



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

POWAM: A Policy-Compliant Low-Code Framework for Idempotent Purchase Order Workflow Automation and Management – An Industrial Case Study in Automotive Manufacturing

Arvind A R¹, Ajay Krishnan R², Praveen P V³, Santhosh R D⁴

^{1,2}Ashok Leyland Ltd., Chennai, Tamil Nadu, India

^{3,4}Department of Computing, Artificial Intelligence and Machine Learning, Coimbatore Institute of Technology, Coimbatore 641014, Tamil Nadu, India

Abstract— Purchase order (PO) processing in large automotive manufacturers is well served by enterprise resource planning (ERP) systems at the transactional level, yet the documentary trail around each order (drawings, quotations, inspection records, delivery challans and invoices) is still commonly organised by hand. This paper reports the design, implementation and eight-week production evaluation of POWAM, a PO workflow automation and management system built entirely inside an organisation's existing Microsoft 365 tenant so that it complies with prevailing cyber-security and IT policies. Following a design-science case-study method, we combine a Microsoft Power Apps client, SharePoint Online as a server-relative document repository, an Excel Online ledger and two Power Automate cloud flows: an event-driven engine that provisions PO folders idempotently after a metadata existence check, and a scheduled engine that renders the ledger into a styled HTML executive report through inline CSS injection. We formalise the path model, ledger schema and idempotency property, and document three integration failures (flow namespace binding, dynamic-content schema drift and connector table-recognition limits) together with their resolutions. Compared with the manual baseline, mean folder provisioning time fell from 185.0 s to 3.2 s (98.27%), duplicate directories fell from 6.8% to zero, naming uniformity rose from 78.4% to 100%, and about 45 min of weekly reporting effort was removed, with 99.8% flow execution success. The study is limited to one organisation and a non-randomised before–after comparison; we discuss these threats and outline how the uniformly structured repository prepares the ground for machine-learning-based document intelligence.

Keywords— Automotive manufacturing, Business process automation, Citizen development, Idempotent workflow, IT governance, Low-code development platforms, Microsoft Power Platform, Purchase order automation.

I. INTRODUCTION

Manufacturing has spent the last decade moving from machine-dominant to data-driven operations, a shift usually summarised as Industry 4.0 [1], [2].

For information-systems engineering this transition is less about robots on the shop floor and more about how reliably information moves between people, departments and systems [3]. Procurement sits at the centre of that flow. It connects design, assembly, suppliers and regulators, and is therefore one of the functions where digitalisation is expected to yield the largest operational gains [4], [5].

In practice the gains are uneven. ERP platforms capture the transactional backbone of a purchase order (approval status, vendor code and line-item value) with strong governance. The evidence that accompanies each order, however, including technical drawings, quotations, quality-inspection records, delivery challans, goods-receipt notes and tax invoices, is typically stored on shared drives organised by informal convention. Because this documentary periphery lies outside the ERP data model, it is largely invisible to central IT, and it tends to be maintained through spreadsheets and email by business users themselves, a pattern the literature describes as shadow IT [6], [7]. The consequences are familiar to anyone who has worked in a purchasing office: folder names drift, parallel operators create duplicate directories, tracking sheets diverge from reality, and weekly status reporting depends on one person's manual effort.

Two families of technology are commonly proposed for such problems. Robotic process automation (RPA) mimics human interaction at the user-interface layer and is attractive because it leaves existing systems untouched [8], [9], but it is known to be fragile and many RPA projects fail to deliver [10]. Low-code development platforms (LCDPs), on the other hand, let domain experts compose applications and workflows from visual building blocks on a managed cloud runtime [11]–[13], enabling so-called citizen development [14]. In regulated enterprises, both approaches meet a further constraint that is rarely discussed in the literature: the solution must survive cyber-security review.

New servers, unvetted third-party connectors, external APIs and data leaving the corporate tenant are usually not permitted.

This paper reports how we addressed these issues in the purchasing division of a large Indian commercial-vehicle manufacturer. As engineering students specialising in artificial intelligence and machine learning (AI/ML), we were initially drawn to the idea of applying document-understanding models to purchase records. We quickly realised that such models [15], [16] presuppose a clean, consistently organised repository, which did not exist. Automating the structure of the repository therefore became the necessary first step, and is the subject of this study. The research objectives (RO) were:

RO1: to design an architecture that automates PO folder provisioning, tracking and reporting using only components already approved within the organisation's Microsoft 365 tenant;

RO2: to guarantee repository integrity by making folder provisioning idempotent;

RO3: to document the integration failure modes encountered in production and how they were resolved; and

RO4: to quantify the effect of the system against the manual baseline.

Correspondingly, the paper contributes (i) a five-tier, policy-compliant reference architecture for document-centric procurement workflows; (ii) a formal path and ledger model with an explicit idempotency property; (iii) a catalogue of three low-code integration defects with root causes and resolutions; (iv) an inline-CSS reporting pipeline that needs no external rendering service; and (v) an empirical before–after evaluation from an eight-week production trial. Section II reviews related work, Section III describes the research method, Sections IV–VI present the requirements, architecture and implementation, Section VII discusses technical challenges, Section VIII reports results, Section IX discusses implications and limitations, and Section X concludes.

II. RELATED WORK

A. From Workflow Management to Robotic and Intelligent Process Automation

Business process management (BPM) provides the conceptual vocabulary for this work: processes are modelled, executed by process-aware information systems and continuously improved [17].

A long-standing concern in that tradition is flexibility, i.e., how a running process system copes with changes to its schema and data [18], which reappears in our study as connector schema drift (Section VII). RPA emerged as a lightweight alternative that automates already-digital but manual tasks by operating on the presentation layer [8]. Hofmann et al. characterise software robots as choreographies of technological modules acting inside existing IT ecosystems [9], and a systematic review by Ivančić et al. confirms that the academic treatment of RPA has lagged behind industrial adoption [19]. Herm et al. derived an implementation framework from 35 project reports precisely because a large share of RPA initiatives fail [10]. Leno et al. propose robotic process mining to discover automatable routines from user-interaction logs [20].

More recent work extends rule-based automation with machine learning. Chakraborti et al. describe the move from RPA to intelligent process automation (IPA) [21], and Engel et al. conceptualise cognitive automation as the probabilistic automation of knowledge work, complementary to deterministic workflow automation [22]. Adoption, however, depends on human factors such as trust and transparency [23]. At the organisational level, Kerpedzhiev et al. and Beverungen et al. argue that digitalisation forces BPM to balance exploitation and exploration, standardisation and flexibility [24], [25]. Our system deliberately sits on the deterministic side of this spectrum, while producing the data foundation that probabilistic automation would later need.

B. Low-Code Platforms and Citizen Development

Bock and Frank define low-code platforms as environments that combine visual modelling, pre-built components and a managed runtime to reduce hand coding [11]. Di Ruscio et al. compare low-code with model-driven engineering and conclude that the two share foundations but differ in target users and deployment model [12], a view echoed by the multi-vocal review of Bucaioni et al. [13]. Empirical evidence on practitioner pain points is growing: in an analysis of roughly 33,000 Stack Overflow posts, Alamin et al. found that application customisation, data storage and third-party integration dominate developer questions [26]. Roşis and Kirikova catalogue similar challenges, including scalability and vendor lock-in [27]. Waszkowski recently proposed the CD score to assess how far platforms actually enable non-IT specialists to build software [28]. Studies of citizen development highlight governance, skills and business–IT collaboration as decisive success factors [14], [29].

C. Governance, Shadow IT and Data Management

When business units build their own systems without IT involvement, the result is shadow IT, which carries security and integration risks [6]. Kopper et al. show, however, that business-managed IT is judged positively when it is aligned with the IT department and limited to local requirements [7]. Data governance frameworks emphasise clear ownership, quality rules and accountability for enterprise data [30]. These insights motivated our decision to treat cyber-security and IT-policy compliance as first-class design requirements rather than as a later approval step.

D. Integration and Cloud Execution Models

Our provisioning logic relies on the uniform interface and method semantics of RESTful services, in particular the safety of GET requests [31]. Cloud flows in Power Automate behave much like managed serverless workflows: event- or schedule-triggered, stateless between runs and billed per execution [32]. This execution model simplifies operations but makes correctness properties such as idempotency the designer's responsibility.

E. Document Intelligence and Machine Learning

Machine learning and deep learning are now standard tools for extracting structure from business data [15], [33]. For procurement documents, the most relevant line is information extraction from visually rich documents such as invoices and forms, which combines text, layout and image features [16]. Generative AI further enables summarisation and question answering over document collections [34]. All of these techniques assume that documents can be reliably located and associated with the correct business object, an assumption that manual folder management violates.

F. Research Gap

Table I summarises the reviewed streams. Existing studies explain what low-code and RPA can achieve and which organisational factors matter, but few report the concrete engineering decisions, failure modes and measured outcomes of a production low-code deployment in manufacturing procurement, and fewer still treat security-policy compliance as an explicit design constraint. This study addresses that gap.

TABLE I
SUMMARY OF RELATED RESEARCH STREAMS AND THE GAP
ADDRESSED BY THIS STUDY

Stream	Representative work	What it provides	What remains open
BPM and workflow flexibility	[17], [18], [24], [25]	Process concepts, schema evolution, capability models	Little guidance for cloud low-code connectors
RPA and IPA	[8]–[10], [19]–[23]	Definitions, implementation frameworks, adoption factors	UI-level automation is brittle; few API-level industrial cases
Low-code and citizen development	[11]–[14], [26]–[29]	Platform concepts, challenge catalogues, evaluation scores	Sparse measured outcomes in manufacturing procurement
Governance and shadow IT	[6], [7], [30]	Risk and alignment perspectives	Security policy rarely treated as a design input
Digital procurement and Industry 4.0	[1]–[5]	Strategic motivation and frameworks	Peripheral document workflows overlooked
Cloud integration and ML	[15], [16], [31]–[34]	REST semantics, serverless model, document AI	Assumes a clean, structured repository

III. RESEARCH METHODOLOGY

We followed a design-science-oriented single-case study, the same research logic used by Herm et al. for automation frameworks [10]. The six activities are shown in Fig. 1. The problem was identified by shadowing purchasing specialists and timing the manual procedure. Objectives were then translated into functional and non-functional requirements together with the organisation's IT-policy constraints. Design and development produced the architecture and the two automation engines. Demonstration took place through user-acceptance testing in which defects were logged and fed back into the design. Finally, the system was evaluated in an eight-week production trial (1 June–26 July 2026) involving two purchasing specialists and 200 purchase orders.

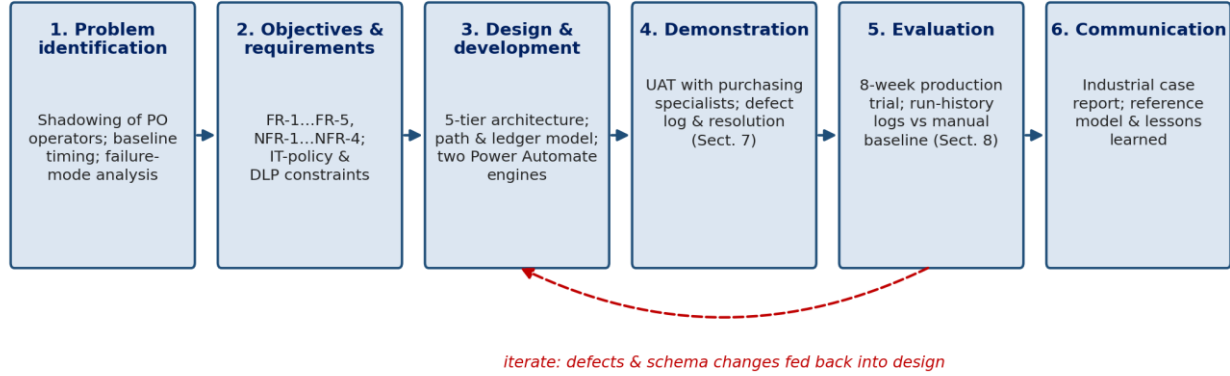


Fig. 1. Design-science research process followed in this study. The dashed arrow marks the feedback loop through which defects found during demonstration (Section VII) were returned to the design activity.

Five measurements were defined before the trial. Provisioning latency L for the manual baseline was timed by observation from receipt of the PO number to completion of the tracker entry; for the automated system it was taken from Power Automate run history (trigger start to completion of the ledger write). Naming uniformity U is the share of PO folders whose names conform to the convention $^{\wedge}\text{PO}_{-}[\text{A-Za-z0-9}]+\$. The duplicate rate D is the share of submissions that resulted in more than one folder for the same normalised identifier. Weekly reporting effort R was obtained from the analyst's time records, and execution success S is the share of flow runs completing without error. Relative improvement is computed as$

$$\Delta = (x_{\text{manual}} - x_{\text{auto}}) / x_{\text{manual}} \times 100\% \quad (1)$$

Results are reported as mean $\pm$ standard deviation. To express the magnitude of the latency change independently of sample size, we report Cohen's d with the pooled standard deviation, complemented by a Welch's t -test with a 95% confidence interval for the latency difference ($n = 200$ manual observations and $n = 200$ automated observations).

IV. PROBLEM FORMULATION AND REQUIREMENTS

A. Process Baseline

Before automation, an approved order was handled as follows. A purchasing specialist received the PO number from the ERP system, navigated to the fiscal-year and department sub-folder on the shared repository, created a new folder with sub-folders for specifications, quotations and invoices, opened a shared tracking spreadsheet and keyed in the PO number, operator name, timestamp and status. Every Friday an analyst compiled these entries into an email for management.

Observation revealed four recurring failure modes: (i) *directory drift*, i.e., typos, missing prefixes and folders nested under the wrong branch, which defeat programmatic search; (ii) *repetitive overhead* of roughly three to five minutes per order; (iii) *data-virtualisation bottlenecks*, because an earlier attempt to embed the large tracker in a low-code screen hit delegation limits and write locks; and (iv) *reporting asymmetry*, with unstyled emails that were late or incomplete.

B. Requirements

The requirements derived from this analysis are listed in Table II. NFR-2 was set at 4.5 s after discussions with users about acceptable waiting time for an interactive action.

TABLE II
FUNCTIONAL (FR) AND NON-FUNCTIONAL (NFR) REQUIREMENTS

ID	Requirement	Acceptance criterion
FR-1	Unified single-screen dashboard for folder creation, document access and summary	All three actions reachable in one click from the home screen
FR-2	Automated, standardised PO directory provisioning	Folder name generated programmatically from validated input
FR-3	Atomic ledger update after successful provisioning	Exactly one ledger row per newly created folder
FR-4	Native redirection to repository and master workbook	Opens in SharePoint/Excel Online without client-side data loading
FR-5	Scheduled executive summary	Styled email delivered weekly without human action
NFR-1	Corporate brand compliance (Navy #002060, Slate #1F4E78)	Approved by the business owner
NFR-2	Latency	Provisioning and logging complete in ≤ 4.5 s
NFR-3	Input validation	Blank or malformed input rejected on the client
NFR-4	Schema resilience	Flows unaffected by changes to non-critical ledger columns

C. Cyber-Security and IT-Policy Constraints

The solution had to be deployable without exceptions to the organisation's IT policy. Table III lists the constraints that shaped the design and the corresponding design response. Several choices that might otherwise look like workarounds, such as styling emails with inline CSS or redirecting users to native SharePoint pages, follow directly from these constraints.

TABLE III
SECURITY AND IT-POLICY CONSTRAINTS AND THE CORRESPONDING DESIGN RESPONSES

Constraint	Design response in POWAM
C1 Tenant boundary: no new servers, hosting or external data stores	All tiers implemented with Microsoft 365 services already licensed and governed by IT
C2 Identity: corporate single sign-on; no stored credentials or shared service passwords	Flows run through the signed-in user's connections; access is governed by the tenant identity provider
C3 Connector policy (DLP): only approved standard connectors in the business data group [35]	Uses only SharePoint, Excel Online (Business), Office 365 Outlook and built-in data operations; no HTTP or custom connectors
C4 No third-party libraries or rendering services	Report styling by inline CSS injected in a Compose action (Section VI-D)
C5 Least privilege and permission inheritance	New folders inherit library permissions; Launch() hands browsing to SharePoint, which enforces access control
C6 Auditability	Every provisioning action is recorded in run history and in the ledger tuple of (4)

V. SYSTEM ARCHITECTURE AND FORMAL MODEL

A. Five-Tier Architecture

POWAM uses the decoupled five-tier architecture shown in Fig. 2. The *presentation* tier is a Power Apps canvas application. The *orchestration* tier consists of two independent Power Automate engines: Engine A is invoked on demand by the app, and Engine B runs on a weekly schedule. The *repository* tier is a SharePoint Online document library addressed through server-relative paths. The *structured-data* tier is an Excel Online table that acts as an operational ledger. The *notification* tier uses Exchange Online to deliver reports. Each tier communicates only through standard connectors, so each can be replaced independently, for example by migrating the ledger to Dataverse (Section IX-D).

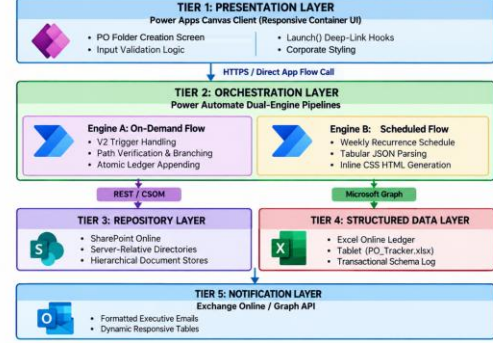


Fig. 2. Five-tier decoupled architecture of POWAM: presentation (Power Apps), orchestration (two Power Automate engines), repository (SharePoint Online), structured data (Excel Online ledger) and notification (Exchange Online).

B. Path, Ledger and Idempotency Model

Let P_{base} denote the fixed server-relative root of the purchasing library (anonymised here):

$$P_{base} = "/sites/<PurchasingSite>/$$

$$\text{Shared Documents}/<Library>/" \quad (2)$$

For a validated PO identifier x , the target directory is obtained by string concatenation ($\parallel$) of the root, a fixed prefix and the trimmed identifier:

$$P_{target}(x) = P_{base} \parallel "PO_ " \parallel \text{Trim}(x) \quad (3)$$

The ledger is an OpenXML table whose rows are tuples $S_{ledger} = \langle PO_ID, DirectoryPath, CreatedBy,$

$$\text{Timestamp, OperationalStatus} \rangle \quad (4)$$

Let R be the set of existing directories and $\text{Prov}(R, x)$ the provisioning operation. Prov returns $R \cup \{P_{target}(x)\}$ and appends one ledger tuple if $P_{target}(x) \notin R$, and returns R unchanged with no ledger write otherwise. Repeated submission of the same identifier therefore leaves the system in the same state, which is the idempotency property we require:

$$\text{Prov}(\text{Prov}(R, x), x) = \text{Prov}(R, x) \quad (5)$$

Equation (5) guarantees at most one folder and one ledger row per PO for sequentially processed requests. The guarantee under truly concurrent submissions of the same identifier is discussed as a limitation in Section IX-C.

VI. IMPLEMENTATION

A. Client Application and Input Validation

The home screen (Fig. 3) is built from nested horizontal and vertical layout containers rather than fixed pixel coordinates, so it renders consistently at 1366×768 and 1920×1080 and at different browser zoom levels. Three elevated cards expose the core functions: Folder Creation, Executive Summary and PO View. Colours follow the corporate design system (navy #002060 and #1F4E78, neutral fill #F8F9FA, charcoal text #212529).

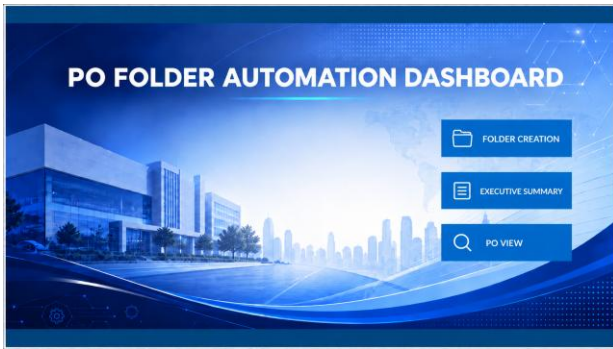


Fig. 3. POWAM dashboard home screen with the Folder Creation, Executive Summary and PO View cards (production screenshot).

The Folder Creation view (Fig. 4) contains a labelled text input (txtPONumber), an action button (btnCreatePO) and a navigation icon. To avoid unnecessary flow invocations, the button's OnSelect property validates the input on the client before calling the flow (Listing 1).



Fig. 4. PO Folder Creation view with the PO number input and the Create PO Folder action (production screenshot).

Listing 1. Power Fx OnSelect logic of btnCreatePO

```

If (
    IsBlank(txtPONumber.Text),
    Notify("Validation Error: Purchase
        Order Number cannot be empty.",
        NotificationType.Error, 4000),
    UpdateContext({locIsSubmitting: true});
    'PO-Folder-Creation-Flow'.Run(
        Trim(txtPONumber.Text));
    Notify("PO Folder Created &
        Cataloged Successfully!",
        NotificationType.Success, 3000);
    Reset(txtPONumber);
    UpdateContext({locIsSubmitting: false})
)

```

B. Decoupling Through Native Deep-Linking

Early prototypes embedded the document library and the master workbook inside the canvas app using gallery controls. Two problems appeared in testing. First, queries that cannot be delegated to the data source are evaluated on the client over a capped number of rows (500 by default, configurable up to 2,000) [36], so large libraries were listed incompletely. Second, simultaneous users of the embedded workbook triggered table-locking errors. We therefore replaced the embedded views with Launch() calls that open the native SharePoint folder and the Excel Online workbook in the browser (e.g., Launch("https://<tenant>.sharepoint.com/sites/<Purchasing Site>/...")). This keeps the app lightweight and leaves access control to SharePoint (constraint C5).

C. Engine A: Event-Driven Idempotent Provisioning

Engine A is started by the *When Power Apps calls a flow* (V2) trigger with the parameter PO NO. Its control flow is shown in Fig. 5. The flow first composes P_{target} (3) and calls the SharePoint action *Get folder metadata using path*, which corresponds to the REST request in Listing 2 [37]. Because a missing folder makes this action fail with HTTP 404, the following condition is configured to run after both success and failure of the metadata action and branches on its status code. If the folder exists (HTTP 200), the flow ends without any write, which prevents duplicates and overwrites. Otherwise it calls *Create new folder* and then *Add a row into a table* to append the tuple of (4) to PO_Tracker.xlsx.

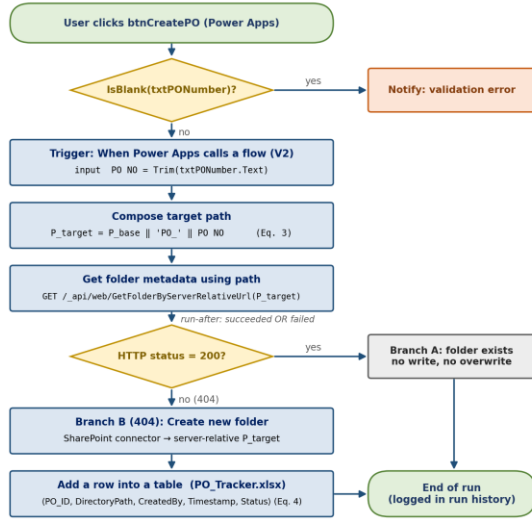


Fig. 5. Control flow of Engine A. Client-side validation precedes the flow call; the metadata check makes provisioning idempotent (5).

Listing 2. REST request issued by the metadata check (identifiers anonymised)

```

GET /_api/web/GetFolderByServerRelativeUrl(
  '/sites/<PurchasingSite>/Shared Documents/
  <Library>/PO_109283') HTTP/1.1
Host: <tenant>.sharepoint.com
Accept: application/json;odata=verbose
  
```

D. Engine B: Scheduled Reporting With Inline CSS Injection

Engine B (Fig. 6) runs every Monday at 08:00 IST. It reads all rows of the ledger with *List rows present in a table*, converts the resulting JSON array into HTML with *Create HTML table* and sends it through *Send an email (V2)*. The built-in HTML table action produces bare markup without borders or padding, and constraint C4 ruled out external templating services. We therefore prepend a style block in a *Compose* action (Listing 3). The output is bound to the email body, producing the layout illustrated in Fig. 7.

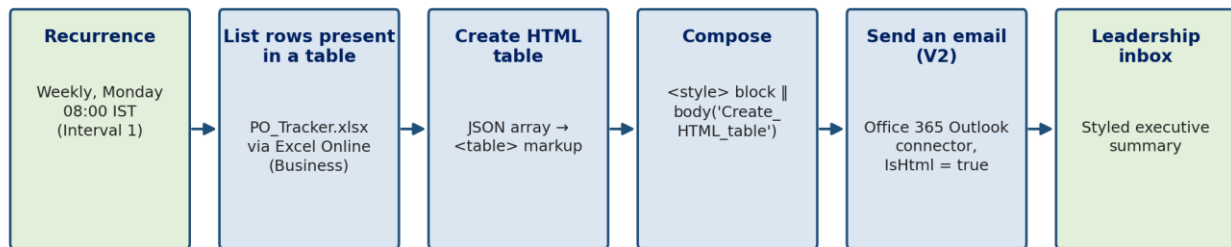


Fig. 6. Pipeline of Engine B, from the weekly recurrence trigger to delivery of the styled executive summary.

Weekly PO Folder Automation Summary

PO_ID	DirectoryPath	CreatedBy	Timestamp	OperationalStatus
PO_4500918231	.../COMMON/PO_4500918231	Operator A	2026-06-01 10:14	Folder created
PO_4500918247	.../COMMON/PO_4500918247	Operator B	2026-06-01 11:02	Folder created
PO_4500918252	.../COMMON/PO_4500918252	Operator A	2026-06-02 09:41	Folder created
PO_4500918266	.../COMMON/PO_4500918266	Operator C	2026-06-03 15:27	Folder created
PO_4500918270	.../COMMON/PO_4500918270	Operator B	2026-06-04 12:08	Folder created

Fig. 7. Illustrative rendering of the weekly summary produced by the stylesheet in Listing 3. Rows contain Anonymised PO identifiers and anonymised operator names; production data are confidential.

Listing 3. Compose action input combining the stylesheet with the generated table (abridged)

```
<style>
  table { border-collapse: collapse;
    width: 100%; font-size: 13px;
    font-family: 'Segoe UI', Arial,
      sans-serif; }
  th { background-color: #1F4E78;
    color: #FFFFFF; font-weight: 600;
    padding: 10px 14px;
    border: 1px solid #D9D9D9; }
  td { padding: 8px 12px; color: #333333;
```

```
border: 1px solid #E0E0E0; }
tr:nth-child(even) {
  background-color: #F2F6FA; }
</style>
@{body('Create_HTML_table')}
```

VII. TECHNICAL CHALLENGES AND RESOLUTIONS

Three defects found during demonstration are summarised in Table IV. They are typical of the integration and data-storage problems that dominate low-code developer discussions [26], [27].

TABLE IV
INTEGRATION DEFECTS OBSERVED DURING DEMONSTRATION, WITH ROOT CAUSES AND RESOLUTIONS

Defect	Symptom	Root cause	Resolution
D1 Flow namespace collision	Compile error: 'Run' is an unknown or unsupported function in namespace 'PO-Folder-Creation-Flow'	Administrative Power Automate management connectors were added through the Data panel; they expose tenant-management APIs, not run hooks of a specific flow	Removed the admin connectors and bound the flow via Action → Power Automate → Add flow, which generates the .Run() contract
D2 Dynamic-content schema drift	Runtime error WorkflowOperationParametersExtraParameter for parameter 'Item/Justification'	A deleted ledger column remained as a cached token in the action's JSON payload; the Excel API rejected it	Deleted the stale token and re-selected the table to force a fresh schema fetch
D3 Workbook not selectable in connector	Master workbook visible in SharePoint but greyed out in the Power Apps Excel dialog	Data stored as a plain cell range rather than a declared table object, which the connector requires	Avoided altering legacy formulas; replaced the embedded connection with Launch() redirection

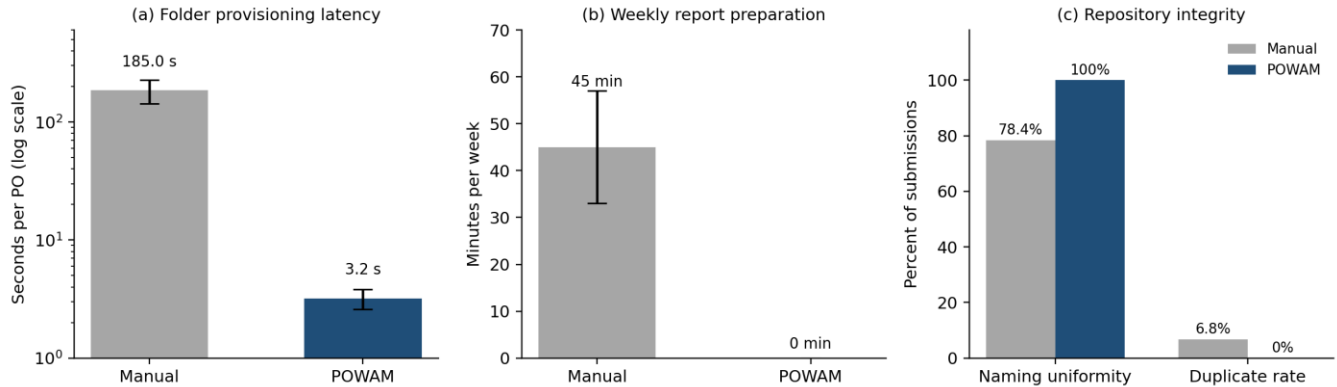


Fig. 8. Comparison of the manual baseline and POWAM: (a) provisioning latency per PO on a logarithmic scale with ± 1 SD error bars; (b) weekly report preparation effort; (c) naming uniformity and duplicate rate.

D2 is a concrete instance of the schema-evolution problem described in the process-flexibility literature [18]: loosely typed stores such as spreadsheets allow the schema to change without notifying dependent processes. It is also the reason behind NFR-4 and the migration recommendation in Section IX-D.

VIII. RESULTS

Table V compares the manual baseline with POWAM over the eight-week trial, and Fig. 8 visualises the main indicators.

TABLE V
MANUAL BASELINE VERSUS POWAM OVER THE EIGHT-WEEK
PRODUCTION TRIAL

Indicator	Manual baseline	POWAM	Change
Folder provisioning latency L	185.0 ± 42.0 s	3.2 ± 0.6 s	-98.27% (1)
Naming uniformity U	78.4%	100.0%	+21.6 pp
Duplicate directory rate D	6.8%	0.0%	Eliminated
Weekly report preparation R	45.0 ± 12.0 min	0 min	-100%
Report presentation	Unstyled plain text	Styled HTML table	Qualitative
Flow execution success S	n/a	99.8%	n/a

Mean provisioning latency fell from 185.0 s to 3.2 s, well within the 4.5 s target of NFR-2. Using the pooled standard deviation of the two conditions (29.7 s), the effect size is Cohen's $d \approx 6.1$, far beyond the conventional threshold for a large effect (Welch's $t = 61.21$, $p < 0.001$, 95% CI of the difference = 175.94–187.66 s). In practical terms each order saves about 182 s, or approximately 50 person-hours per 1,000 purchase orders; removing the weekly report preparation saves a further 39 h per year (45 min $\times$ 52 weeks). Folder naming became fully uniform because names are generated programmatically, and the metadata check prevented any duplicate directory. Of all flow runs, 99.8% completed successfully; the remaining runs failed because of transient SharePoint and Power Automate connector errors and were re-submitted without creating duplicates, as guaranteed by (5).

IX. DISCUSSION

A. Interpretation

Three design principles explain most of the improvement. The first is *pragmatic decoupling*: rather than forcing document browsing and large-table manipulation into the canvas app, POWAM delegates them to the native SharePoint and Excel interfaces, which avoids delegation limits and write locks. The second is *idempotency as an integrity mechanism*: a single existence check turned an error-prone manual convention into a property that holds by construction. The third is *policy compliance as a design driver*. Restricting ourselves to approved standard connectors initially felt limiting, but it produced a system that passed security review without exceptions and that IT can support.

This is consistent with Kopper et al., who found business-managed IT to be successful when it is aligned with the IT department and scoped to local needs [7], and with the emphasis on governance in citizen-development research [14], [29].

B. Implications for AI/ML-Based Document Intelligence

From an AI/ML perspective, the most important outcome is not speed but structure. Every PO now has exactly one folder with a predictable path and a ledger row that links it to an identifier, an operator and a timestamp. This turns the repository into a labelled corpus in which documents can be associated with business objects without manual curation, a prerequisite for supervised information extraction from invoices and specifications [16], [33]. It also makes it feasible to move towards intelligent process automation [21], [22] and generative summarisation of order histories [34], provided the trust and transparency concerns identified by Mayr et al. are addressed [23].

C. Limitations and Threats to Validity

Several limitations should be noted. *External validity* is limited because the study covers one division of one organisation on one vendor platform; results may differ in other tenants, policies or ERP landscapes. *Internal validity* is limited by the non-randomised before–after design: baseline and automated latencies were measured with different instruments (observation versus run-history logs), and operators may have worked faster or slower while being observed. The automated latency is measured server-side and excludes client round-trip time. On the technical side, the existence check and the folder creation are separate operations, so two simultaneous submissions of the same new PO could in principle both pass the check; this did not occur during the trial but is a known check-then-act window. The client also displays a success message without waiting for the flow's outcome, so a user who resubmits an existing PO is not told that the folder already existed. Finally, the Excel ledger remains vulnerable to schema drift (D2) and to concurrency limits at higher volumes.

D. Scalability Roadmap

Planned extensions address these limitations. Migrating the ledger to Microsoft Dataverse or Azure SQL Database will provide typed schemas, row-level security and relational integrity. Returning a status payload from Engine A to the app will replace the optimistic notification, and enforcing uniqueness at the data layer will close the concurrency window.

Triggering provisioning directly from PO approval events in the organisation's SAP S/4HANA system would remove manual entry altogether. Integrating a document-intelligence service, subject to the same policy review, would allow vendor invoices and specification sheets uploaded to PO folders to be parsed and validated automatically.

X. CONCLUSION

This paper presented POWAM, a purchase order workflow automation and management system implemented on the Microsoft Power Platform within the cyber-security and IT-policy constraints of a large automotive manufacturer. By combining a Power Apps client, a SharePoint Online repository, an Excel Online ledger and two Power Automate engines, the system replaced manual folder provisioning and weekly reporting with an idempotent, auditable and policy-compliant pipeline. In an eight-week production trial it reduced mean provisioning time by 98.27%, eliminated duplicate directories, achieved full naming uniformity and removed manual reporting effort. Beyond these operational gains, the work shows that a well-governed low-code system can create the structured data foundation on which AI/ML-based document intelligence can later be built. We hope that the architecture, formal model and documented failure modes serve as a practical reference for similar deployments in manufacturing.

DECLARATIONS

Funding: The authors did not receive support from any organisation for the submitted work.

Competing interests: The authors declare no competing interests.

Ethics approval: The study used operational system logs and process observation within the organisation; no personal data beyond operator identifiers in system logs were processed.

Data availability: Operational data are proprietary to the case organisation and cannot be shared publicly.

REFERENCES

- [1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.
- [2] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *J. Intell. Manuf.*, vol. 31, no. 1, pp. 127–182, 2020, doi: 10.1007/s10845-018-1433-8.
- [3] C. Legner, T. Eymann, T. Hess, et al., "Digitalization: opportunity and challenge for the business and information systems engineering community," *Bus. Inf. Syst. Eng.*, vol. 59, no. 4, pp. 301–308, 2017.
- [4] W. Schnellbacher and D. Weise, *Jumpstart to Digital Procurement: Pushing the Value Envelope in a New Age*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-51984-1.
- [5] A. P. Iglesias-Pardo, D. L. de Mattos Nascimento, R. Goyannes Gusmão Caiado, J. Moyano-Fuentes, and J. M. Maqueira, "From technology adoption to digital transformation: a strategic framework for responsive operations and supply chain management," *Oper. Manag. Res.*, vol. 19, no. 3, 2026, doi: 10.1007/s12063-026-00610-2.
- [6] S. Haag and A. Eckhardt, "Shadow IT," *Bus. Inf. Syst. Eng.*, vol. 59, no. 6, pp. 469–473, 2017.
- [7] A. Kopper, M. Westner, and S. Strahinger, "From shadow IT to business-managed IT: a qualitative comparative analysis to determine configurations for successful management of IT by business entities," *Inf. Syst. e-Bus. Manage.*, vol. 18, no. 2, pp. 209–257, 2020, doi: 10.1007/s10257-020-00472-6.
- [8] W. M. P. van der Aalst, M. Bichler, and A. Heinzl, "Robotic process automation," *Bus. Inf. Syst. Eng.*, vol. 60, no. 4, pp. 269–272, 2018, doi: 10.1007/s12599-018-0542-4.
- [9] P. Hofmann, C. Samp, and N. Urbach, "Robotic process automation," *Electron. Markets*, vol. 30, no. 1, pp. 99–106, 2020, doi: 10.1007/s12525-019-00365-8.
- [10] L.-V. Herm, C. Janiesch, A. Helm, F. Imgrund, A. Hofmann, and A. Winkermann, "A framework for implementing robotic process automation projects," *Inf. Syst. e-Bus. Manage.*, vol. 21, no. 1, pp. 1–35, 2023, doi: 10.1007/s10257-022-00553-8.
- [11] A. C. Bock and U. Frank, "Low-code platform," *Bus. Inf. Syst. Eng.*, vol. 63, no. 6, pp. 733–740, 2021, doi: 10.1007/s12599-021-00726-8.
- [12] D. Di Ruscio, D. Kolovos, J. de Lara, A. Pierantonio, M. Tisi, and M. Wimmer, "Low-code development and model-driven engineering: two sides of the same coin?" *Softw. Syst. Model.*, vol. 21, no. 2, pp. 437–446, 2022, doi: 10.1007/s10270-021-00970-2.
- [13] A. Bucaioni, A. Cicchetti, and F. Ciccozzi, "Modelling in low-code development: a multi-vocal systematic review," *Softw. Syst. Model.*, vol. 21, no. 5, pp. 1959–1981, 2022.
- [14] B. Binzer and T. J. Winkler, "Democratizing software development: a systematic multivocal literature review and research agenda on citizen development," in *Software Business (ICSOB 2022), Lecture Notes in Business Information Processing*, vol. 463. Cham, Switzerland: Springer, 2022, pp. 244–259.
- [15] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Markets*, vol. 31, no. 3, pp. 685–695, 2021.
- [16] H. Gbada, K. Kalti, and M. A. Mahjoub, "Deep learning approaches for information extraction from visually rich documents: datasets, challenges and methods," *Int. J. Doc. Anal. Recognit.*, 2024, doi: 10.1007/s10032-024-00493-8.
- [17] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*, 2nd ed. Berlin, Heidelberg: Springer, 2018.
- [18] M. Reichert and B. Weber, *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Berlin, Heidelberg: Springer, 2012.
- [19] L. Ivančić, D. Suša Vugec, and V. Bosilj Vukšić, "Robotic process automation: systematic literature review," in *Business Process Management: Blockchain and Central and Eastern Europe Forum (BPM 2019), Lecture Notes in Business Information Processing*, vol. 361. Cham, Switzerland: Springer, 2019, pp. 280–295.

- [20] V. Leno, A. Polyvyanyy, M. Dumas, M. La Rosa, and F. M. Maggi, "Robotic process mining: vision and challenges," *Bus. Inf. Syst. Eng.*, vol. 63, no. 3, pp. 301–314, 2021.
- [21] T. Chakraborti, V. Isahagian, R. Khalaf, Y. Khazaeni, V. Muthusamy, Y. Rizk, and M. Unuvar, "From robotic process automation to intelligent process automation," in *Business Process Management: Blockchain and Robotic Process Automation Forum (BPM 2020)*, Lecture Notes in Business Information Processing, vol. 393. Cham, Switzerland: Springer, 2020, pp. 215–228.
- [22] C. Engel, P. Ebel, and J. M. Leimeister, "Cognitive automation," *Electron. Markets*, vol. 32, no. 1, pp. 339–350, 2022, doi: 10.1007/s12525-021-00519-7.
- [23] A. Mayr, P. Stahmann, M. Nebel, and C. Janiesch, "Still doing it yourself? Investigating determinants for the adoption of intelligent process automation," *Electron. Markets*, vol. 34, 2024, doi: 10.1007/s12525-024-00737-9.
- [24] G. D. Kerpedzhiev, U. M. König, M. Röglinger, and M. Rosemann, "An exploration into future business process management capabilities in view of digitalization," *Bus. Inf. Syst. Eng.*, vol. 63, no. 2, pp. 83–96, 2021.
- [25] D. Beverungen, J. C. A. M. Buijs, J. Becker, et al., "Seven paradoxes of business process management in a hyper-connected world," *Bus. Inf. Syst. Eng.*, vol. 63, no. 2, pp. 145–156, 2021.
- [26] M. A. A. Alamin, G. Uddin, S. Malakar, S. Afroz, T. Haider, and A. Iqbal, "Developer discussion topics on the adoption and barriers of low code software development platforms," *Empir. Softw. Eng.*, vol. 28, Art. no. 4, 2023, doi: 10.1007/s10664-022-10244-0.
- [27] K. Rokis and M. Kirikova, "Challenges of low-code/no-code software development: a literature review," in *Perspectives in Business Informatics Research (BIR 2022)*, Lecture Notes in Business Information Processing, vol. 462. Cham, Switzerland: Springer, 2022, pp. 3–17, doi: 10.1007/978-3-031-16947-2_1.
- [28] R. Waszkowski, "Evaluating low-code development platforms with the CD score," *Sci. Rep.*, vol. 15, Art. no. 38548, 2025.
- [29] B. Binzer and T. J. Winkler, "Die vier Phasen von Citizen-Development-Initiativen: Treiber, Herausforderungen und Handlungsempfehlungen," *HMD Prax. Wirtsch. Inform.*, vol. 61, no. 5, pp. 1136–1158, 2024.
- [30] B. Otto, "Data governance," *Bus. Inf. Syst. Eng.*, vol. 3, no. 4, pp. 241–244, 2011.
- [31] C. Pautasso, "RESTful web services: principles, patterns, emerging technologies," in *Web Services Foundations*, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, Eds. New York, NY, USA: Springer, 2014, pp. 31–51.
- [32] I. Baldini, P. Castro, K. Chang, et al., "Serverless computing: current trends and open problems," in *Research Advances in Cloud Computing*, S. Chaudhary, G. Somani, and R. Buyya, Eds. Singapore: Springer, 2017, pp. 1–20.
- [33] I. H. Sarker, "Machine learning: algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, Art. no. 160, 2021.
- [34] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Bus. Inf. Syst. Eng.*, vol. 66, no. 1, pp. 111–126, 2024.
- [35] Microsoft, "Data policies (data loss prevention) for Power Platform," Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/power-platform/admin/wp-data-loss-prevention> (accessed Jun. 8, 2026).
- [36] Microsoft, "Understand delegation in a canvas app," Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/delegation-overview> (accessed Jun. 10, 2026).
- [37] Microsoft, "Get to know the SharePoint REST service," Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-to-know-the-sharepoint-rest-service> (accessed Jun. 14, 2026).