TEOX INTELLIGENT AUTOMATION FOR MANUFACTURING
TOTAL ENGINEERING OPTIMISATION & EXECUTION

CASE STUDY DOCUMENT CONTROL & RISK GATE

Automating Engineering Document Control

How a multi-division engineering manufacturer replaced manual filing, registers and transmittals with an email-driven, AI-routed document-control system built entirely on its existing Microsoft 365 environment.

ANNUAL SAVING

£120,000

of document-control effort released, every year

RETURN ON INVESTMENT

Four hundred document transactions a month. Issues, returns, transmittals and comment-response sheets, each carrying about twenty-five minutes of reading, comparing, filing and retyping. The workflow does the handling. People do the engineering.

The arithmetic: 400 transactions per month × 25 minutes removed × £60 per hour blended loaded cost × 12 months = £120,000 a year.

Every TEOX build is priced so that it pays for itself inside twelve months on your own figures. That is a binding test we run before quoting, not a sales claim.

Zero	**1**	**100%**	**M365**
manual filing, one drop-zone	live register across all projects	auditable, append-only event log	you already own, no new platform

CLIENT CONTEXT

A multi-division manufacturer of engineered electrical equipment delivering project-based orders to industrial and energy-sector clients. Every project generates a continuous flow of controlled documents (drawings, datasheets, calculations) that must be issued to the customer, returned with comments, revised and re-issued, with every transaction logged and auditable. Across many concurrent projects and several operating divisions, this document-control burden was almost entirely manual.

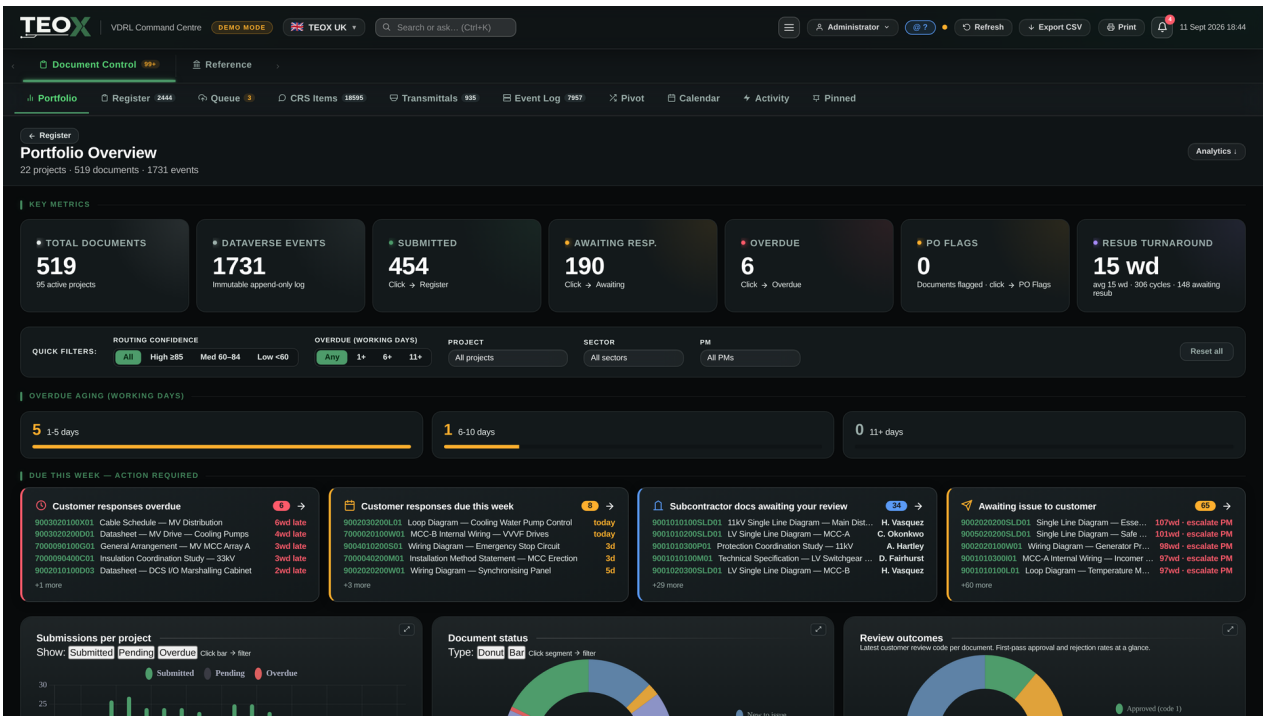
THE CHALLENGE

- Engineers filed documents by hand into per-project locations, needing to know exactly where each one belonged.
- The document register lived in spreadsheets that could not handle concurrent updates, offered no audit trail, and broke down at scale.
- Customer-returned, marked-up documents were processed by hand: someone read every comment, compared it against the issued revision, drafted a comment-response sheet, and distributed the actions.
- Transmittals and submission packages were assembled manually for every issue.

- Commercial deviations buried in technical comments (scope or rated-value changes outside the purchase order) were easy to miss until they became disputes.
- Subcontractor and sub-supplier documents needed an internal review gate before reaching the customer, also managed manually.
- There was no single, real-time view of document status across projects.

THE APPROACH

The system was built entirely within the organisation's existing Microsoft 365 environment, so there was no new infrastructure to procure and no data leaving the tenancy. SharePoint holds the document files; Microsoft Dataverse holds the data as a single source of truth; Power Automate is the engine; and a large language model API provides the intelligence that reads, classifies, compares and drafts. It was designed and configured by a domain expert directing AI, not by a traditional development team.



THE SOLUTION

Architecture: two fixed entry points and one central hub

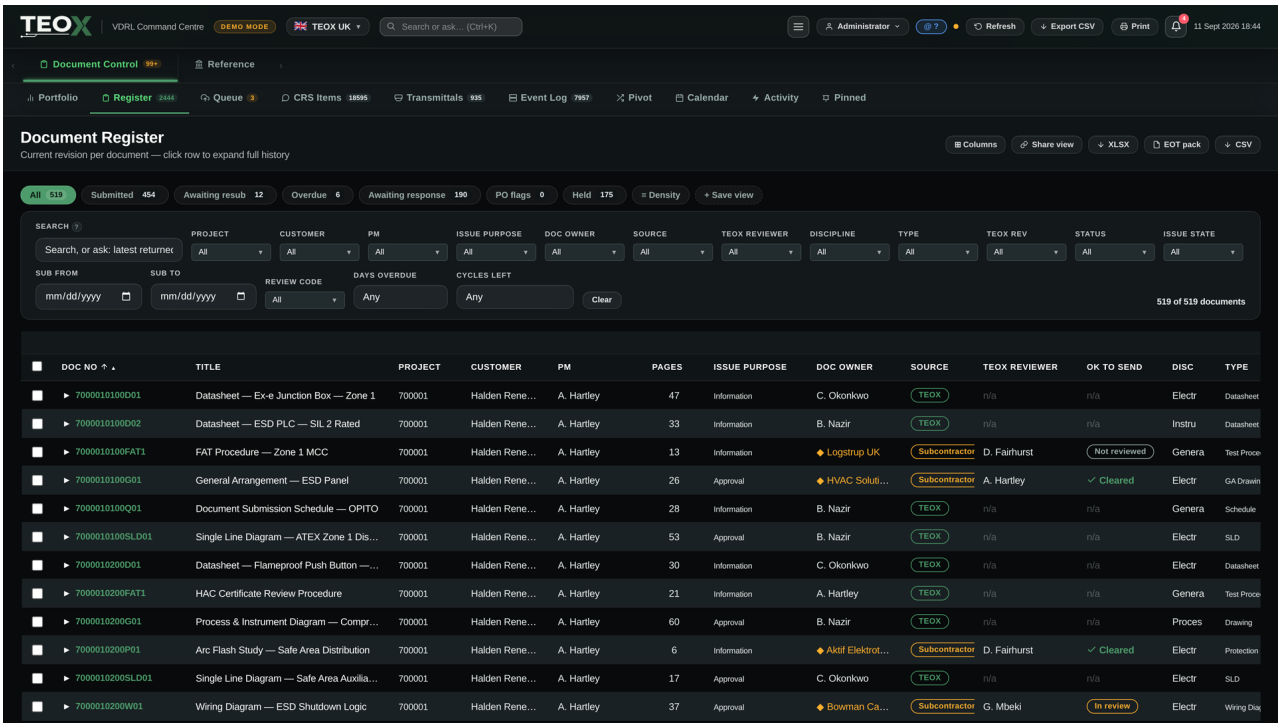
- A dedicated inbound mailbox receives all customer-returned documents. The workflow watches it, so engineers and project managers never monitor it manually.
- A dedicated outbound mailbox carries every submission package to the document controller, giving one consistent, auditable record of everything issued.
- A central SharePoint hub provides a single drop-zone. Engineers drop outgoing documents in one place; the workflow reads each document, identifies the project, and files it to the correct project site automatically.
- A central, append-only register in Dataverse records every event, while per-project SharePoint sites hold the files.

Inbound workflow: customer returns to comment-response sheet

1. **Trigger.** An email arrives at the returns mailbox and the workflow fires instantly, capturing the body and subject for context alongside the attachments.
2. **Extract and identify.** Attachments are stripped and the file type detected per item: tracked-changes Word, annotated PDF, or scanned handwritten markup, with AI vision reading handwriting and markup symbols.
3. **Retrieve baseline.** AI identifies the document number and revision, then pulls the issued version and the project purchase order as the baseline for comparison.
4. **Diff and classify.** AI compares the returned document against the issued revision, separating design changes, clarifications, contractual comments and cosmetic markups.
5. **Purchase-order check.** Every comment is checked against the purchase order. Anything implying out-of-scope work, a changed rated value, or a referenced exclusion is flagged immediately, regardless of confidence.
6. **Confidence scoring and routing.** Each comment is scored. High-confidence items are auto-filed and distributed; mid-confidence items are routed to a reviewer with the AI's reasoning shown; low-confidence items are held for human resolution; and any commercial deviation is escalated to the project and commercial leads at once.
7. **Assemble and distribute.** A comment-response sheet is compiled to a standard template, filed to the project site, and action owners are notified, with a full audit entry written.

Outbound workflow: drop-zone to customer submission

8. **Drop.** An engineer drops one or more documents into the central hub. No naming convention is enforced and no project selection is required; documents for different projects can be dropped together and are split and routed independently.
9. **Read and route.** AI reads each document, identifies the project, and files it to the correct project site.
10. **Transmittal, register and package.** A transmittal is generated, the central register updated, and a complete, ready-to-send submission package (including the covering email) produced for the document controller. Subcontractor documents pass through an internal review gate before they can be issued to the customer.



Data backbone: why a relational store, not a spreadsheet

The register is structured data, not a document, so it belongs in a relational database. Dataverse sits inside the existing Microsoft environment, integrates natively with Power Automate, SharePoint and Power BI, and is governed by the organisation's existing data policies, so there is no third-party data processor and no data leaving the tenancy. It supports row-level security (project managers see their own projects, controllers see everything, and customers can be given read-only views of only their own records), logs every write for a complete audit trail, and scales to millions of rows without performance loss. Shared spreadsheets and SharePoint lists cannot handle concurrent automated writes, querying or auditing at this scale.

An immutable event log

The register is an append-only event log. Every submission, return, revision and status change is a new row. Nothing is edited and nothing is deleted, and current status is a calculated view over those events. That gives you a complete history that can be queried at any time and cannot be quietly rewritten, which is the right model for regulated and contractual project work. It scales to hundreds of thousands of document events a year. Everything surfaces through one role-secured dashboard that many people can use at once.

IMPACT

- Zero manual filing: engineers interact with one drop-zone and the system files everything to the right place.
- A single source of truth: one register across all projects, always current and queryable in real time.
- Nothing slips: customer returns are read, compared and routed automatically, and commercial deviations from the purchase order are caught the moment they appear in a technical comment.
- Transmittals and submission packages are assembled by the workflow, not by hand.
- A complete audit trail: every event logged by user, time and change, suitable for regulated and contractual work.

- Cross-project visibility and control through one role-secured dashboard, built to scale across divisions and concurrent projects with no new infrastructure.

THE ECONOMICS: WHY THIS APPROACH

Because the system runs inside the existing Microsoft 365 tenancy, there was no new platform to buy, no data leaving the environment, and no third-party data processor to assess. The intelligence layer is a low-cost, pay-per-use AI API. Specified, built and iterated by a domain expert directing AI rather than delivered as a multi-month systems-integration project, it reached a working state far faster and at a fraction of the cost of a conventional document-management implementation.

> **WHAT TRANSFERS TO YOUR BUSINESS**
>
> Any project-based engineering, manufacturing or contracting business that issues and receives controlled documents carries the same burden. The transferable design is consistent: separate the files (SharePoint) from the data (a proper relational store); make the register an immutable event log; give people one entry point and let automation handle the filing and routing; use AI to read and compare documents and to catch commercial risk hidden in technical comments; and secure a single dashboard by role for visibility. It sits on tools most organisations already own.

What the handling removed is worth, across the portfolio	Figure
Document transactions per month, across live projects	400
Manual handling removed per transaction	25 minutes
Time recovered per month	167 hours
Client’s blended loaded labour cost	£60 per hour
Monthly benefit	£10,000
Annual benefit	**£120,000**

Deliberately excluded from that arithmetic: the commercial deviations caught the moment they appear in a technical comment rather than at final account, and the delay and extension-of-time evidence the event log makes available. Both are worth more than the handling time on a single contested project, and neither is claimed here, because a number that depends on a dispute happening is not a number you can plan against.

Tools used: Microsoft 365 (SharePoint Online, Exchange, Power Automate, Dataverse, Power BI, Entra ID) · large language model API (Claude) for document reading, classification, comparison and drafting.