of evaluation methods and tools (Almasi et al., 2023). Observing representative users can reveal concrete terminology, context, navigation, and interpretation problems that are not evident from implementation alone (Richter Lagha et al., 2020).

Task-level questionnaires can complement observation by capturing the perceived ease of a just-completed task. The seven-point post-task ease rating evaluated by Sauro and Dumas (2009),

commonly referred to as the Single Ease Question (SEQ), provides a standardized one-item measure of perceived task ease.

The System Usability Scale (SUS) is a short standardized questionnaire that summarizes perceived usability on a scale from 0 to 100 (Brooke, 1996). Bangor et al. (2008) provide adjective and acceptability interpretations for SUS scores. In this study, those global interpretations are treated as descriptive complements to, rather than substitutes for, task-level evidence. SUS cannot show whether a proton-therapy parameter was interpreted correctly, an unusual pattern was detected, or a monitoring decision was operationally appropriate. Task observations and qualitative feedback are needed alongside it.

Small-sample usability studies can be useful for formative diagnosis, but sample size limits the precision and generalizability of population-level estimates (Sauro & Lewis, 2016). Individual scores and comments can nevertheless identify concrete interface problems and guide revision. The evaluation in this project was used to characterize the baseline interface and inform redesign; statistical validation across PSI/CPT users was outside its scope.

The evaluated software version also matters. Findings from version 0.3.0 can motivate later changes, but they cannot establish the usability of the final implementation.

2.5 Exploratory Robust and Anomaly-Oriented Methods

The anomaly-oriented component is exploratory and examines whether the prepared data can support a traceable review queue when complete fault labels are unavailable. In anomaly-detection literature, an anomaly is an observation or pattern that differs from an expected model of normal behaviour; whether it represents an error, fault, or meaningful event depends on application context (Chandola et al., 2009).

Operational measurements may contain outliers, gaps, varying observation counts, and changes in operating regime. Means and standard deviations can be strongly affected by extreme values, whereas medians and robust scale estimates reduce the influence of individual observations (Huber & Ronchetti, 2009). Robust descriptive statistics are useful for expressing distance from recent history, but they remain descriptive indicators rather than clinically defined tolerances.

Statistical process-control methods offer another way to review changes in location, variability, and process behaviour over time (Montgomery, 2019; Xiao & Li, 2024). Their use in radiation-therapy quality assurance also exposes an important constraint: thresholds and chart rules depend on process stability, data quality, and the cost of false signals. Robust scores are used here as review cues, not automatic pass/fail decisions.

Isolation Forest is an unsupervised anomaly-detection method based on random partitioning. Observations that can be isolated with shorter paths receive higher anomaly scores (Liu et al., 2012). The method is computationally practical and does not require labelled faults, which makes it suitable for a feasibility study. However, a high score only indicates that an observation is unusual relative to the fitted data. It does not establish whether the cause is a technical fault, a regime change, a data-quality problem, or valid but rare operation.

The prototype combines robust statistical indicators with Isolation Forest scores. Agreement and disagreement between the methods are retained as separate candidate classes, so a reviewer can see whether a candidate is driven by a transparent statistical deviation, an unsupervised model, or both. The combination is used to structure the review queue and has not been validated as a classifier.

Temporal ordering is an additional requirement for retrospective monitoring. A historical observation should be scored only from information that would have been available at that time.

Long gaps and confirmed interventions may also separate operating regimes. The implementation therefore uses preceding observations, defined segments, transition periods, and insufficient-history states. These choices reduce look-ahead and regime-mixing risks. Labelled outcomes were unavailable for validation.

2.6 Synthesis and Implications for the Research Questions

The literature sets the main constraints used in this thesis. Reproducible processing requires explicit source interpretation, separated processing stages, validation before publication, identifiable configuration, and provenance. Dashboard design should start from domain tasks and preserve the context needed to move from overview to detail. Usability claims require representative tasks and must remain tied to the software version that was evaluated.

For the exploratory anomaly workflow, robust statistics and Isolation Forest provide complementary review signals rather than validated alarms. Model provenance and context are therefore retained for expert review. The research contribution lies in integrating these established methods for the historical Gantry 2 and OPTIS 2 data and the monitoring work at PSI/CPT, rather than in proposing a new algorithm or visualization technique.

3 Methods

This chapter describes how the prototype was extended and evaluated. It defines the case context and source scope, the processing and validation approach, the dashboard revision method, the exploratory anomaly workflow, and the formative user evaluation. The methods are organized around the evidence needed to address the three research questions while keeping technical verification, user evaluation, and operational acceptance distinct.

3.1 Research Design

An applied, artefact-centred case-study design was used. Design-science reasoning fits the project because a purposeful information-technology artefact was constructed to address an organizational problem, with its utility and quality assessed alongside the implementation (Hevner et al., 2004). The case-study element bounds the inquiry to Gantry 2 and OPTIS 2 at PSI/CPT and makes the local data, users, and infrastructure constraints part of the interpretation (Runeson & Höst, 2009).

The unit of analysis is the extended monitoring artefact in the PSI/CPT case. Work proceeded from the P5 baseline through baseline assessment, data-pipeline extension, dashboard revision, and evaluation of the available technical and user evidence. No causal before-and-after comparison of the dashboard was possible.

Table 3.1 summarizes the operationalization of the research questions.

3.2 Case Context, Scope, and Data Governance

The intended users are PSI/CPT staff in technical, engineering, and medical-physics roles. Representative activities include reviewing trends, comparing related operating contexts, checking warnings and interlocks, and investigating unusual observations (Elmiger, 2026b).

The artefact is read-only and processes project exports without writing to treatment, control, or source systems. Source-system warnings and interlocks are kept distinct from statistically inferred anomaly candidates. Clinical acceptability remains outside the dashboard's role.

Table 3.1: Operationalization of the research questions.

Question	Main construct	Evidence used	Main limitation
RQ1	Robust, reproducible, monitoring-ready processing	Supported source variants, published products, validation and lineage, controlled publication, rerun behaviour, tests, and production-like execution	Limited to reviewed variants and a proxy deployment environment
RQ2	Clear and consistent support for monitoring tasks	Task-to-view mapping, literature-based design rationale, implemented pages, and interaction paths	Final interface not evaluated with users
RQ3	Usability, interpretability, and practical usefulness	Task-based formative evaluation, task ratings, SUS, and qualitative comments	Evaluation covered version 0.3.0; final implementation not evaluated

Some source data contain coded patient and fraction reference fields because these provide acquisition context. The available project data do not contain sufficient information to identify individual patients. Processing took place in the protected project environment. These reference values were not reproduced in the thesis, and the submitted screenshots were sanitized.

3.3 Baseline Assessment and Revision Process

The baseline comprised the P5 pipeline, the Gantry 2 dashboard, and source handling for a limited 2025 period (Elmiger, 2026a). The assessment combined repository review, execution of the baseline system, the preceding report, the project agreement, the formative evaluation, and discussions with PSI/CPT and the academic supervisors.

The assessment identified issues in historical and cross-system compatibility, traceability and publication, dashboard orientation and contextual analysis, and evidence for practical usefulness. Revision was iterative. Because no complete statement-to-change log was maintained, the evaluation trace maps themes to final design responses rather than attributing individual interface elements to single participant comments.

3.4 Data Basis and Source Contracts

The inputs are CSV exports organized by treatment system and source family. A source family is a recurring category of export with a common operational purpose. A source contract is the versioned set of rules that states how files in that family are recognized, parsed, standardized, and validated. Historical inspection considered file naming, delimiters, headers, timestamps, columns, data types, missing fields, duplicates, and known malformed structures. Historically different layouts were represented as named source variants in these contracts, and the file-reading code did not guess among unknown layouts.

Table 3.2 lists the integrated source families and their main analytical role.

Within RQ1, “previously unseen variants” means variants not supported by the P5 baseline but encountered and analysed during this project. Automatic support for arbitrary future files is outside this definition. Unsupported structures require review, a documented contract, and corresponding tests before publication.

3.5 RQ1: Pipeline Extension and Validation

Processing Strategy.

Table 3.2: Integrated source families and intended analytical use.

System	Source family	Main analytical use
Gantry 2	BTVS (Beam Tune Verification System)	Long-term beam-transport trends, tune-energy comparisons, and left/right paired-channel differences
Gantry 2	Field parameters	Treatment-field, energy, patient, fraction, and gantry-angle context
Gantry 2	ILK (interlocks)	Source-system safety-interlock review
Gantry 2	Monitor statistics	Dose-monitor and correction statistics
Gantry 2	Position residuals	Residual summaries by patient, fraction, field, and angle
Gantry 2	Warnings	Measured, expected, tolerance, status, and treatment context where available
OPTIS 2	BTVS	Beam-transport trends with patient and fraction context
OPTIS 2	ILK (interlocks)	Source-system safety-interlock review
OPTIS 2	NBVS / NPVS (PSI source labels)	Nozzle beam- and position-verification measurements
OPTIS 2	Nozzle settings	Nozzle configuration by patient and fraction
OPTIS 2	Warnings	Operational warning records

The pipeline separates source-oriented ingestion, standardization, and monitoring-oriented preparation through Bronze, Silver, and Gold stages. The separation is intended to isolate source-specific handling from the stable data structures used for analysis and presentation. Bronze preserves the source-oriented representation and how each physical file was interpreted. Silver maps supported variants to a canonical, or standardized, structure with stable names and types, constructs timestamps, normalizes values, and removes deterministic duplicates. Gold combines related records across prepared data (data joins) and produces the summaries required by the monitoring tasks.

Runtime settings and source assumptions are kept in versioned configuration instead of being scattered through program code. Contracts define the applicable layout, aliases, required fields, type rules, duplicate keys, and validation expectations. The effective configuration receives a content hash—a digital fingerprint calculated from its contents—so that changes in source interpretation can be distinguished from reruns under unchanged settings.

Discovery is restricted to configured input directories. A source file must be observed as stable before processing so that a file still being copied is not ingested prematurely. Changed inputs invalidate the affected source-year partition, which is the unit rebuilt when upstream data change. Unchanged inputs may reuse an existing data artefact when the source state, transformation definition, and relevant configuration agree.

Architecture and Responsibility Boundaries.

The implementation separates controlled input synchronization, explicit source interpretation, standardization, analytical preparation, and presentation. Figure 3.1 summarizes the main data flow and responsibility boundaries. Source exports first enter a read-only landing area. Discovery and versioned source contracts determine which files and layouts are supported before source-oriented ingestion into Bronze. Silver then canonicalizes the supported variants, and Gold prepares monitoring-oriented joins and aggregates. Gold products form the stable interface for the dashboard and the exploratory anomaly-scoring branch; the dashboard does not read the source exports directly. A shared metadata and lineage store records source observations, artefact hashes, validation outcomes, configuration state, publication decisions, and derivation relationships across the processing stages.

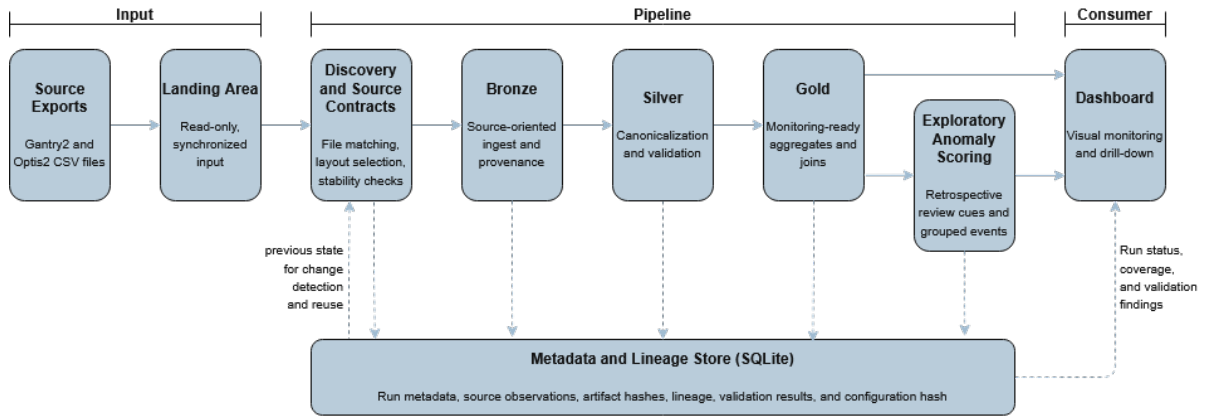


Figure 3.1: Logical architecture and responsibility boundaries of the monitoring artefact. The diagram separates synchronized input and source interpretation from staged processing, exploratory anomaly scoring, dashboard consumption, and cross-cutting metadata and lineage recording.

The repository mirrors these boundaries through separate pipeline, dashboard, shared configuration/statistics, source-contract, and deployment areas. This keeps source interpretation apart from presentation and environment-specific deployment concerns; the code layout itself is an implementation choice rather than a research result.

Validation, Publication, and Traceability.

Validation is applied before publication. Data-quality methodologies distinguish multiple quality dimensions and assessment rules (Batini et al., 2009); in this project, validation rules are therefore source- and product-specific, with severity determined by the consequences of a defect for downstream use. Findings that make a canonical product unsafe block publication; non-fatal findings remain visible as warnings.

Figure 3.2 illustrates the stage-local reuse and rebuild mechanism using Bronze as the concrete example. After a landing file has passed the stability check, the pipeline computes a Bronze build fingerprint from the observed source state, the applicable source definitions, and relevant stage options. If a reusable published Bronze artefact has the same fingerprint, that artefact is reused for the current run. Otherwise, the stage builds a candidate and evaluates its validation results before publication. In the figure, “Validation OK?” denotes satisfaction of the configured publication criteria; recorded warnings may still be present. A publication-blocking candidate is quarantined and leaves the current published state unchanged. Silver and Gold apply the same stage-local principle, but their fingerprints are derived from the current upstream artefacts together with their own transformation definitions and options.

A successfully published candidate is retained as an immutable data artefact, meaning that the stored version is not modified after creation. It is content-addressed: its identity includes a hash of the file contents. Only after validation does the pipeline update the recorded current version used by downstream readers. The previous valid artefact remains available if rebuilding or publication fails. Execution metadata record source observations, artefacts, validation outcomes, configuration hashes, lineage, publication state, and reuse relationships.

Reproducibility was assessed at three levels: equivalent outputs for unchanged supported inputs and configuration, traceability from a published product to its sources and transformation state, and successful execution of the release in both development and production-like environments.

Technical Verification and Supplementary Evidence.

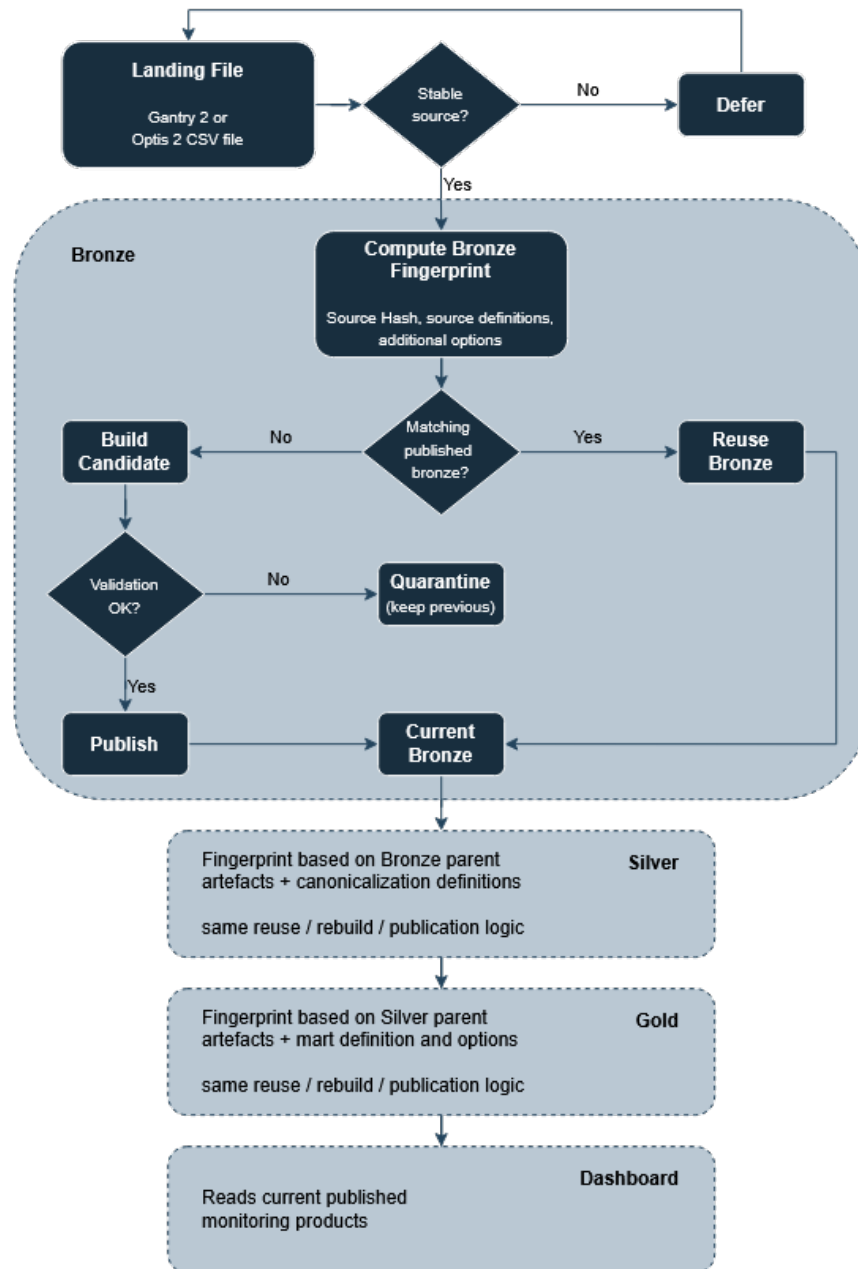


Figure 3.2: Stage-local reuse, rebuild, validation, and publication logic, illustrated for Bronze. A matching reusable published artefact is reused; otherwise a candidate is built and may become current only after the publication criteria are satisfied. Blocking findings quarantine the candidate without replacing the published state. Silver and Gold apply the same mechanism with their respective upstream artefacts and transformation definitions.

The implementation was checked through automated code-quality checks (linting), static type checking (checking whether values are used consistently with their declared data types without running the application), automated tests, complete pipeline executions, and release-specific deployment tests. The production-like test used an AlmaLinux virtual machine as a RHEL-compatible proxy for the intended environment. The batch pipeline and continuously available dashboard were deployed as separate operating-system-managed services, reflecting the intended operating model in which new data can be published without rebuilding or restarting the dashboard.

The release test checked whether the fixed release could be rebuilt, representative dashboard functions were available, the operating-system-managed services started and restarted cleanly, and unchanged-input execution avoided unnecessary rebuilding or dashboard interruption. This tested technical deployability on the proxy environment, not operational acceptance on the final PSI infrastructure.

The submitted repository contains the implementation, versioned configuration, automated tests, and deployment definitions, while the sanitized evidence archive retains the release-specific records used for technical verification (Appendix B). The detailed execution audit trail remains in the evidence archive; the main report includes only the outcomes used in the research argument.

3.6 RQ2: Dashboard Revision Method

Task Abstraction.

The redesign started from representative monitoring tasks rather than the existing page structure. The recurring sequence was to establish data readiness, review recorded events, inspect trends, compare relevant context, and investigate unusual observations. This follows the overview-to-detail principle (Shneiderman, 1996) and separates domain tasks from visual and implementation decisions (Munzner, 2009).

Table 3.3 shows the resulting task-to-view mapping.

Table 3.3: Task-to-view mapping used for the dashboard revision.

Monitoring task	Main view or interaction	Required context
Establish data readiness	Overview and data-quality pages	Latest product date, run state, coverage, validation findings
Review recorded events	Warnings and interlocks page	System, time, record type, measured/expected values, available treatment context
Inspect a long-term trend	System-specific daily trend page	Date range, parameter, daily summary, observation count, reference distribution
Compare related conditions	Energy-band, paired-channel, nozzle, or angle views	Energy, channel, angle, patient, fraction, or selected day
Investigate an unusual observation	Candidate table and contextual link	Detection method, score, driver, event period, and contributing records

Design Principles.

Pages were grouped by treatment system and monitoring purpose. A shared date window preserves temporal context, while repeated page structure, limited task-relevant controls, clear labels, observation counts, and contextual links provide consistency. Conventional line charts,

bars, heatmaps, summary cards, and tables were chosen because familiar encodings were sufficient; the design work focused on task organization and interpretation.

System-specific context was preserved. Gantry 2 views emphasize energy bands, paired channels, fields, residuals, and gantry angles; OPTIS 2 views retain patient, fraction, and nozzle context. Warnings and interlocks are presented separately from inferred anomaly candidates to preserve the distinction between source-system records and model-derived review cues.

3.7 Exploratory Anomaly-Oriented Method

The anomaly component was an exploratory feasibility study. It combined transparent robust-statistical deviations from preceding observations with Isolation Forest scoring (Chandola et al., 2009; Liu et al., 2012). Historical scoring was time-ordered: each observation was assessed only from earlier data, avoiding use of future information, with explicit states for insufficient history and transitions between segments.

Candidates were classified as statistical only, machine-learning only, or detected by both methods. Adjacent candidates were grouped into review events and linked to the corresponding dashboard context. Model provenance and configuration identifiers were retained. Exact thresholds, features, and model parameters are versioned in the submitted repository. The resulting workflow converts unusualness into a traceable review queue without making an automatic pass/fail decision.

No complete set of confirmed faults was available, so sensitivity, specificity, and predictive value were not estimated.

3.8 RQ3: Formative Evaluation

Participants and Setting.

The formative evaluation was conducted remotely on 22 May 2026 with dashboard version 0.3.0. Three PSI participants represented technician, physicist, and engineering roles. They were the complete set of prospective local users identified and available for the evaluation. The sample therefore covers the identified local user group well, while remaining too small for statistical generalization.

Tasks and Measures.

Participants completed five task groups: orientation, single-parameter trend review, interpretation of a BTVS relationship, interpretation of a patient-statistics relationship, and review of warning/interlock context.

Participants were asked to think aloud. The session took place in a shared remote meeting, but task ratings were entered independently before group discussion. After each task, participants rated ease, interpretation confidence, clarity, and practical usefulness on seven-point scales. The ease item used the seven-point post-task ease format evaluated by Sauro and Dumas (2009), commonly referred to as the Single Ease Question (SEQ); the other items were project-specific diagnostic indicators.

After all tasks, participants completed the System Usability Scale (Brooke, 1996). Standard SUS scoring was applied.

Available Evaluation Material and Limitations.

The retained material contains task ratings, participant roles, SUS responses, and two short comments. Detailed moderator notes, task-completion times, formal success criteria, and interpretation-accuracy measures were not available, so the analysis is descriptive.

The evaluation characterizes perceived usability and usefulness of version 0.3.0 and supplied input for the redesign. No release-equivalent follow-up evaluation was completed, so there is no before-and-after comparison of user outcomes.

3.9 Analysis and Reporting Procedure

RQ1 was analysed against the criteria in Table 3.1. The 3 August published-data state was used for coverage and validation results. The 6 August release evidence was used for build, service, interaction, lifecycle, reuse, and continuity results. Unchanged-input executions were used to assess reuse behaviour, not complete-rebuild performance.

RQ2 was analysed by tracing each monitoring task to the required context, prepared product, implemented view, and interaction path. Screenshots document the resulting artefact. RQ3 was analysed descriptively: task ratings are reported as medians and ranges, SUS as individual values and a mean, and comments as formative qualitative evidence.

4 Results

This chapter reports the results used to answer the research questions. Detailed commands, configuration records, and logs are retained in the repository and evidence package described in Appendix B.

4.1 Overall Evidence Status

Table 4.1 summarizes the evidence available at hand-in. The data-processing work is supported by executed and retained technical evidence; the dashboard revision was technically exercised; and the user study applies to version 0.3.0 rather than the final interface.

Table 4.1: Evidence status by research question and additional project scope.

Area	Status	Scope and limitation
RQ1	Answered within scope	Reviewed Gantry 2 and OPTIS 2 variants were processed into validated, traceable products and exercised in a production-like environment; arbitrary future variants and the final PSI host were not tested
RQ2	Implemented and technically verified	Task-oriented views and interaction paths were implemented; improved clarity or performance was not measured
RQ3	Partially answered	Version 0.3.0 was evaluated with all three identified prospective local users; the final version was not evaluated
Exploratory anomaly workflow	Feasibility demonstrated	Traceable candidates and grouped review events were produced; no labelled-fault validation was possible
PSI operational acceptance	Outstanding	The final on-premises deployment was not completed before hand-in

Scope Closure Against the Project Agreement.

Table 4.2 compares the agreed work with the state reached at hand-in. Follow-up evaluation was conditional on feasibility, and exploratory work had lower priority than the core scope (Elmiger, 2026b). A release-equivalent user study and the final on-premises deployment could not be scheduled during the final project month because the required users and infrastructure staff were not sufficiently available.

Table 4.2: Planned scope and achieved state at hand-in.

Agreed area	State	Evidence and remaining work
Production-oriented ingestion concept	Technically achieved	Managed-service separation and a production-like release exercise were completed. Final PSI deployment and operational acceptance remain outstanding; the people needed for the infrastructure work were not sufficiently available during the final project month.
Historical Gantry 2 extension	Achieved within reviewed scope	Source contracts, published products, validation, lineage, and reuse are documented. Arbitrary future variants are not covered.
Selected OPTIS 2 integration	Achieved within agreed scope	Four prepared dashboard products cover the reviewed source families. The scope is limited to the available source semantics.
Dashboard revision	Implemented and technically verified	Task-oriented views, drill-down paths, screenshots, and browser smoke tests exercise the final implementation. Improved user performance was not measured.
Baseline usability evaluation	Completed	Version 0.3.0 task ratings, SUS, and qualitative comments were collected from three participants. The study is formative and descriptive.
Follow-up evaluation of the revised dashboard	Not completed	Availability among prospective users was too limited during the summer holidays to schedule a release-equivalent study before hand-in. Final-interface usability is therefore unverified.
Secondary alerting / historical special case	Not claimed as an operational result	Source warnings and interlocks remain separate from inferred candidates. No operational alarm mechanism or validated special-case detector is claimed.
Exploratory anomaly analysis	Feasibility demonstrated	Causal scoring, method classes, grouped review events, and contextual follow-up were produced. No labelled-fault or alarm validation was possible.

4.2 RQ1: Data Pipeline and Data Basis

Supported Source Scope.

Both systems were integrated through the same execution, metadata, validation, and publication framework, with source-specific differences encoded in the contracts. Gantry 2 included historical BTVS layouts and five additional operational source families. OPTIS 2 included BTVS, interlocks, nozzle verification, nozzle settings, and warnings. Known historical defects were handled through defined mappings or limited repairs, while unsupported structures were rejected instead of being guessed. Table 4.3 summarizes how this supported source scope maps to the resulting monitoring products.

Table 4.3: Implemented source scope by treatment system.

System	Integrated scope	Monitoring result
Gantry 2	BTVS (beam-tune verification), field parameters, interlocks, monitor statistics, position residuals, and warnings	Historical trend, comparison, field/residual context, and warning/interlock event products, with BTVS coverage beginning in 2014
OPTIS 2	BTVS, interlocks, NBVS/N-PVS nozzle-verification records, nozzle settings, and warnings	Trend, nozzle-context, patient/fraction, and warning/interlock event products, with coverage beginning in 2015

Published Data State.

The latest retained published-data state was produced by a successful execution completed on 3 August 2026. All ten prepared dashboard datasets were available. The products covered data from 2 January 2014 to 27 March 2026, with the exact interval depending on source availability and required joins.

The dashboard reported 5,413,520 rows (table records) written during the execution. This is cumulative processing volume across artefacts and stages, not a count of unique operational observations. Table 4.4 summarizes the availability and coverage of the prepared products; product-level row counts and coverage are also visible in Figure 4.1 and retained in the supplementary evidence package.

Table 4.4: Summary of the published dashboard-product state.

System	Products available	Coverage	Principal analytical content
Gantry 2	6 of 6	2014-01-02–2026-03-27	Daily BTVS, energy-band and paired-channel comparison, combined field/residual context, angle and patient context
OPTIS 2	4 of 4	2015-11-17–2026-02-20	Daily BTVS, same-energy comparison, nozzle context, and combined source detail
Overall	10 of 10	Latest data day 2026-03-27	Published inputs for the implemented monitoring workflows

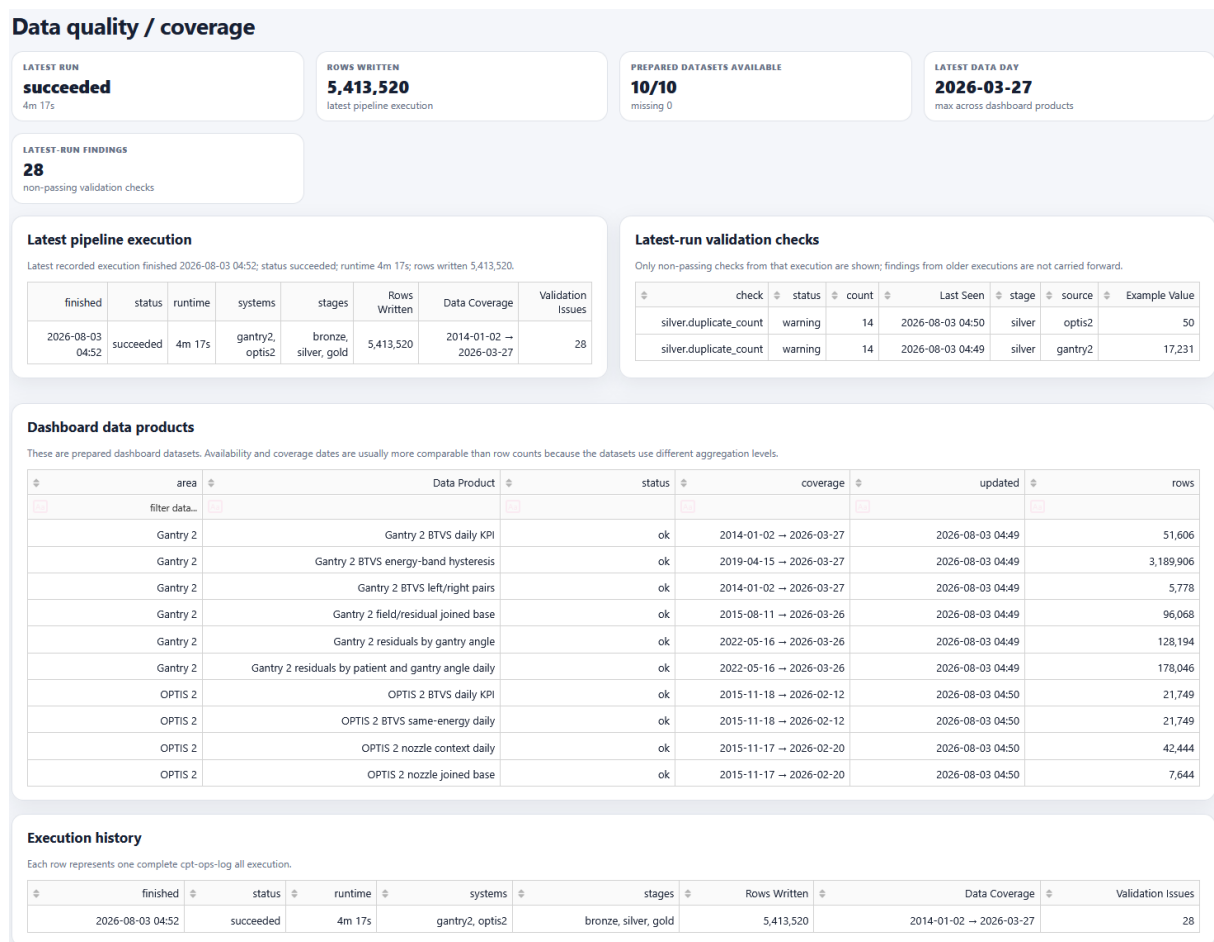


Figure 4.1: Data-quality page showing the latest published run, prepared products, validation findings, and execution history.

Validation, Traceability, and Reuse.

The published execution displayed 28 non-passing validation records, all grouped as duplicate-count warnings for Gantry 2 and OPTIS 2. These counts refer to validation records attached to affected Silver artefacts, not to 28 duplicate source rows. Deterministic duplicate removal occurred before publication, and no publication-blocking failure was shown. A successful run indicates that the configured publication criteria were satisfied even when non-blocking validation warnings remained.

Published artefacts were linked to source observations, configuration hashes, validation outcomes, and lineage records. Candidate outputs were retained separately until they met publication criteria, and the previous valid state remained available if a rebuild failed.

Two unchanged-input executions of the final release completed successfully in approximately 58 seconds each. The recorded plans reused Bronze, Silver, Gold, and anomaly products, consistent with the stage-local reuse mechanism illustrated in Figure 3.2, and the dashboard remained available throughout. These runs test rerun and reuse behaviour, not complete-rebuild performance.

The codebase contains a metadata-database verification routine that checks artefact existence and hashes, lineage references, and publication pointers, and this routine is covered by unit tests. It is not exposed as a release command, and no release-specific verification report over the final data state was retained.

Technical Verification and Production-Like Execution.

The local software quality gate reported no findings from the automated linting and formatting checks, no static type-checking issues, and 101 of 101 automated tests passing. The final release was then built from its fixed, version-tagged source and exercised on an AlmaLinux 9 test virtual machine. Batch processing and the continuously available dashboard ran as separate operating-system-managed services, matching the intended deployment model.

Table 4.5 summarizes the release checks relevant to RQ1. The supplementary evidence package retains the detailed list of tested web pages and requests, service start/stop/restart logs, service definition, container-image records, and file checksums.

Table 4.5: Production-like verification of the final release.

Criterion	Result	Interpretation
Tagged source built successfully	Passed	The submitted release could be built in the test environment
Representative dashboard functions	Passed	Core pages and interactions were technically functional
Controlled service restart	Passed	The dashboard terminated and restarted cleanly under service management
Two unchanged-input pipeline executions	Passed	Published products and anomaly outputs were reused without unnecessary rebuilding
Dashboard continuity during ingestion	Passed	The dashboard remained active and available during the batch runs
Final PSI infrastructure	Not tested	Technical deployability was demonstrated on a related platform, not operationally accepted at PSI

4.3 RQ2: Dashboard and Interaction Revision

Implemented Monitoring Workflow.

The revised dashboard is organized by treatment system and monitoring purpose, with internal processing modules kept out of the navigation structure. The overview reports data readiness for both systems and exposes three main review paths: recorded events, system trends, and unusual observations. A shared observation window persists across pages.

Figure 4.2 shows the revised starting point. Appendix A documents the same workflow in more detail: Figure A.1 records the overview, Figure A.2 the Gantry 2 trend and energy comparison, Figure A.3 the anomaly-candidate workflow, Figure A.4 the observation-level hover context, Figure A.5 the warning/interlock workflow, and Figure A.6 the data-quality and coverage view.

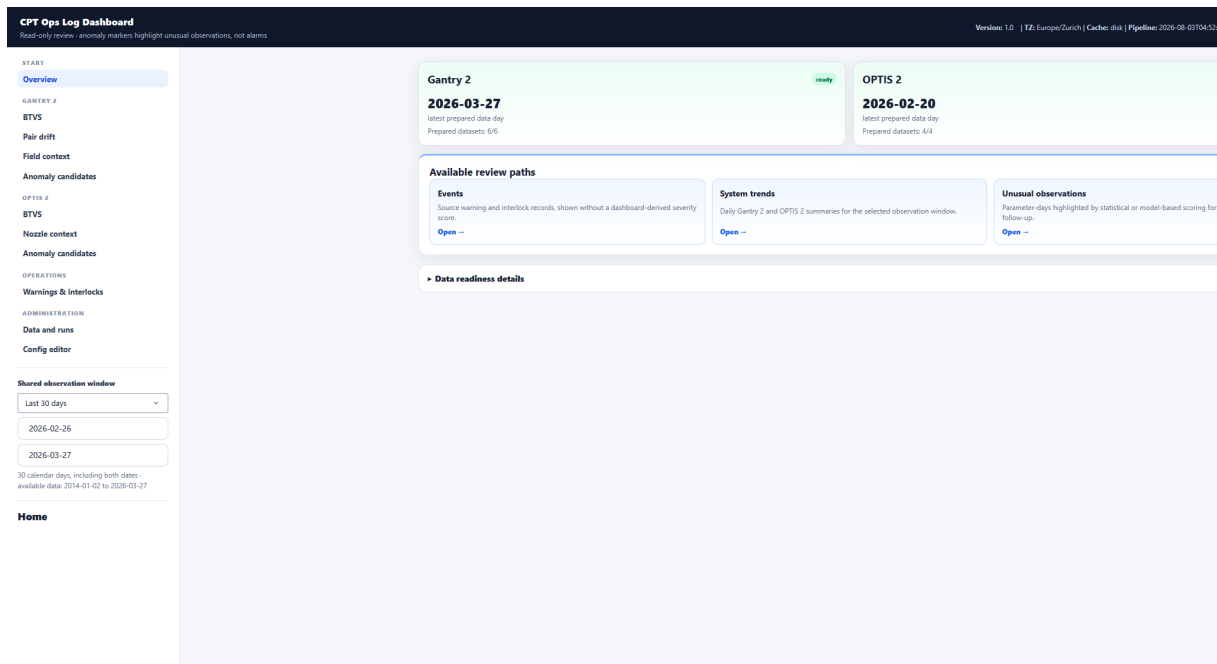


Figure 4.2: Revised overview with data-readiness status and principal review paths.

The implementation provides the task paths defined in Table 3.3: data-readiness review, warning/interlock review, long-term trend inspection, comparison of relevant operating context, and contextual follow-up of anomaly candidates. Gantry 2 and OPTIS 2 retain different analytical grains within a shared navigation structure, and source-system warnings and interlocks remain separate from inferred anomaly candidates. A release-specific browser smoke test of representative pages, requests, and interactions completed successfully; it tested technical operation rather than user interpretation or efficiency.

4.4 RQ3: Formative Evaluation of Practical Usefulness

Participant and Task Results.

All three identified prospective local users completed the formative evaluation of version 0.3.0. They represented technician, physicist, and engineering roles. Table 4.6 reports the task ratings as median and range.

Ratings were generally positive. Practical usefulness was consistently high, while clarity and confidence varied more when terminology, contextual information, or the next analytical step

Table 4.6: Task-level ratings for dashboard version 0.3.0, reported as median [range].

Task	Ease	Confidence	Clarity	Usefulness
Orientation	6 [5–7]	6 [4–6]	6 [3–6]	6 [6–7]
Trend review	6 [6–6]	6 [5–7]	5 [5–6]	6 [6–7]
Parameter relationship	6 [5–7]	6 [4–6]	6 [4–6]	6 [6–7]
Patient/statistics relationship	6 [3–7]	6 [5–6]	6 [5–7]	6 [5–7]
Warnings and context	6 [5–7]	5 [4–7]	5 [4–7]	7 [4–7]

was less evident. The two retained comments described the dashboard as almost complete with minor changes remaining and as a useful, well-developed tool.

System Usability Scale.

The individual SUS scores were 87.5, 92.5, and 80.0; the mean was 86.7. Table 4.7 reports the individual values used for this descriptive summary.

Table 4.7: System Usability Scale results for dashboard version 0.3.0.

Participant	SUS score
P1	87.5
P2	92.5
P3	80.0
Mean	86.7

For these three participants, the scores indicate positive perceived usability of version 0.3.0; they do not measure interpretation accuracy.

Use of the Formative Findings.

Table 4.8 links the main baseline findings to the design changes implemented afterward. None of those changes was tested in a follow-up user study.

Table 4.8: Trace from formative baseline evidence to the final dashboard response.

Baseline evidence	Interpretation	Final design response	Final version evaluated?
Orientation and clarity varied	Initial orientation and next steps were not uniformly clear	Overview page, grouped navigation, shared date context, and stated review paths	No
Trend usefulness exceeded trend clarity	More visible context and guidance were needed	Daily summaries, observation counts, energy-band comparison, hover values, and guidance text	No
Relationship tasks showed wider response ranges	Related parameters required dedicated contextual paths	Paired-channel, nozzle, angle, selected-day, and contributing-record views	No
Warning/context confidence and clarity were lower	Event records required a dedicated review workflow	Filterable warning/interlock page separated from anomaly candidates	No

4.5 Exploratory Anomaly-Oriented Results

Historical scoring and grouped review events were produced for both systems. Within the recorded dashboard review window, Gantry 2 contained 371 candidate parameter-days and

OPTIS 2 contained 222; one parameter-day represents one monitored parameter on one day. Most candidates were produced by the robust statistical method rather than Isolation Forest. Table 4.9 separates the candidate volume by method agreement.

Table 4.9: Exploratory candidate counts in the recorded dashboard review window.

System	Both methods	Machine-learning only	Statistical only	Total
Gantry 2	54	2	315	371
OPTIS 2	19	1	202	222

Without labelled outcomes, these counts describe review workload rather than detection performance or machine reliability. They show that method-specific candidates can be retained and inspected in context; fault-detection and alarm validity were not evaluated.

5 Discussion

5.1 Overall Interpretation

The project reached its primary technical objective within the scope defined in Chapter 3. The P5 prototype was extended to process reviewed historical Gantry 2 and OPTIS 2 data, publish validated monitoring products, support the intended analytical paths, and run under the production-oriented service model in a RHEL-compatible test environment. The evidence is strongest for the data-processing work. The final dashboard was technically exercised, while user evidence comes from the earlier version 0.3.0.

5.2 RQ1: Data Pipeline and Data Basis

The pipeline results support the use of source contracts, separated processing responsibilities, validation before publication, and recorded provenance for the reviewed Gantry 2 and OPTIS 2 data. These choices are consistent with work on reproducible processing and data provenance (Freire & Chirigati, 2018; Leipzig et al., 2021; Moreau & Missier, 2013) and with context-dependent data-quality assessment (Batini et al., 2009).

Historical source differences were represented in the contracts, unsupported structures were rejected, monitoring products were published after validation, and unchanged inputs reused existing products. The ten prepared dashboard products therefore provide a common analytical basis across two treatment systems and different historical periods while retaining the source state behind each published result. Reproducibility depends on the observed inputs, configuration, validation decisions, and publication state as well as the code; the separation described in Section 3.5 reflects the literature’s emphasis on identifiable processing context (Freire & Chirigati, 2018; Leipzig et al., 2021).

The production-like test extends verification beyond the developer machine: the tagged release was built, served, restarted, and exercised with repeated batch execution in a RHEL-compatible environment. This establishes technical deployability in that setting, but not acceptance on the final PSI infrastructure. The robustness claim is also limited to the reviewed variants and their available semantic interpretation. A stable column name could conceal a change in meaning, daily aggregation can hide short-lived behaviour, and the target host may expose filesystem, security, synchronization, monitoring, or organizational issues not present on the test VM.

5.3 RQ2: Visual Analytics and Interaction Design

The dashboard revision used an overview-to-detail workflow, a task-to-view mapping, repeated page structure, and preservation of system-specific context. This is consistent with Shneiderman’s overview-first principle (Shneiderman, 1996), Munzner’s nested model (Munzner, 2009), and interaction taxonomies that emphasize task-directed selection, filtering, exploration, and connection (Yi et al., 2007).

Consistency was implemented at the navigation and interaction level while the views remain system-specific. Gantry 2 and OPTIS 2 share navigation principles but retain the energy, channel, angle, patient, fraction, nozzle, warning, and source-record context required by their respective tasks. The workflow favours analytical progression: reviewers begin with data readiness or a broad observation and then move into task-relevant context. This follows the outer levels of the nested model—domain task and abstraction—before chart and implementation choices (Munzner, 2009).

The dashboard can preserve a shared time context, expose observation counts, and link to contributing records because the prepared products retain those analytical relationships. Interface structure cannot compensate for context discarded during data preparation.

The browser smoke test verified representative navigation and interactions, not user comprehension, interpretation accuracy, task efficiency, or consistency. The formative evaluation motivated several redesign choices, but changes introduced after version 0.3.0 have no user-based outcome evidence. The final implementation can therefore be assessed as technically realized, not as demonstrably easier or more accurate to use.

5.4 RQ3: Evaluation of Practical Usefulness

The formative evaluation indicates that the three identified prospective local users perceived version 0.3.0 as usable and practically relevant. The mean SUS score was 86.7, and task ratings were generally positive. Variation in clarity and confidence pointed to problems with orientation, context, and the visibility of the next analytical step. These findings provided concrete input for the subsequent revision.

The findings apply to version 0.3.0. A follow-up evaluation had been planned where feasible (Elmiger, 2026b), but no release-equivalent study was completed before hand-in. The baseline results therefore identify useful redesign themes but do not provide a before-and-after measure of usability, interpretation accuracy, efficiency, or practical usefulness.

All three identified prospective local users participated, which gives good coverage of that small local group but is insufficient for population inference. The shared remote setting and developer involvement may also have influenced responses. Completion times, formal success criteria, and interpretation-accuracy measures were unavailable, so the evaluation is stronger as evidence of perceived usability and usefulness than of measured task effectiveness.

5.5 Exploratory Anomaly-Oriented Extension

The anomaly extension supports past-only historical scoring, recorded model provenance, method-specific candidates, and contextual follow-up. Robust statistics produced a broader review queue than Isolation Forest in the recorded window, but without labelled faults this difference says nothing about comparative detection performance. Statistical unusualness may reflect a technical change, an operating regime, a data-quality issue, or uncommon but valid operation (Chandola et al., 2009). Keeping model-derived candidates separate from source-system warnings and operational alarms preserves the role of domain experts in deciding relevance.

Further threshold tuning without labels would not resolve the underlying validity problem. The more useful next step is structured expert review of a predefined candidate sample, with retained judgements that can later support threshold calibration or supervised evaluation. Until then, the anomaly component should be used as a prioritization and investigation aid.

5.6 Threats to Validity

Construct and Data Validity.

The robustness claim is confined to correct processing within declared source contracts, controlled handling of invalid input, and reproducible publication. Arbitrary future files were outside that scope. Canonical mappings depend on the available documentation and domain interpretation, and daily summaries may conceal short events or unstable low-count periods.

Several design choices mitigate these risks within the stated scope. Source assumptions are versioned, unsupported structures are rejected, validation is tied to publication, and product coverage and observation counts remain visible. The reported rows-written value is defined as processing volume, validation-warning counts refer to recorded checks, and anomaly-candidate counts refer to model output volume. These definitions avoid interpreting those quantities as unique observations, duplicate source rows, or fault frequency. Semantic completeness and sub-daily interpretation remain residual limitations.

Internal and Implementation Validity.

Automated tests, static checks, release-specific builds, repeated executions, service start/stop/restart tests, and browser functionality checks reduce the risk of straightforward implementation defects. Together they cover several implementation layers, but they do not cover every historical file, malformed input, semantic interpretation, or displayed value. The browser smoke test checks technical operation. A metadata-database verification routine exists and is unit-tested, but it is not exposed as a release command and no release-specific report over the final data state was retained.

Confidence is strongest for the exercised release paths and recorded products. Arbitrary inputs and exhaustive end-to-end semantic correctness were not verified. The retained lineage and reuse records remain available, but an independent release-specific metadata-verification report is absent. The handover plan therefore includes a controlled full run followed by retained execution of the implemented verification routine.

Evaluation Validity.

The formative evaluation involved the three identified prospective local users, but the small case-specific sample does not support population-level inference. The shared remote setting and developer involvement may have influenced responses, and the retained material did not include completion times, predefined task-success criteria, or interpretation-accuracy measures. Most importantly, the evaluation concerned version 0.3.0 rather than the final interface; its results therefore support redesign decisions and perceived usability, not validated effectiveness of the final release.

External and Operational Validity.

The case is specific to PSI/CPT source data, users, and monitoring tasks. Applying the system elsewhere would require new source analysis, semantic validation, task definition, and operational acceptance. The transferable elements are the architecture and evidence strategy—source

contracts, staged processing, validation-controlled publication, provenance, task abstraction, and scoped evaluation. The concrete source definitions and dashboard pages are specific to this case.

AlmaLinux provided a RHEL-compatible test environment, but it was not the final PSI host. The deployment result supports technical readiness for handover, not completed production deployment. The final on-premises deployment was not completed before hand-in. Storage, permissions, synchronization, security controls, service ownership, and monitoring may differ on the target host and remain untested there. The successful AlmaLinux run should therefore not be extrapolated to operational acceptance.

5.7 Contribution of the Thesis

The main technical contribution is the integration of heterogeneous historical exports from Gantry 2 and OPTIS 2 into a traceable, publication-controlled monitoring pipeline and a task-oriented dashboard. Source interpretation, validation, lineage, and rerun behaviour are recorded, while the interface preserves the different operating contexts of the two treatment systems. The visual encodings themselves are conventional; the contribution lies in the data preparation, task organization, and connection between views.

A second contribution is the release and evidence package prepared for handover. It records build identity, service behaviour, reuse, continuity, and representative interface state, and makes clear which conclusions come from technical execution and which come from the formative user study. The result is an inspectable prototype for the supported PSI/CPT case rather than a claim of clinical validation or completed production acceptance.

5.8 Practical Implications and Handover

The artefact is suitable for technical handover as a read-only monitoring prototype. The submitted repository provides the implementation and versioned configuration, and the evidence package provides the detailed release and execution record. The codebase separates pipeline, dashboard, shared configuration/statistics, source contracts, tests, and deployment definitions, while the release process applies automated code-quality checks, type checking, automated tests, container-image builds, and release packaging. These features support maintainability and reproducibility; sustained operation still requires clear ownership.

The remaining work falls into three categories:

1. **Operational acceptance:** deploy the fixed, versioned release on the final PSI infrastructure, execute and retain a controlled full run followed by metadata verification using the implemented verification routine, and confirm environment-specific service, storage, security, and synchronization behaviour;
2. **Scientific and user validation:** evaluate the release-equivalent dashboard with predefined task-success and interpretation criteria, and review a selected sample of anomaly candidates with domain experts; and
3. **Operational governance:** assign ownership for source-contract changes, failed runs, validation findings, release updates, and dashboard support.

The main maintainability risk lies in the transition from a project-maintained prototype to an operationally owned service whose source contracts and validation rules will require controlled change over time. Completing the acceptance, validation, and ownership activities above would provide the evidence needed to judge sustained operational usefulness.

6 Conclusion

The project extended the Gantry 2 monitoring prototype to cover reviewed historical Gantry 2 and selected OPTIS 2 data through a traceable, read-only processing and visualization workflow. Source contracts, staged transformation, validation before publication, lineage, and reuse were applied to ten prepared dashboard products covering data from 2014 to March 2026. The fixed release also passed the retained build, service, restart, reuse, continuity, and browser checks in the AlmaLinux test environment. These results answer the data-processing question within the reviewed source scope; arbitrary future formats and the final PSI infrastructure were not tested.

The dashboard was reorganized around monitoring tasks rather than implementation modules. Its overview-to-detail paths connect trends and recorded events with energy, channel, angle, patient, fraction, nozzle, anomaly, and source-record context as required by the two treatment systems. The final implementation was technically exercised, but its effect on user performance cannot be inferred from those tests. The formative study involved all three identified prospective local users and produced generally positive task ratings and a mean SUS score of 86.7, together with concrete redesign themes. Because that study used version 0.3.0, it does not establish the usability of the final interface.

The exploratory anomaly workflow produced traceable, time-ordered candidates from robust statistical indicators and Isolation Forest scoring. Without labelled faults or expert validation, the candidate counts describe review workload rather than detection or alarm performance. The workflow should therefore remain an aid for expert investigation rather than an operational alarm mechanism.

The next steps are a controlled deployment of the fixed release on the final PSI infrastructure, a release-equivalent user evaluation with predefined task-success and interpretation criteria, and expert review of a selected anomaly-candidate sample. Operational ownership is also needed for source-contract changes, validation findings, failed runs, releases, and dashboard support. These activities determine whether the prototype is suitable for sustained operational use.

References

- Actis, O., Meer, D., König, S., Weber, D. C., & Mayor, A. (2017). A comprehensive and efficient daily quality assurance for PBS proton therapy. *Physics in Medicine & Biology*, 62(5), 1661–1675. <https://doi.org/10.1088/1361-6560/aa5131>
- Almasi, S., Bahaadinbeigy, K., Ahmadi, H., Sohrabei, S., & Rabiei, R. (2023). Usability evaluation of dashboards: A systematic literature review of tools. *BioMed Research International*, Article 9990933. <https://doi.org/10.1155/2023/9990933>
- Arjomandy, B., Taylor, P. A., Ainsley, C. G., Safai, S., Sahoo, N., Pankuch, M., Farr, J. B., Park, S. Y., Klein, E. E., Flanz, J., Yorke, E. D., Followill, D. S., & Kase, Y. (2019). AAPM task group 224: Comprehensive proton therapy machine quality assurance. *Medical Physics*, 46(8), e678–e705. <https://doi.org/10.1002/mp.13622>
- Bach, B., Freeman, E., Abdul-Rahman, A., Turkay, C., Khan, S., Fan, Y., & Chen, M. (2023). Dashboard design patterns. *IEEE Transactions on Visualization and Computer Graphics*, 29(1), 342–352. <https://doi.org/10.1109/TVCG.2022.3209448>
- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6), 574–594. <https://doi.org/10.1080/10447310802205776>
- Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Computing Surveys*, 41(3), Article 16, 1–52. <https://doi.org/10.1145/1541880.1541883>
- Brooke, J. (1996). SUS: A “quick and dirty” usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Article 15, 1–58. <https://doi.org/10.1145/1541880.1541882>
- Cui, W. (2019). Visual analytics: A comprehensive overview. *IEEE Access*, 7, 81555–81573. <https://doi.org/10.1109/ACCESS.2019.2923736>
- Ding, X., Younkin, J. E., Shen, J., Bues, M., & Liu, W. (2021). A critical review of the practices of proton daily quality assurance programs. *Therapeutic Radiology and Oncology*, 5, Article 22. <https://doi.org/10.21037/tro-21-11>
- Elmiger, C. (2026a). *Dashboard to display daily measured and evaluated parameters from Gantry2 at the center for proton therapy at PSI* (P5 Project Report). FHNW School of Computer Science.
- Elmiger, C. (2026b). *A validated visual analytics prototype for monitoring and alerting of gantry 2 and OPTIS 2 parameters at PSI CPT* (Project Agreement for Student Projects No. 26FS_IIT52). FHNW School of Computer Science and Paul Scherrer Institute, Center for Proton Therapy.
- Freire, J., & Chirigati, F. (2018). Provenance and the different flavors of computational reproducibility. *IEEE Data Engineering Bulletin*, 41(1), 15–26. Retrieved August 5, 2026, from <https://sites.computer.org/debull/A18mar/p15.pdf>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- Huber, P. J., & Ronchetti, E. M. (2009). *Robust statistics* (2nd ed.). John Wiley & Sons. <https://doi.org/10.1002/9780470434697>
- International Organization for Standardization. (2018). *ISO 9241-11:2018, ergonomics of human-system interaction — part 11: Usability: Definitions and concepts* (International Standard No. ISO 9241-11:2018). International Organization for Standardization. Retrieved August 5, 2026, from <https://www.iso.org/standard/63500.html>

- Kanterakis, A., Potamias, G., Swertz, M. A., & Patrinos, G. P. (2018). Creating transparent and reproducible pipelines: Best practices for tools, data, and workflow management systems. In C. G. Lambert, D. J. Baker, & G. P. Patrinos (Eds.), *Human genome informatics: Translating genes into health* (pp. 15–43). Academic Press. <https://doi.org/10.1016/B978-0-12-809414-3.00002-4>
- Leipzig, J., Nüst, D., Hoyt, C. T., Ram, K., & Greenberg, J. (2021). The role of metadata in reproducible computational research. *Patterns*, 2(9), Article 100322. <https://doi.org/10.1016/j.patter.2021.100322>
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1), Article 3, 1–39. <https://doi.org/10.1145/2133360.2133363>
- Mitchell, S. N., Lahiff, A., Cummings, N., Hollocombe, J., Boskamp, B., Field, R., Reddyhoff, D., Zarebski, K., Wilson, A., Viola, B., Burke, M., Archibald, B., Bessell, P., Blackwell, R., Boden, L. A., Brett, A., Brett, S., Dundas, R., Enright, J., . . . Reeve, R. (2022). FAIR data pipeline: Provenance-driven data management for traceable scientific workflows. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2233), Article 20210300. <https://doi.org/10.1098/rsta.2021.0300>
- Montgomery, D. C. (2019). *Introduction to statistical quality control* (8th ed.). John Wiley & Sons.
- Moreau, L., & Missier, P. (Eds.). (2013, April 30). *PROV-DM: The PROV data model: W3C recommendation*. World Wide Web Consortium. Retrieved August 5, 2026, from <https://www.w3.org/TR/prov-dm/>
- Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 921–928. <https://doi.org/10.1109/TVCG.2009.111>
- Paul Scherrer Institute, Center for Proton Therapy. (n.d. a). *Gantry 2*. Retrieved August 7, 2026, from <https://www.psi.ch/en/protontherapy/gantry-2>
- Paul Scherrer Institute, Center for Proton Therapy. (n.d. b). *Optis*. Retrieved August 7, 2026, from <https://www.psi.ch/en/protontherapy/optis-2>
- Paul Scherrer Institute, Center for Proton Therapy. (n.d. c). *Therapy facilities*. Retrieved August 7, 2026, from <https://www.psi.ch/en/protontherapy/therapy-facilities>
- Richter Lagha, R., Burningham, Z., Sauer, B. C., Leng, J., Peters, C., Huynh, T., Patel, S., Halwani, A. S., & Kramer, B. J. (2020). Usability testing a potentially inappropriate medication dashboard: A core component of the dashboard development process. *Applied Clinical Informatics*, 11(4), 528–534. <https://doi.org/10.1055/s-0040-1714693>
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131–164. <https://doi.org/10.1007/s10664-008-9102-8>
- Sauro, J., & Dumas, J. S. (2009). Comparison of three one-question, post-task usability questionnaires. *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, 1599–1608. <https://doi.org/10.1145/1518701.1518946>
- Sauro, J., & Lewis, J. R. (2016). *Quantifying the user experience: Practical statistics for user research* (2nd ed.). Morgan Kaufmann.
- Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, 336–343. <https://doi.org/10.1109/VL.1996.545307>
- Taylor, P. A., Miles, E., Hoffmann, L., Kelly, S. M., Kry, S. F., Sloth Møller, D., Palmans, H., Akbarov, K., Aznar, M. C., Clementel, E., Corning, C., Effeney, R., Healy, B., Moore, A., Nakamura, M., Patel, S., Shaw, M., Stock, M., Lehmann, J., & Clark, C. H. (2023). Prioritizing clinical trial quality assurance for photons and protons: A failure modes and

- effects analysis (FMEA) comparison. *Radiotherapy and Oncology*, 182, Article 109494. <https://doi.org/10.1016/j.radonc.2023.109494>
- Valiati, E. R. d. A., Pimenta, M. S., & Freitas, C. M. D. S. (2006). A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, 1–6. <https://doi.org/10.1145/1168149.1168169>
- Xiao, Q., & Li, G. (2024). Application and challenges of statistical process control in radiation therapy quality assurance. *International Journal of Radiation Oncology*Biography*Physics*, 118(1), 295–305. <https://doi.org/10.1016/j.ijrobp.2023.08.020>
- Yi, J. S., Kang, Y. A., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1224–1231. <https://doi.org/10.1109/TVCG.2007.70515>

Declaration of Independent Authorship (Eigenständigkeitserklärung)

I hereby declare that the present thesis submitted for assessment is entirely the product of my own effort,

- that I have correctly cited all text passages and other sources, such as images, that do not originate from me, in accordance with standard academic citation rules, and that I have clearly identified all sources used;
- that I have declared in footnotes or in an “Index of auxiliary tools” (Hilfsmittelverzeichnis) all aids used, including AI-assistance systems such as chatbots (e.g., ChatGPT), translation tools (e.g., DeepL), paraphrasing tools (e.g., QuillBot), or programming applications (e.g., GitHub Copilot), and indicated their use at the corresponding parts of the work;
- that I have acquired all intangible rights to any materials I may have used, such as images or graphics, or that these materials were created by me;
- that the topic, the thesis, or parts of it have not been used in an assessment of another module, unless this was expressly agreed with the lecturer in advance and is stated in the thesis;
- that I am aware that my work may be checked for plagiarism and for third-party authorship of human or technical origin (artificial intelligence);
- that I am aware that the FHNW School of Computer Science will pursue a violation of this declaration of independent authorship or of the underlying student obligations under the applicable study and examination regulations, and that disciplinary consequences, including a reprimand or exclusion from the study programme, may result.

Place, date: Murgenthal, 7 August 2026

Name: Christoph Elmiger

Signature:



Use of AI-Assisted Tools (Hilfsmittelverzeichnis)

The preceding Declaration of Independent Authorship is the formal declaration of independent authorship. The following table documents the principal uses of generative-AI assistance in the work. The categories are deliberately broad because ChatGPT was used iteratively rather than for a small number of isolated prompts.

AI-based tool	Form of use	Affected parts of the work	Period
OpenAI ChatGPT, primarily GPT-5.5 and GPT-5.6	Idea development and critical review: brainstorming alternatives, questioning assumptions, and discussing implementation, evaluation, report structure, and presentation	Project reasoning and decisions documented across Chapters 1–6	2026 (P6)
OpenAI ChatGPT, primarily GPT-5.5 and GPT-5.6	Orthographic, linguistic, and structural review; alternative formulations; terminology checks; LaTeX troubleshooting; bibliography and citation-consistency checks	Front matter, Chapters 1–6, Appendices A–B, and References	2026 (P6)
OpenAI ChatGPT, primarily GPT-5.5 and GPT-5.6	Code review, debugging, interpretation of error messages, test ideas, library/API explanations, and proposed code fragments or implementation patterns	<code>src/</code> , primarily <code>src/cpt_ops_log_dashboard/</code> ; additional reviewed code where relevant	2026 (P6)
OpenAI ChatGPT, primarily GPT-5.5 and GPT-5.6	Assistance with technical documentation, Linux and shell commands, Podman/Quadlet questions, and review and sanitization of technical evidence	Deployment material in <code>deploy/</code> and the technical evidence package	2026 (P6)

AI assistance in software development was most extensive for Dash, Plotly, HTML, and CSS. SQL queries, Docker and systemd configuration, CI configuration, and YAML/TOML formatting were not generated with AI, although these and other parts of the codebase were at times reviewed or checked with AI assistance.

Where the tool proposed wording, code, commands, explanations, or review comments, these were treated as working material rather than as authoritative sources or evidence. I reviewed and adapted the material, executed and tested the software, and checked factual and scientific claims against project material, executed outputs, technical documentation, evaluation records, and cited literature. I remain responsible for the implementation, analysis, citations, conclusions, and submitted artefacts.

Other auxiliary tool. diagrams.net (draw.io) was used to create the pipeline architecture and stage-logic diagrams in Chapter 3. The diagram content, structure, and labels were created by the author.

A Interface Evidence

This appendix contains the six sanitized screenshots retained for the final submission. They record the implemented review workflow and the published data state used in the report. Patient and fraction columns remain visible where they are part of the source data structure, but no patient or fraction identifiers are shown. These figures are static implementation evidence. The accompanying sanitized evidence package, identified in Appendix B, retains the release, service, browser-interaction, service start/stop/restart, and pipeline-reuse records used for the technical verification.

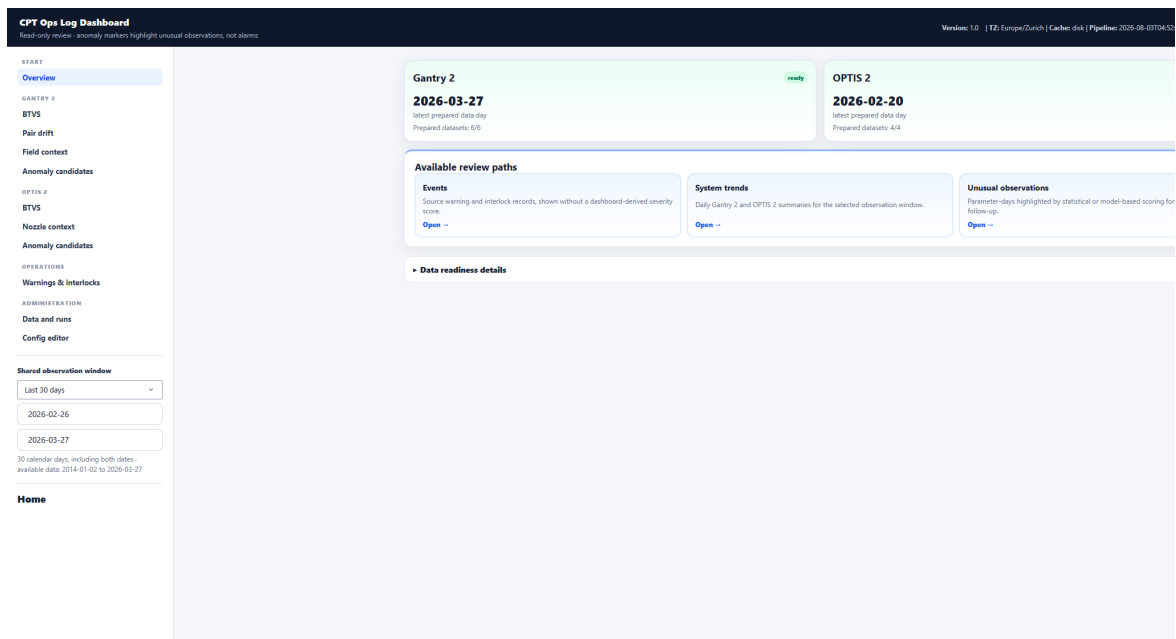


Figure A.1: Monitoring overview with data-readiness status, the shared observation window, and the three principal review paths.

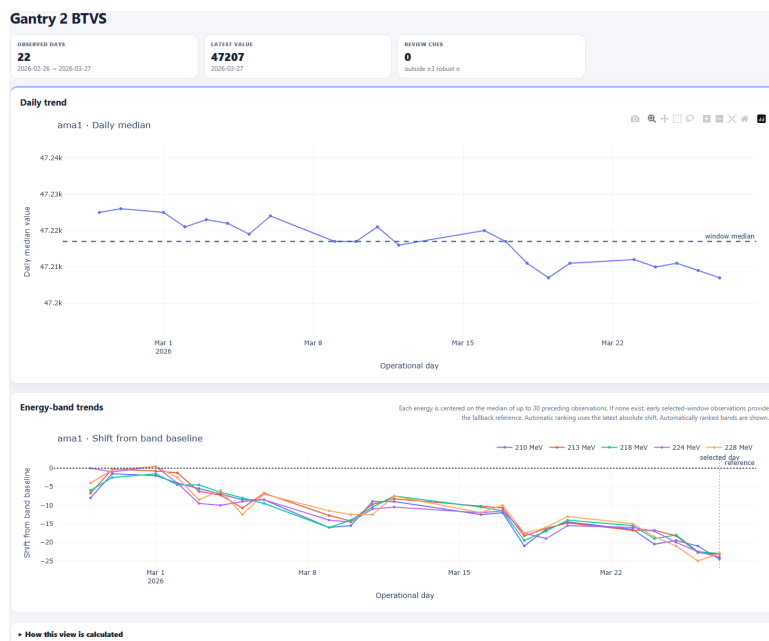


Figure A.2: Gantry 2 BTVS page with a daily trend, selected parameter, and energy-band comparison.

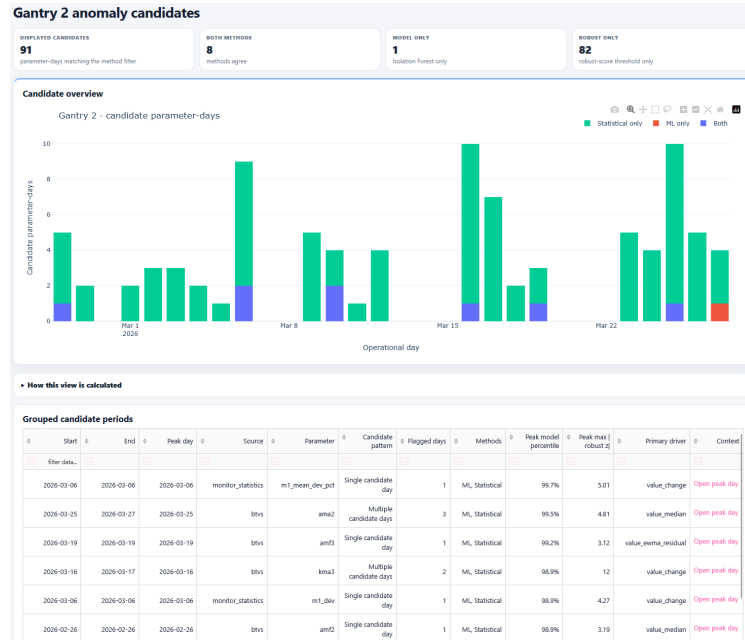


Figure A.3: Gantry 2 anomaly-candidate page with candidate volume, method classes, and grouped review periods.

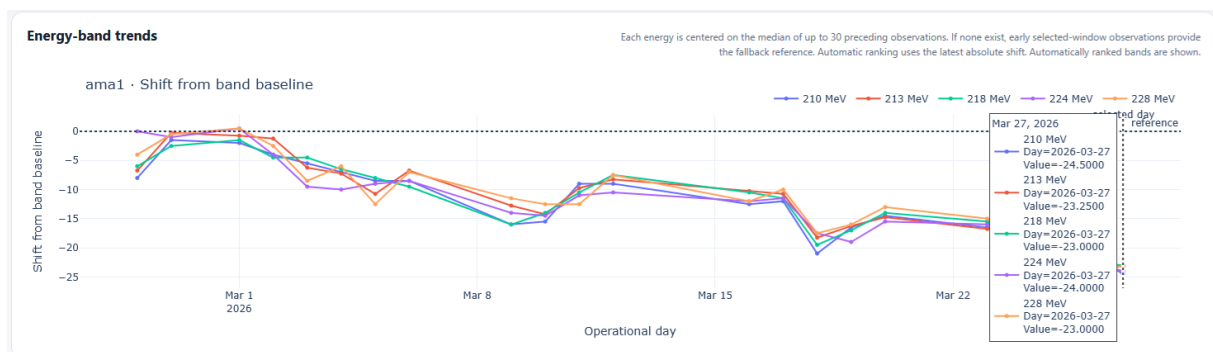


Figure A.4: Hover information in the Gantry 2 BTVS view, exposing the observation date, energy level, and the displayed shift relative to the reference baseline.

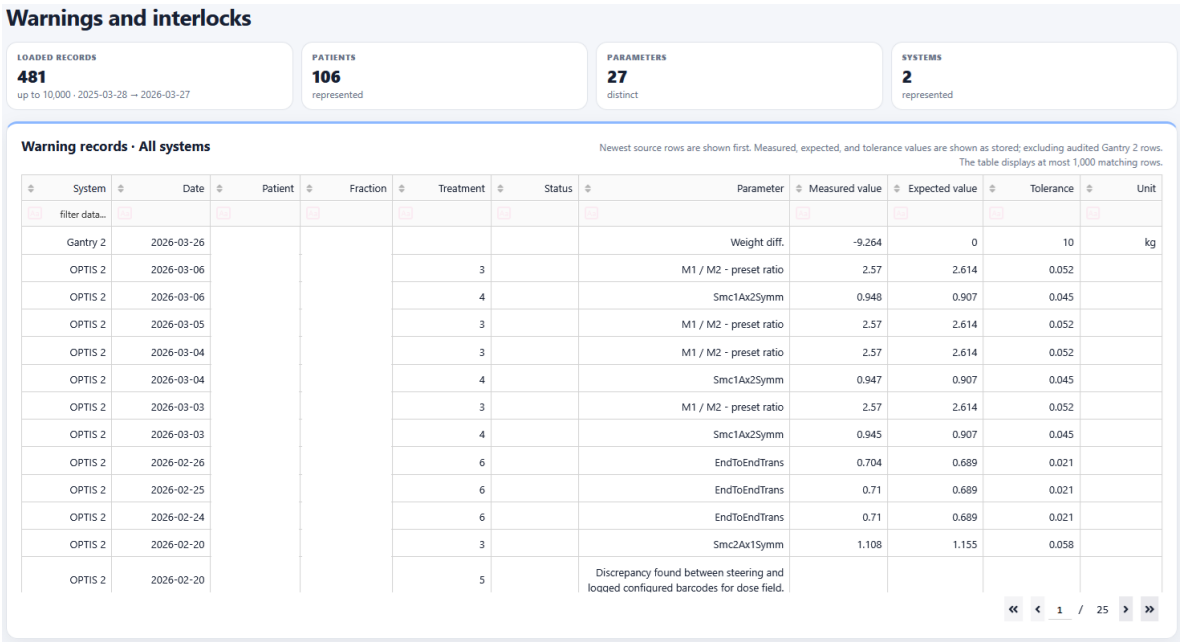


Figure A.5: Warnings and interlocks page with source-system records, filters, and available operational context.

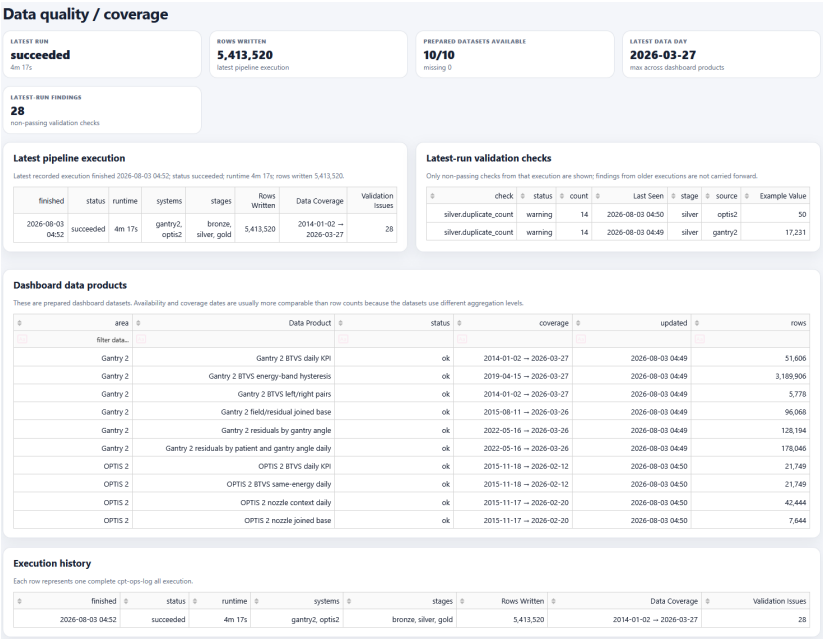


Figure A.6: Data-quality and coverage page showing the latest published pipeline execution, prepared products, validation findings, and execution history.

B Supplementary Material and Evidence

The hand-in package includes the source repository and a sanitized technical evidence archive. They support reproducibility and auditability but do not replace the results reported in Chapter 4. The report states the observations required for its conclusions; the supplementary material retains the detailed implementation and execution record. Table B.1 identifies the relevant artefacts and the technical revision used for verification.

Table B.1: Supplementary artefacts included with the hand-in.

Artefact	Identifier	Purpose
Source repository	Technical baseline: Git tag <code>v1.0.3</code> ; commit <code>2a6c003324d1</code>	The hand-in repository contains the implementation and final report source. The identifier names the fixed technical revision exercised during evidence collection; later report-only commits are not part of that tested baseline.
Sanitized evidence archive	<code>cpt-evidence-20260806-190929-public.tar.gz</code>	Sanitized records of the test environment, technical revision, local and managed-service execution, browser interaction, service lifecycle, repeated ingestion, dashboard continuity, and interface state

Git tag `v1.0.3` points to commit `2a6c003324d17245038322dfaa26e0cd65653bb2`. This is the technical baseline used during the 6 August 2026 evidence collection. It is intentionally not presented as the final hand-in commit: subsequent repository changes concern the report itself, while the verification evidence remains tied to the fixed technical revision. The evidence archive is identified by the following SHA-256 checksum, split over two lines for readability:

```
ce7b7e65246d948df5d86e728c636dba
1b0e2548854d7ff22214de484e07c7a
```

Table B.2 provides a compact route from the main technical claims to the corresponding records inside the sanitized archive. Paths are relative to the archive root.

Table B.2: Technical claim-to-evidence index for the sanitized archive.

Evidence area	Archive records
Test environment and tool versions	02_platform_public.txt 03_versions.txt
Technical baseline and local image build	04_repository_identity_public.txt 08_production_contract_status.txt 10_local_build_identity.txt 12_local_image_inspect.txt 13_local_gunicorn_version.txt
Managed service and deployed image	30_dashboard_unit.txt 34_deployed_image_inspect.txt 35_deployed_gunicorn_version.txt
Web availability and browser interaction	24_local_http_smoke.tsv 36_managed_http_smoke.tsv 44_browser_callback_smoke.txt
Lifecycle and restart behaviour	28_local_graceful_stop_logs.txt 29_local_stopped_container_state.txt 40_restart_identity.txt 41_restart_journal.txt 43_dashboard_logs_after_restart.txt
Pipeline execution, reuse, and dashboard continuity	50_ingest_run1_command.txt 52_ingest_run1_result.txt 53_ingest_run2_command.txt 55_ingest_run2_result.txt 56_ingest_dashboard_continuity.txt 60_acceptance_extract.txt
Interface state and archive integrity	screenshots/S01_overview.png to screenshots/S06_data_quality.png SHA256SUMS.txt

The CI configuration defines a tagged release job that packages the committed `deploy/` definitions as `production-contract-v1.0.3.tar.gz`. That CI-generated tarball was not available when the sanitized evidence set was collected; `08_production_contract_status.txt` records this gap. The committed deployment definitions, local build identity, and managed-service records are retained separately in the repository and archive, but the report does not claim that the missing CI artefact was verified or that deployment on the final PSI infrastructure occurred.

The technical evidence was collected on 6 August 2026. Operational logs and configuration records are retained because they make the release tests auditable, but the main report references them only at summary level. A separate raw archive with more detailed system information is retained privately. It is not required to interpret the thesis results; the submitted sanitized archive is the evidence set referenced by this report.