

Design and Verification of an Interpretable Low-Code Maintenance Decision Support Application for Automotive Assembly

Adriana ALDEA, Adriana FLORESCU, Catrina CHIVU

Department of Engineering and Industrial Management, Transilvania University of Brasov, Brasov, Romania

Abstract—Industrial maintenance digitalization is often associated with sensor-rich predictive models, yet brownfield factories may first need reliable decision support from sparse historical records and familiar software. This study designs and verifies an interpretable low-code application for maintenance scenario assessment in automotive assembly. The application combines planned production and cyclic calendar context, selects three similar historical situations, and displays normalized downtime, incident, efficiency, and failure-family indicators with an audit trail. The evaluation used twelve synthetic monthly records from which confidential and equipment-specific information was excluded. The application correctly reproduced all twelve stored monthly cases, generated valid results for thirty-six scenarios, rejected nine categories of invalid input, and responded correctly to forty-eight changes in planned production. An additional test estimated each month using the remaining eleven months. The proposed configuration produced an average difference of 5.52 points on the 0-100 indicator scale. It performed better than methods based only on production similarity or the historical average, although a simple calendar-based method achieved a lower difference. Alternative parameter settings produced comparable results. These findings confirm that the application operates consistently and transparently, but further testing with real industrial data and maintenance users is required before predictive effectiveness can be claimed.

Keywords—Decision support system; maintenance management; low-code application; spreadsheet engineering; explainable analytics; automotive assembly

I. INTRODUCTION

A. Problem Context

Manufacturing systems increasingly combine automation, connected equipment, and data-intensive decision processes. Smart manufacturing research accordingly emphasizes the integration of physical assets with computation, communication, and analytics [1]. Within this setting, maintenance is no longer only a technical response to failure; it is a coordinated management function that influences availability, quality, production stability, cost, and learning. The Smart Maintenance agenda links data-driven capability with human and organizational capability [2], while Maintenance 5.0 adds human centricity, resilience, and sustainability to the design requirements [3].

The dominant research trajectory is predictive maintenance based on sensors, large data streams, and machine-learning models. Reviews document substantial progress, but also

recurring problems: data scarcity, inconsistent labels, model transferability, integration effort, and weak operational explainability [4], [5]. These constraints are pronounced in brownfield automotive assembly, where event logs may be available, but synchronized condition-monitoring signals are not. Automotive studies also identify functional safety, heterogeneous operating conditions, and life-cycle data access as practical challenges [6]. A factory may therefore need useful and governable decision support before it is ready for a full predictive-maintenance platform.

Spreadsheet software remains widespread in this transitional space because maintenance engineers can inspect formulas, update records, and adapt outputs without a separate software-development cycle. Its accessibility is valuable, but unmanaged spreadsheets are also end-user applications and must be engineered accordingly. A maintenance workbook becomes decision support only when inputs, logic, outputs, traceability, validation, and change control are deliberately separated.

B. Research Questions, Objective, and Contributions

This study investigates how a governed, low-code workbook can support maintenance planning when the available history is sparse and monthly. The objective is not to replace condition monitoring or machine learning, but to provide a transparent intermediate layer that supports scenario assessment, exposes dominant failure families, and prepares a disciplined data foundation for later digitalization. Accordingly, the study addresses four research questions:

Research Question 1 (RQ1). How can a low-code spreadsheet application transform sparse monthly maintenance history and planned production into transparent scenario indicators for automotive assembly?

Research Question 2 (RQ2). Does the implemented application reproduce the documented procedure across historical replay, scenario, invalid-input, and traceability tests?

Research Question 3 (RQ3). How does the proposed estimator compare with simple baselines under leave-one-month-out testing, and how sensitive are its outputs to alternative production-calendar priorities and numbers of historical neighbors?

Research Question 4 (RQ4). Do the implemented design and governance controls make each tested output traceable and reconstructable from the disclosed calculation path?

- A layered, technology-light architecture that separates controlled inputs, historical data, scenario logic, indicators, visualization, and audit information.
- An interpretable comparison model that gives production similarity an eighty-percent priority and cyclic calendar proximity a twenty-percent priority.
- A privacy-preserving case-study protocol that removes organizational identifiers and replaces operational magnitudes with synthetic indices.
- A verification framework covering calculation agreement, scenario bounds, input rejection, sensitivity behavior, traceability, and spreadsheet-quality controls.

The remainder of the study reviews related work, specifies the research design, presents the application and algorithm, reports verification results, and discusses deployment limits and future extensions.

II. RELATED WORK

A. Predictive and Smart Maintenance

Machine-learning approaches can identify complex patterns in degradation data, but model accuracy alone does not ensure operational adoption [4]. Explainable predictive-maintenance research shows that operators need to understand the features, evidence, and uncertainty behind a recommendation, particularly when maintenance actions affect safety or production [7]. Recent work in the International Journal of Advanced Computer Science and Applications (IJACSA) applies Random Forest and Gradient Boosting models to real sensor data for preventive maintenance in fibreboard production [8]. Such models represent an important target state, but they require a larger and more integrated data foundation than the monthly records considered here.

Data-driven criticality assessment provides a closer conceptual precedent. Gopalakrishnan, Subramaniam, and Skoogh integrate production and maintenance data to support prioritization and productivity-oriented decisions [9]. Their systems perspective motivates the present combination of production load and maintenance outcomes. In contrast, the artifact developed here is intentionally lightweight: it works with monthly historical records, produces scenario estimates rather than remaining-useful-life predictions, and exposes every transformation to the user.

B. Computerized Maintenance Management, Enterprise Asset Management, and Decision Support

Computerized maintenance management systems (CMMS) and enterprise asset management (EAM) systems provide asset registers, work-order histories, spare-parts data, and maintenance schedules. Research on smart CMMS workflows proposes integrating machine learning and multi-criteria decision methods so that stored data can support prioritization rather than passive reporting [10]. An IJACSA maintenance decision system based on digital twins and ontologies illustrates the more integrated end of this design space [11]. Industrial case studies also emphasize careful requirements specification before CMMS implementation [13]. The present artifact

addresses an earlier maturity stage with inspectable monthly calculations and no real-time digital-twin requirement.

Technology implementation remains socio-technical. Brundage et al. argue that organizations need guidance for selecting maintenance technologies according to existing data and capability [14]. Maintenance performance frameworks likewise require indicators that connect operational measures to business objectives [15]. Industrial managers also report changes in leadership, competence, and collaboration as maintenance digitalizes [16]. These findings support an incremental application that engineers can inspect and later integrate with a CMMS or manufacturing execution system, consistent with empirical evidence on data-driven maintenance coordination [17].

C. Spreadsheet Engineering and Research Gap

Spreadsheets offer rapid development and direct manipulation, but formula errors can propagate into decisions. Reviews of spreadsheet errors identify logic, omission, copy-and-paste, and input errors as persistent risks [18]. Auditing protocols therefore recommend structural mapping, formula inspection, input testing, and risk-based review [19]. A recent systematic review of spreadsheet quality assurance confirms the importance of preventive design, testing, anomaly detection, and user-oriented tools [20]. These controls are essential when a workbook is used as an operational application rather than a temporary analysis.

TABLE I. COMPARISON OF MAINTENANCE DIGITALIZATION APPROACHES

Technology approach	What it offers	Role in this study
Sensor-based predictive maintenance (PdM)	Learns failure patterns from condition data	Future extension; requires sensors and labelled events
Digital twin	Simulates asset behavior across the life cycle	Long-term architecture; requires system integration
Computerized maintenance management system/enterprise asset management (CMMS/EAM)	Stores assets, work orders, and maintenance history	Approved data source and destination
Spreadsheet decision support system (DSS)	Provides fast, inspectable calculations	Implemented platform; requires formula and change controls
Proposed low-code application	Uses an 80/20 comparison and three similar historical records	Transparent transition layer for monthly data

Taken together, these studies cover the two ends of the problem-static maintenance reporting and high-infrastructure predictive maintenance-but say less about the intermediate case. The smart CMMS workflow in [10] combines machine learning and multi-criteria decision methods, the digital-twin studies in [11] and [12] illustrate a more integrated design space, the CMMS case in [13] emphasizes requirements specification, and spreadsheet-quality studies [18]-[20] address formula reliability rather than maintenance-specific scenario logic. The present application addresses the narrower

intermediate case through explicit scenario weighting, cyclic seasonality, spreadsheet-quality controls, and a visible calculation path. Table I summarizes this position.

D. Interpretability and Progressive Digitalization

Interpretability has two meanings in this context. Model interpretability concerns whether a user can follow the relationship between inputs, historical evidence, and outputs. Process interpretability concerns whether the organization can explain who supplied the data, when they were updated, which formula version was used, and how an indicator influenced a decision. Explainable predictive-maintenance research mainly addresses the first meaning [7], whereas technology-implementation and socio-technical studies show that the second is equally important for adoption [14], [16]. IJACSA has also published dashboard-based decision support that combines explicit criteria and graphical drill-down [21]. The proposed artifact therefore treats trace fields, version information, and warnings as first-class outputs rather than administrative details.

Progressive digitalization also requires an explicit capability boundary. A low-code workbook is appropriate when data volume is moderate, update frequency is low, the user group is small, and the consequence of a delayed or unavailable result is manageable. It becomes inappropriate when calculations must run in real time, many users edit concurrently, decisions are safety-critical, or audit and access-control requirements exceed file-level governance. Defining this boundary prevents a successful pilot from becoming an ungoverned permanent system.

Finally, transparent models support learning. When a maintenance planner can see that two historical months dominate an estimate because their production levels are similar, disagreement can be expressed as a data or model issue: the months may have different product mix, a tooling change may be missing, or the production weight may be too high. A black-box score often hides that discussion. The present design intentionally favors a modest model that generates correct questions over a complex model whose numerical precision cannot yet be validated.

III. RESEARCH DESIGN AND DATA PROTECTION

A. Artifact-Oriented Case Study

The research follows an artifact-oriented industrial case-study design. The starting problem was a recurring need to compare maintenance outcomes across calendar months while accounting for planned production. The source material contained one year of monthly downtime, incident frequency, efficiency, realized production, and failure-type distributions for a generic asset group in an automotive component assembly process. The application was originally conceived in spreadsheet software because that environment was available to maintenance personnel and supported data validation, formulas, charts, and macro-controlled refresh actions.

The unit of analysis is the decision-support artifact, not the performance of a named plant or asset. The evaluation therefore tests whether the application transforms valid inputs into consistent, bounded, and traceable outputs.

It does not estimate economic savings, causal maintenance effects, failure probability, or remaining useful life. This distinction protects the scientific validity of the claims and makes the design transferable to similar plants with sparse historical data.

B. Confidentiality-Preserving Transformation

Industrial confidentiality was treated as a design constraint. The manuscript contains no company name, specific equipment identifier, product code, absolute production capacity, shift configuration, absolute downtime, maintenance cost, or screen capture from the operational workbook. Asset identifiers are replaced by generic labels and failure modes are consolidated into broad functional families. After the operational records were removed, the disclosed 0-100 values were constructed manually to retain only the broad qualitative ordering and seasonal variation needed for the demonstration. Original differences, ratios, means, variances, correlations, and distributions are not claimed to be preserved, and the disclosed values cannot be converted back into operational measurements.

This strategy permits methodological replication without exposing operational performance. A user can populate the same schema with local data and reproduce all calculations. Where governance requires stronger protection, the application can store only pseudonymous asset keys and obtain source data through a controlled export rather than embedding raw work orders.

C. Data Schema and Indicators

The minimal schema is deliberately small so that missingness and unit inconsistencies are visible. Each monthly record is uniquely identified by a pseudonymous asset group and calendar month. Table II separates raw inputs, derived indicators, and governance fields.

TABLE II. DATA FIELDS, MEANINGS, AND VALIDATION RULES

Data field	User enters?	Meaning	Validation or display rule
Asset group	Yes	Generic group analyzed	Approved dropdown list
Target month	Yes	Planning month	January–December; December is adjacent to January
Planned production index	Yes	Expected production level	Number from 0 to 100
Downtime risk index	No	Relative interruption level	Displayed from 0 to 100
Incident risk index	No	Relative incident frequency	Displayed from 0 to 100
Efficiency index	No	Relative operating performance	Displayed from 0 to 100
Failure-family share	No	Distribution by general failure category	All shares total 100%
Data-quality status	No	Valid, missing, or stale	Warning required before output

Downtime and incident values are decision indices, not calibrated probabilities. Efficiency is reported separately so that different maintenance signals are not hidden inside one score. Failure-family shares are ranked from largest to smallest, while

the data-quality status prevents the interface from presenting stale or incomplete records as normal estimates.

D. Preprocessing and Data-Quality Rules

Preprocessing begins with a data contract. The asset key and month form a unique record key; production, downtime, incidents, and efficiency have declared units; failure families use a controlled vocabulary; and every record carries an update date and approval state. Duplicate keys are rejected rather than silently aggregated. Missing production or month prevents scenario calculation because both variables are required by the similarity model. A missing failure-family distribution can be tolerated only if the dashboard marks the profile as unavailable and does not reuse a previous chart.

Numeric validation is performed before the comparison begins. Negative production, downtime, or incident values are invalid. Efficiency must lie within the organization-defined percentage range. Failure-family shares must be nonnegative and sum to one hundred within a rounding tolerance. The history must also contain more than one production level; otherwise, planned production cannot help distinguish between months. In that case, the interface displays a warning rather than attempting a misleading comparison.

The parameters used to compare scenarios are stored with the model version. Because changing the historical records can change which months are considered most similar, an archived decision should retain the input values, data snapshot identifier, eighty-twenty priority, selected records, and output. The workbook can implement this through a protected append-only log or an exported decision record. When operating data are refreshed, regression test cases must be rerun before the new version is released.

The synthetic dataset follows the same contract. Its values were created after confidential records were removed and were manually checked for allowed ranges, unique month keys, and complete generalized failure shares. Because only qualitative ordering was preserved, the dataset is suitable for interface and calculation verification but not for estimating plant-level effect sizes.

IV. APPLICATION ARCHITECTURE AND MODEL

A. Functional Requirements

Requirements were derived from the industrial workflow and spreadsheet-risk literature. The application must allow an authorized user to select an asset group and target month, enter a planned production level, obtain indicators with one refresh action, inspect the dominant failure families, and trace each output to the records and weights that produced it. Invalid or incomplete inputs must stop calculation. Model cells must be separated from editable cells, and chart refresh must not change calculation logic.

- Usability: three controlled scenario inputs and a concise dashboard.
- Functional suitability: monthly lookup, weighted scenario estimation, indicator generation, and failure-family ranking.

- Reliability: explicit bounds, missing-data flags, and deterministic formulas.
- Maintainability: named ranges, separated worksheets, documented weights, and a version log.
- Security and confidentiality: pseudonymous identifiers, protected formula sheets, and no confidential screenshots in shared reports.

B. Layered Architecture

The workbook is organized into three logical worksheets. Input_Dashboard contains editable scenario fields and charts. History contains the monthly table and a data dictionary. Model_Audit contains comparison values, relevance scores, selected records, calculation checks, version information, and data-quality flags. This separation reduces accidental edits and makes a calculation trace visible without exposing implementation details on the user screen.

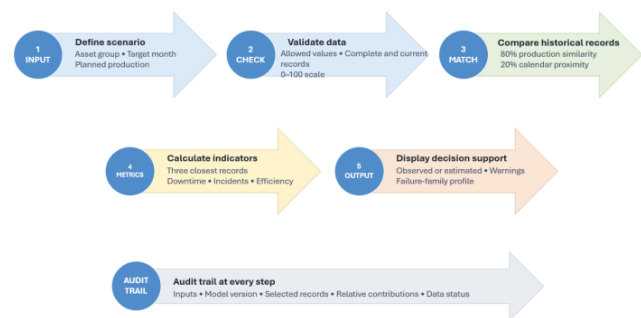


Fig. 1. Five-step information flow. Each estimate retains its inputs, selected historical records, relative contributions, and data-quality status.

As shown in Fig. 1, the application follows five visible steps: scenario entry, data checks, transparent comparison, indicator calculation, and dashboard presentation. Audit information accompanies the result and records the input, model version, three selected historical records, their relative contributions, and data-quality status.

C. Interpretable Scenario Assessment

The application uses a transparent comparison procedure rather than a black-box prediction model. For each historical month, it records the absolute difference between planned and historical production on the common 0-100 scale. It also records circular calendar distance, for which a one-month gap equals one sixth of the maximum six-month separation, and December remains adjacent to January. The two distances are combined using the selected production-calendar priority; the lowest combined value is the closest historical situation.

Decision priority: 80% production similarity + 20% calendar proximity

Production receives the larger share because workload is the main planning factor in the industrial case. Calendar proximity provides context for recurring seasonal conditions. The month comparison is circular, so December and January are treated as neighboring periods.

After comparing all historical records, the application keeps the three closest situations. Production distance is the absolute production difference divided by the full 0-100 scale. Calendar

distance is the shortest circular month gap divided by the maximum six-month separation. The two values are combined using the stated 80% production and 20% calendar priority. Records are ranked from the smallest combined distance, and smaller distances receive larger contributions by using their reciprocal after adding a numerical safeguard of 0.0001. The three contributions are then rescaled to total 100%. Three records are retained as a readability-oriented design choice rather than an optimized neighborhood size. Future validation on multi-year operational data will compare alternative record counts using out-of-sample error and result stability.

Downtime, incident, efficiency, and failure-family indicators are weighted averages of the selected records. The displayed values remain on a normalized 0-100 scale. When planned production is outside the supported historical range or no sufficiently similar case exists, the application displays a warning instead of presenting the result as certain. The pre-specified values—80% production, 20% calendar, and three records—encode industrial judgment and are configuration parameters, not fitted constants.

For transparency, the dashboard shows whether a value is observed or estimated, and the audit sheet records the historical months that contributed to the result. A maintenance planner can therefore understand the basis of an estimate without interpreting advanced mathematical notation.

This procedure is deliberately simple and auditable. It is intended for scenario assessment and maintenance planning, not for predicting the exact probability or date of a failure.

A separate exact-lookup mode retrieves descriptive statistics for a selected historical month. This dual behavior lets users distinguish recorded values from scenario estimates and prevents an estimated result from being presented as an observed fact.

D. Observed and Estimated Operating Modes

The interface uses two visually distinct operating modes. Observed mode is selected when the user requests a month that exists in the history without changing its recorded production level. The application retrieves the stored indicators and labels them as historical observations. No smoothing or load correction is applied. This mode is useful for reviewing past performance, checking source data, and discussing a completed period.

Estimated mode is selected when planned production differs from the stored level or when the user explicitly requests scenario assessment. The dashboard then labels all values as estimates, displays the contributing records, and shows how production and calendar context influenced the comparison. If no sufficiently similar historical situation exists, the application does not hide the weak match; it displays a low-similarity warning. This avoids presenting extrapolation as a well-supported estimate.

The two modes also have different audit semantics. An observed value must agree exactly with the history table and can be reconciled to a source report. An estimated value must agree with the documented step-by-step procedure and can be reconciled to the selected records and their relative

contribution. Mixing these paths in a single unlabeled chart would undermine both traceability and user trust. The mode label is therefore included in every screen export.

TABLE III. ALGORITHM 1. INTERPRETABLE SCENARIO EVALUATION

Step	User-readable action
1	Confirm the asset group, target month, and planned production.
2	Compare planned production with every historical month.
3	Compare months on a circular calendar: December is next to January.
4	Apply the 80% production and 20% calendar priority; keep the three closest situations.
5	Combine their downtime, incident, efficiency, and failure-family indicators.
6	Display 0–100 results, source records, relative contributions, and warnings.

E. Spreadsheet Implementation

Data-validation lists restrict asset and month selection. Planned production accepts only numeric values within an organization-defined operating range. Named ranges replace direct sheet coordinates so that spreadsheet logic remains readable after table expansion. Exact monthly retrieval can be implemented with INDEX and MATCH or an equivalent structured lookup. For scenario assessment, the workbook ranks the historical rows, retains the closest situations, and combines their indicator values through visible worksheet calculations. The macro is limited to validation, recalculation, chart refresh, and navigation; it does not contain hidden business logic.

The interface mock-up in Fig. 2 reproduces the interaction pattern without copying any operational screen. For the worked example, the user selects a generic asset group, June, and a planned production index of 85. For May, June, and April, the normalized production distances are 0.0500, 0.1300, and 0.0800; the circular calendar distances are 0.1667, 0, and 0.3333; and the combined 80/20 distances are 0.0733, 0.1040, and 0.1307. Converting the smaller distances into larger contributions and rescaling them to 100% gives 44.1%, 31.1%, and 24.8%. The resulting downtime, incident, and efficiency indices are 41.8, 35.0, and 82.1, displayed as 42, 35, and 82. The audit sheet exposes each step of this calculation (Table III)(Fig. 3).

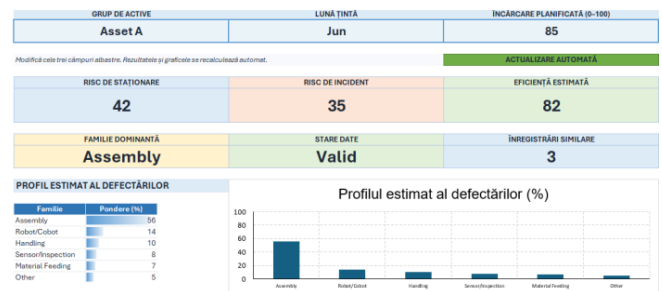


Fig. 2. Implemented Excel dashboard for the June scenario.

Record ID	Asset Group	Month	Month No	Historical Lead	Eligible	Production Distance	Calendar Distance	Scenario Score	Rank	Selected
1	Asset A	Jan	1	79	1	0.090	0.833	0.215	7	0
2	Asset A	Feb	2	81	1	0.040	0.967	0.165	5	0
3	Asset A	Mar	3	81	1	0.040	0.500	0.132	4	0
4	Asset A	Apr	4	77	1	0.080	0.333	0.131	3	1
5	Asset A	May	5	80	1	0.050	0.167	0.073	1	1
6	Asset A	Jun	6	72	1	0.130	0.000	0.104	2	1
7	Asset A	Jul	7	64	1	0.210	0.167	0.201	6	0
8	Asset A	Aug	8	56	1	0.270	0.333	0.263	8	0
9	Asset A	Sep	9	54	1	0.310	0.500	0.348	12	0

Fig. 3. Calculation data.

To reduce spreadsheet risk, editable cells use a distinct fill, formula cells are locked, and workbook structure is protected. Protection is an error-prevention mechanism rather than a security boundary. Files containing operational data must still be governed through access control, retention rules, backups, and approved distribution channels.

F. Auditability and Change Control

Calculation design follows a small number of reusable patterns. Each derived column uses one consistent calculation copied through a structured table. The eighty-twenty priority, number of reference records, accepted input ranges, and output limits are stored once in a parameter block with units and rationale. They are not repeated inside dashboard cells. This arrangement allows an auditor to verify that the same rules are applied throughout the workbook.

The audit worksheet exposes five calculation stages: validated inputs, production and month comparisons, ranked historical records, selected-reference flags, and final indicator calculations. Conditional checks confirm that the required number of references was selected, failure shares sum to one hundred, and the displayed dashboard values agree with the model outputs. A release status remains false if any check fails. The purpose is not to prove the absence of every spreadsheet defect, but to make high-impact defects observable before a decision is made.

Macro code is kept thin because hidden procedural logic is difficult for non-developers to inspect. The refresh routine reads validated inputs, requests a full recalculation, checks the release-status cell, refreshes chart series, and records a timestamp. It does not overwrite history, calculate indicators, or suppress errors. If macros are disabled, formulas remain visible, and the user can calculate manually, while the interface displays that automated refresh is unavailable.

Changes follow a lightweight release process. A maintenance-domain owner approves indicator definitions and failure families; a model owner reviews formulas and tests; an information owner approves data access and retention; and a designated user accepts the interface. The version log records the reason for change, affected ranges, reviewer, approval date, and test outcome. These controls address the principal spreadsheet risks identified in [18]-[20] without requiring a full software-development platform.

V. VERIFICATION METHOD

A. Test Strategy

Verification follows three levels. First, structural tests confirm that required worksheets, named ranges, validation lists, calculations, and chart sources exist. Second, numerical tests compare workbook outputs with an independent implementation of the documented calculation procedure. Third, behavioral tests exercise boundaries, invalid inputs, and changes in planned production. Software-quality criteria are aligned with functional suitability, reliability, usability, security, and maintainability from the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 25010:2023 standard [22]. Asset-management vocabulary and intent follow the

International Organization for Standardization (ISO) 55000:2024 standard [23].

The verification dataset contains twelve synthetic monthly records for one generic asset group and six generalized failure families. For each month, scenarios are evaluated at ninety, one hundred, and one hundred ten percent of the reference production index, yielding thirty-six scenario calculations. Nine negative tests cover missing asset, invalid month, nonnumeric load, out-of-range load, missing indicator, stale record, failure shares not summing to one hundred, unavailable macro, and an attempted edit of a protected calculation cell (Table IV).

TABLE IV. VERIFICATION PLAN AND ACCEPTANCE CRITERIA

Verification check	Cases	What passing means
Historical replay	12	The dashboard exactly reproduces each stored month
Scenario output bounds	36	Every indicator remains between 0 and 100
Planned-production sensitivity	48 comparisons	Changing planned production changes a valid output
Invalid-input handling	9 rules	Calculation stops, and a clear warning appears
Traceability	36 scenarios	Selected records and relative contributions are visible
Confidentiality review	Entire manuscript	No company, asset identifier, product code, or raw operational value

B. Comparative Verification and Threat Controls

Comparative verification uses leave-one-month-out testing on the twelve-month synthetic sequence. One month is removed, its production level and calendar position are retained as scenario inputs, and its downtime, incident, and efficiency indices are estimated from the remaining eleven months. Three transparent baselines are used: the historical record with the nearest production level, the average of the two adjacent calendar months, and the mean of all remaining months. Performance is summarized with mean absolute error (MAE), measured in normalized index points. The same protocol is repeated with production-calendar priorities of 70/30, 80/20, and 90/10 and with two, three, or four selected records.

These tests answer whether the estimator behaves consistently and whether the pre-specified setting is competitive on the disclosed synthetic example. Because the twelve training folds overlap and come from one temporally related annual sequence, their MAEs are descriptive comparisons rather than independent estimates of generalization error. They cannot establish plant-level predictive accuracy. Historical replay verifies implementation correctness only, while parameter sensitivity verifies responsiveness rather than a causal relationship between production and failure. No maintenance-user experiment or operational impact study was conducted; these evaluations require future work with multi-year operational data.

C. Independent Oracle and Test-Case Design

Numerical verification uses an independent reference implementation rather than a duplicate set of workbook formulas. The reference implementation receives the same synthetic table and scenario inputs, repeats the documented

comparison steps in a separate execution environment, and returns the selected historical records, their relative contributions, and the resulting indicators. Workbook and reference outputs are compared at a precision appropriate to displayed values. This approach detects incorrect references, stale charts, rounding errors, and copy defects more effectively than visual inspection alone.

Test cases are partitioned by purpose. Historical replay uses the exact historical record and its recorded production level so that exact agreement is expected. Scenario cases use three reference records and three planned-load levels for every month. Boundary cases evaluate minimum, maximum, and equal production values. Negative cases remove or corrupt one required element at a time. Traceability cases compare the displayed reference list and relative contributions with the independent implementation, not only the final indicator. A failure in an intermediate field is therefore visible even when compensating errors produce a plausible total.

Acceptance criteria were defined before execution. Numerical tests require exact agreement for integer indices and a small rounding tolerance for combined results. Behavioral tests require finite outputs inside configured bounds and a changed response when planned load changes. Global monotonicity is not required because changing production can change the selected historical references; forcing monotonicity would conceal the model's actual behavior. Instead, large discontinuities are flagged for model-owner review.

VI. RESULTS

A. Functional and Numerical Verification

The independent implementation confirmed exact historical replay for 12 of 12 months when observed mode was selected. All 36 of 36 three-level scenarios returned downtime, incident, and efficiency indices within the defined 0-100 bounds. Planned-load changes produced a finite, bounded, non-invariant risk response in 48 of 48 comparisons. All nine invalid-input cases blocked calculation rather than returning a plausible-looking value.

These results establish deterministic consistency between the documented procedure and the artifact (Table V). They do not demonstrate predictive accuracy on unseen failures. The main practical result is that a maintenance user can reconstruct why a value appeared: the selected historical records, production and month comparisons, relative contributions, and final limits are all visible.

TABLE V. VERIFICATION RESULTS

Verification result	Outcome	Practical meaning
Historical replay	12/12 passed	Stored monthly values were reproduced correctly
Scenario bounds	36/36 passed	No result fell outside 0–100
Planned-production sensitivity	48/48 passed	The planned-production input actively influences output
Invalid-input protection	9/9 rules implemented	Invalid data cannot produce a plausible-looking result
Traceability	Complete	Every estimate shows its source records and contributions
Confidentiality	Passed	No operational identifier or raw value was disclosed

B. Sensitivity and Boundary Behavior

All forty-eight planned-load comparisons produced a finite, bounded, non-invariant response. Leave-one-month-out testing added a more demanding comparison. The proposed 80/20 configuration with three historical records achieved mean absolute errors of 9.13 for downtime, 4.34 for incidents, and 3.09 for efficiency, giving an overall mean of 5.52 index points. This was better than production-only nearest-neighbor (11.56) and the historical mean (8.06), but worse than the adjacent-calendar baseline (4.28). Fig. 4 reports all three indicators separately and prevents the overall mean from hiding where errors occur.

The baseline result shows that the disclosed synthetic sequence has a strong local calendar pattern and that the industrial 80/20 priority is a pre-specified design choice rather than an accuracy-optimal default. Boundary testing also distinguishes interpolation inside the observed production range from extrapolation outside it. The interface displays an extrapolation warning and the distance from the nearest supported load. Output limits protect the screen from impossible indices but do not make a weak scenario reliable.

Traceability tests confirmed that the dashboard, audit table, and independent implementation used the same historical references and relative contributions for every scenario. This property is central to the contribution: a reviewer can reproduce an output without inspecting macro code or accepting a black-box claim. When a result appears counterintuitive, the contributing evidence is immediately available for domain review.

C. Visibility of Seasonal and Failure Patterns

The synthetic monthly demonstration in Fig. 4 shows the exact generic sequence used for the worked example and comparative tests. Production, downtime, and incident indices share the same normalized 0-100 scale, and June is highlighted as the example target. The pattern is not evidence of causality: calendar position can represent shutdowns, staffing, product mix, environmental conditions, or maintenance campaigns.

Failure-family aggregation remains visible in the dashboard in Fig. 2. The comparative chart in Fig. 4 addresses a different question: whether the proposed estimator adds value over simple alternatives. The result is mixed rather than uniformly favorable, which supports a verification claim and motivates multi-year industrial validation before operational use.

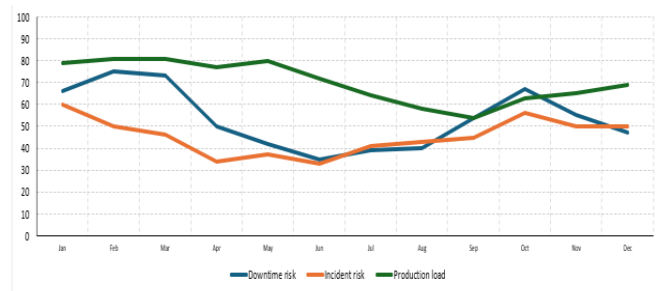


Fig. 4. Monthly evolution of the selected group.

D. Parameter Sensitivity

Table VI isolates the two configuration choices. Holding three records fixed, changing the production-calendar priority from 70/30 to 90/10 changed the overall mean absolute error from 5.45 to 6.81. Holding the 80/20 priority fixed, changing the record count from two to four changed it from 4.69 to 6.08. The mean absolute change in individual outputs relative to the pre-specified setting was 0.45 points for 70/30 with three records, 2.64 for two records, 1.57 for four records, and 2.09 for 90/10 with three records. These movements are moderate on a 0-100 scale, but the better result for two records confirms that three records are a transparency-oriented design choice, not an optimized choice. Objective selection of the record count is reserved for future validation on multi-year operational data.

TABLE VI. PARAMETER SENSITIVITY IN LEAVE-ONE-MONTH-OUT TESTING

Production/c alendar priority	Reco rds used	Downtime MAE	Incide nt MAE	Efficie ncy MAE	Overall MAE
70% / 30%	3	8.97	4.32	3.07	5.45
80% / 20%	2	7.46	4.20	2.42	4.69
80% / 20% (pre- specified)	3	9.13	4.34	3.09	5.52
80% / 20%	4	9.96	4.75	3.52	6.08
90% / 10%	3	10.78	5.88	3.77	6.81

VII. DISCUSSION

A. Answers to the Research Questions and Contribution

1) *Answer to research question 1 (RQ1).* The application converts three controlled inputs—generic asset group, target month, and planned production—into transparent maintenance indicators through a visible five-step workflow. Production and calendar comparisons, selected historical situations, relative contributions, outputs, and warnings remain inspectable.

2) *Answer to research question 2 (RQ2).* The implementation reproduced 12 of 12 historical observations, kept 36 of 36 scenario outputs inside the 0-100 bounds, produced a valid response in 48 of 48 planned-load comparisons, implemented all nine invalid-input rules, and preserved record-level traceability. These tests establish calculation consistency rather than predictive validity.

3) *Answer to research question 3 (RQ3).* The proposed setting outperformed production-only and historical-mean baselines but did not outperform the adjacent-calendar baseline. Its overall mean absolute error ranged from 4.24 to 6.88 across the full tested weight-neighbor grid; the isolated settings in Table VI show that the pre-specified setting is reasonably stable but not empirically optimal on one synthetic year.

4) *Answer to research question 4 (RQ4).* For all tested scenarios, the inputs, selected records, relative contributions, and final indicators remained visible and agreed with the independent calculation. This establishes structural traceability and reconstructability for the tested outputs. User

understanding and operational suitability require future empirical study.

The contribution is not algorithmic complexity. It is the disciplined design of a transparent transition artifact. First, the model formalizes an industrial intuition—production matters more than calendar month—through visible comparisons and an eighty-twenty priority. Second, the circular treatment of the calendar avoids a common defect at the year boundary. Third, the architecture separates data, calculations, interface, and audit evidence, responding directly to spreadsheet-quality risks. Fourth, the privacy protocol demonstrates how an industrial case can be communicated without exposing performance values or asset identifiers.

The work complements predictive-maintenance literature by defining a lower-infrastructure maturity level. Sensor-based learning remains preferable when sufficient data, labels, and integration capability exist. Before that point, a low-code application can improve data definitions, reveal missing values, establish indicator ownership, and create user expectations for traceability. These capabilities reduce the organizational distance to a future CMMS, manufacturing execution, or predictive platform.

B. Practical Implications

The artifact can be piloted with limited software procurement and short training, but deployment should be conditional on local verification. Maintenance engineers can inspect the calculations, while managers receive a consistent monthly view. The baseline comparison also prevents users from assuming that additional weighting automatically improves accuracy: in a strongly seasonal process, a simple calendar rule may be the more defensible starting point.

A recommended deployment sequence is: define the data dictionary; clean at least one annual cycle; validate the workbook against independent calculations; pilot with one pseudonymous asset group; establish data and model owners; review the eighty-twenty weights; and only then extend the schema. If the application begins to influence safety-critical or high-cost decisions, migration to a governed database and tested application layer becomes necessary.

C. Governance and Migration Roadmap

Governance should scale with decision consequence. During an analytical prototype, peer review and protected formulas may be sufficient. During a controlled pilot, the organization needs approved data extracts, named owners, a versioned test pack, backup, and a restricted distribution list. Operational use requires scheduled data refresh, release approval, incident handling, training, and periodic validation. When several assets or users are added, a shared database and application service become more reliable than duplicated workbooks (Table VII).

Migration does not require discarding the artifact. The workbook can serve as an executable specification for a later system: named fields become database attributes, calculation rules become tested service functions, validation lists become reference tables, and audit fields become log events. The dashboard layout also provides a documented starting point for

a web interface. Preserving these semantics reduces the risk that a technically advanced replacement loses the interpretability expected by maintenance users.

Integration with computerized maintenance management systems (CMMS) and manufacturing execution systems (MES) should be staged. A read-only import is safer than immediate bidirectional synchronization. After identifiers and units are reconciled, the application can consume approved history extracts and export a scenario record or maintenance-planning note. Automatic work-order creation should be considered only after the decision rule, uncertainty, responsibilities, and reversal process are formally validated. Table VI summarizes the recommended maturity path.

TABLE VII. GOVERNANCE AND MIGRATION ROADMAP

Maturity stage	Required control	Evidence and readiness to progress
Prototype	Peer review of data fields and calculations	Calculation map and sample tests; data structure is stable
Controlled pilot	Named data, model, and user owners	Versioned tests and user feedback; workflow is understood
Operational workbook	Access control, release approval, and backup	Audit log and approved versions; use is repeatable and governed
Integrated application	Shared database and automated tests	Interface and security logs; several users and assets are supported
Predictive-maintenance extension	Monitored and externally validated model	Validation and model-change reports; measurable decision benefit

D. Limitations and Validity

The study has six principal limitations. The historical granularity is monthly, so short-term precursors and event sequences are unavailable. One annual cycle is insufficient to estimate stable seasonality. The pre-specified 80/20 priority and three-record setting originate from industrial judgment rather than optimization on an external dataset. Synthetic indices permit disclosure but prevent independent assessment of plant performance and may favor the calendar baseline by construction. The leave-one-month-out months are temporally related and therefore are not independent observations. Finally, no maintenance-user study was conducted, so the study does not claim improved decision time, interpretation accuracy, adoption, maintenance effectiveness, or economic benefit.

The model is also vulnerable to concept drift. Changes in product mix, automation, maintenance policy, staffing, or equipment modification can make historical similarity misleading. Data-quality flags and periodic weight review are therefore mandatory. Macros introduce compatibility and security considerations; organizations should sign approved code, disable untrusted macros, and keep business logic in visible formulas. Spreadsheet protection cannot substitute for authentication, authorization, logging, or centralized backups.

E. Future Development

Future work should validate the model on several years, asset groups, and industrial sites; compare it with multi-year seasonal-naïve, regression, tree-based, and probabilistic baselines; and report uncertainty intervals. A preregistered user

study should measure decision time, interpretation accuracy, perceived trust, workload, and disagreement resolution. A governed implementation can move the history to a database, connect to CMMS and MES platforms, and expose the same transparent calculations through a web interface. When condition-monitoring data become available, the scenario layer can remain as a planning view while sensor models provide short-horizon alerts.

VIII. CONCLUSION

This study presented the design and verification of an interpretable low-code application for maintenance scenario assessment in automotive assembly. The application converts controlled inputs and sparse monthly history into traceable downtime, incident, efficiency, and failure-family indicators. Its configurable comparison uses planned production and cyclic month proximity, while the selected historical records and relative contributions remain visible. Spreadsheet-quality controls, fail-closed validation, and separation of input, model, and audit layers reduce the risks of end-user computing.

The privacy-preserving case study excluded the company, specific equipment identifiers, operational capacities, absolute times, and confidential screens. Verification established calculation agreement, bounded scenario behavior, input rejection, record-level traceability, and sensitivity to parameter choices. Comparative testing showed that the proposed setting was better than production-only and historical-mean baselines but not the adjacent-calendar baseline. The artifact is therefore supported as a transparent planning aid and transition specification—not as a validated failure-probability, remaining-life, or optimization model.

DATA AVAILABILITY

The raw industrial dataset and operational workbook are not publicly available because they contain confidential manufacturing information. All values and interface elements shown in this study are synthetic, generalized, and non-identifying. The disclosed twelve-month sequence, the step-by-step worked example, baseline definitions, test counts, and acceptance criteria support reproduction of the reported verification calculations. Additional non-confidential implementation details may be provided on reasonable request, subject to institutional and publication requirements.

DECLARATION ON GENERATIVE ARTIFICIAL INTELLIGENCE

Generative artificial intelligence tools were used to assist with drafting, language refinement, calculation scripting, document organization, and visualization layout. Scientific responsibility remains with the authors, who retain full responsibility for independently verifying all domain information, calculations, results, references, interpretations, originality, and integrity of the final manuscript.

REFERENCES

- [1] A. Kusiak, "Smart manufacturing," *Int. J. Prod. Res.*, vol. 56, nos. 1-2, pp. 508-517, 2018, doi: 10.1080/00207543.2017.1351644.
- [2] J. Bokrantz, A. Skoogh, C. Berlin, T. Wuest, and J. Stahre, "Smart Maintenance: A research agenda for industrial maintenance management," *Int. J. Prod. Econ.*, vol. 224, Art. no. 107547, 2020, doi: 10.1016/j.ijpe.2019.107547.

- [3] F. Psarommatas, G. May, and V. Azamfirei, "Envisioning maintenance 5.0: Insights from a systematic literature review of Industry 4.0 and a proposed framework," *J. Manuf. Syst.*, vol. 68, pp. 376-399, 2023, doi: 10.1016/j.jmsy.2023.04.009.
- [4] T. Zonta et al., "Predictive maintenance in the Industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, Art. no. 106889, 2020, doi: 10.1016/j.cie.2020.106889.
- [5] P. Nunes, J. Santos, and E. Rocha, "Challenges in predictive maintenance - A review," *CIRP J. Manuf. Sci. Technol.*, vol. 40, pp. 53-67, 2023, doi: 10.1016/j.cirpj.2022.11.004.
- [6] A. Theissler, J. Perez-Velazquez, M. Kettelgerdes, and G. Elger, "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry," *Reliab. Eng. Syst. Saf.*, vol. 215, Art. no. 107864, 2021, doi: 10.1016/j.res.2021.107864.
- [7] L. Cummins et al., "Explainable predictive maintenance: A survey of current methods, challenges and opportunities," *IEEE Access*, vol. 12, pp. 57574-57602, 2024, doi: 10.1109/ACCESS.2024.3391130.
- [8] S. Suwatcharachaitiwong et al., "Machine Learning-Driven Preventive Maintenance for Fibreboard Production in Industry 4.0," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 3, pp. 942-950, 2025, doi: 10.14569/IJACSA.2025.0160392.
- [9] M. Gopalakrishnan, M. Subramaniyan, and A. Skoogh, "Data-driven machine criticality assessment - Maintenance decision support for increased productivity," *Prod. Plan. Control*, vol. 33, no. 1, pp. 1-19, 2022, doi: 10.1080/09537287.2020.1817601.
- [10] E. G. Gorski, E. F. R. Loures, E. A. P. Santos, R. E. Kondo, and G. R. D. N. Martins, "Towards a smart workflow in CMMS/EAM systems: An approach based on ML and MCDM," *J. Ind. Inf. Integr.*, vol. 26, Art. no. 100278, 2022, doi: 10.1016/j.jii.2021.100278.
- [11] M. Abadi, C. Abadi, A. Abadi, and H. Ben-Azza, "A Smart Decision-Making System for the Optimization of Manufacturing Systems Maintenance using Digital Twins and Ontologies," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 8, 2022, doi: 10.14569/IJACSA.2022.0130811.
- [12] I. Errandonea, S. Beltran, and S. Arrizabalaga, "Digital twin for maintenance: A literature review," *Comput. Ind.*, vol. 123, Art. no. 103316, 2020, doi: 10.1016/j.compind.2020.103316.
- [13] I. Lopes et al., "Requirements specification of a computerized maintenance management system - A case study," *Procedia CIRP*, vol. 52, pp. 268-273, 2016, doi: 10.1016/j.procir.2016.07.047.
- [14] M. P. Brundage et al., "Where do we start? Guidance for technology implementation in maintenance management for manufacturing," *J. Manuf. Sci. Eng.*, vol. 141, no. 9, Art. no. 091005, 2019, doi: 10.1115/1.4044105.
- [15] P. Muchiri, L. Pintelon, L. Gelders, and H. Martin, "Development of maintenance function performance measurement framework and indicators," *Int. J. Prod. Econ.*, vol. 131, no. 1, pp. 295-302, 2011, doi: 10.1016/j.ijpe.2010.04.039.
- [16] C. Lundgren, C. Berlin, A. Skoogh, and A. Kallstrom, "How industrial maintenance managers perceive socio-technical changes in leadership in the Industry 4.0 context," *Int. J. Prod. Res.*, vol. 61, no. 15, pp. 5282-5301, 2023, doi: 10.1080/00207543.2022.2101031.
- [17] M. Hinrichs, L. Prifti, and S. Schneegass, "Data-driven decision-making in maintenance management and coordination throughout the asset life cycle: An empirical study," *J. Qual. Maint. Eng.*, vol. 30, no. 1, pp. 202-223, 2024, doi: 10.1108/JQME-04-2023-0038.
- [18] S. G. Powell, K. R. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," *Decis. Support Syst.*, vol. 46, no. 1, pp. 128-138, 2008, doi: 10.1016/j.dss.2008.06.001.
- [19] S. G. Powell, K. R. Baker, and B. Lawson, "An auditing protocol for spreadsheet models," *Inf. Manage.*, vol. 45, no. 5, pp. 312-320, 2008, doi: 10.1016/j.im.2008.03.004.
- [20] P.-L. Poon, M. F. Lau, Y. T. Yu, and S.-F. Tang, "Spreadsheet quality assurance: A literature review," *Front. Comput. Sci.*, vol. 18, no. 2, Art. no. 182203, 2024, doi: 10.1007/s11704-023-2384-6.
- [21] Okfalisa, Mahyarni, W. Anggraini, Saktioto, and B. Pranggono, "Assessing Digital Readiness of Small and Medium Enterprises: Intelligent Dashboard Decision Support System," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 98-108, 2022, doi: 10.14569/IJACSA.2022.0130412.
- [22] ISO/IEC 25010:2023, *Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Product Quality Model*, International Organization for Standardization, Geneva, Switzerland, 2023.
- [23] ISO 55000:2024, *Asset Management - Vocabulary, Overview and Principles*, International Organization for Standardization, Geneva, Switzerland, 2024.