

TECH & AI CASE STUDY - INSURANCE - 2025

Low-Code Internal Apps That Cleared a Six-Month Backlog in Three Weeks

Based on a delivery engagement with Aldergate Mutual

**6 months to 3
weeks**

IT REQUEST BACKLOG

-71%

ENDORSEMENT TURNAROUND

14

SHADOW SPREADSHEETS RETIRED

\$1.1M

ANNUAL OPERATING SAVINGS

A fillable practice edition: the full story, 8 actions,
6 reflections and a 90-day plan - work it with your team.

Free - no email required
emersontechnology.com

MAKE THIS PRACTICE BOOK YOURS

Work it, do not just read it

This practice book turns the Aldergate Mutual case study into a working session for your team. Read the full story first - what happened, what was built and what it changed. Then tick the 8 actions you can commit to, write short answers to the 6 reflections, and finish with a 90-day plan. Every field in this PDF is fillable: open it in Adobe Acrobat Reader, Chrome, Edge or macOS Preview, type directly into the boxes and save.

YOUR NAME	ROLE
COMPANY	TEAM / DEPARTMENT
DATE	EMAIL (OPTIONAL)

MY GOAL FOR THIS PRACTICE SESSION

- 01 The story** - What happened at Aldergate Mutual - before, during and after the project.
- 02 The results** - The numbers that changed, plus industry benchmarks from public research.
- 03 The 8-action checklist** - Tick boxes as your organization completes each step. Owner and date fields included.
- 04 The 6 reflections** - Fillable questions that pressure-test your own roadmap.
- 05 My 90-day plan** - Three fillable phase cards that turn this case study into your roadmap.

RUN IT AS A 90-MINUTE TEAM SESSION

25 min READ Read the story together. Ask everyone to note the one number that surprised them most.	40 min PRACTICE Work the 8-action checklist and the 6 reflections. Assign an owner and a date to every action you commit to.	25 min COMMIT Fill in the 90-day plan. One goal per phase, one number that tells you it worked. Book a review date before you leave the room.
--	--	---

THE STORY

What happened, and why it worked

01 The Company

Aldergate Mutual is a regional insurer with a 92-year history, 620 employees, and a book of business concentrated in small commercial and personal lines across its home region. Longevity is the company's identity: generations of producers have placed policies with Aldergate because the insurer pays claims fairly and knows its agents by name. The operations organization — underwriting support, policy services, claims intake and producer management — runs on deep institutional knowledge held largely by long-tenured staff.

That longevity, however, was inherited by the technology. The company's core policy administration system is a legacy platform so stable and so entrenched that changing anything in it took a nine months' development cycle, and the surrounding operations had grown around that constraint. Underwriting, claims and producer management each maintained their own spreadsheet tools to bridge the gaps. The result was an organization whose people were excellent and whose processes were held together by the diligence of the same people — a structure that worked until the volume of exceptions, the pace of competitor response and the patience of auditors all arrived at once.

02 Where Things Stood

Fourteen spreadsheet tools ran real operational processes, maintained by underwriters and operations staff themselves. Each spreadsheet had a human owner, an undocumented logic layer and a failure mode nobody had written down: a broken formula in a commissions workbook, a stale license-tracking tab, an endorsement log whose version history lived in file names like "final_v3_USE_THIS." Business requests queued behind a six-month IT backlog, so operations improvised, and improvisation at scale becomes infrastructure. Data was re-entered across systems, which meant every handoff was an opportunity for transcription error, and audit findings had piled up around version control — the auditors could not reconstruct who changed what, when, or why.

The operational cost was visible in the numbers. Simple endorsements — the kind of policy change a competitor's system processed in an afternoon — took five days and nine manual touchpoints at Aldergate. The CFO estimated process waste at \$1.1M annually in duplicated labor, error correction and rework. Meanwhile, competitors quoted small commercial policies in hours, and the producer community noticed. Gartner has projected that by 2025 roughly 70% of new applications developed by enterprises will use low-code or no-code technologies, up from under 25% in 2020 — and Aldergate's staff had already voted with their spreadsheets, building the applications the business needed through the only channel open to them.

Hiring developers was unrealistic in the local market; every opening competed with metro salaries the insurer could not match. McKinsey research suggests insurers that digitize core processes can cut cost-to-serve by 20-30% while reducing error rates and audit exposure, but Aldergate's constraint was not conviction — it was capacity. Operations leadership needed a way to build governed applications without waiting on scarce engineering resources, and the IT organization needed to stop being the bottleneck for every checkbox and status field.

03 The Challenge

The challenge was to clear a six-month IT request backlog and retire 14 shadow spreadsheets by building governed applications — governed being the word that separates this program from simply buying a low-code license and hoping. A regulated insurer cannot let citizen-built applications accumulate the way citizen-built spreadsheets did, or the same audit findings would return wearing nicer screens. Every app touching policyholder data needed SSO, role-based access, audit logging and a lifecycle that could survive an examination. The second challenge was organizational. Operations analysts understood the processes but had never built software; IT understood software but did not have capacity; and the two groups' incentives were not naturally aligned. The program needed a mixed delivery model — platform developers, operations analysts and an IT architect working as one team — plus explicit rules about who could build what, so that empowerment did not become abdication. And it needed visible early wins: a backlog measured in months had taught operations to treat IT requests as lost causes, and only delivered applications, retired spreadsheets and a CFO who could see his \$1.1M returning would change that culture.

04 Discovery and Diagnosis

Emerson Technology began with a three-week discovery centered on the spreadsheet estate: every spreadsheet running a real process was inventoried with its owner, its users, the data it touched and what breaks when it does. The inventory doubled as an app roadmap and a risk register — it showed which spreadsheets touched policyholder data, which made calculations with financial or regulatory consequence, and which were simply habit. Alongside the inventory, the discovery mapped the endorsement process end to end, counting its nine manual touchpoints and five-day elapsed time, and reviewed the audit findings to understand exactly what evidence gaps had been cited.

The diagnosis confirmed the pattern the spreadsheet estate suggested: the core policy-admin system was not the whole bottleneck — it was the absence of a governed middle layer where operational workflows could live. Requests that looked trivial from operations' side (a new intake form, a status field, a dispute queue) were expensive in the core system's development cycle, so they never ranked high enough to ship. Interviews with operations analysts showed strong process literacy and genuine appetite to build, provided guardrails existed first — governance added after twenty apps exist is governance nobody follows. The final plan stood up an enterprise low-code platform with hard guardrails, seeded delivery through a mixed team, and committed to retiring one legacy tool per launch so the program could not become a new form of accumulation.

05 The Solution

We stood up an enterprise low-code platform with guardrails designed for a regulated insurer, then delivered the first wave of applications through a deliberately mixed team.

Guardrails before builders

The governance frame preceded any citizen development: a shared component library of lookups, validation rules and UI patterns, so every new app starts from proven parts that already pass review; mandatory audit logging of every write, view and approval against named user identities; SSO and role-based access wired to the insurer's identity system; and a security review baked into every release. Risk-rated build lanes drew the line explicitly — operations analysts assemble simple apps inside predefined templates, while anything touching billing, ratings or claims decisions routes to IT, because the lanes prevent costly well-meaning disasters. A quarterly review retires whatever nobody uses, keeping the platform's population honest the way the retirement commitment keeps the spreadsheet estate honest.

The first twelve weeks of delivery

In the first 12 weeks a mixed team — two platform developers, four operations analysts and one IT architect — delivered 11 production apps, including endorsement intake, producer license tracking, claims triage and a commissions dispute workflow. The pairing model mattered: platform developers built the hard integration and data foundations, operations analysts owned process logic and interfaces, and the IT architect reviewed everything against the core system's constraints. Endorsement intake went first as the pilot — painful enough to prove the model, contained enough to avoid policyholder-data risk while the team learned. Each subsequent app inherited the component library and the release pipeline, so the ninth app shipped faster than the second, which is the entire argument for the platform.

Integration and data with a memory

Integration connectors sync everything with the core policy-admin system through APIs rather than file exports, because exports recreate the shadow-IT problem with nicer screens and worse governance. Application data lands in PostgreSQL with full history, so every endorsement, approval and producer record carries its own timeline — who changed what, when and under whose authority. That history is what converted audit findings into audit strengths: internal audit now cites the platform's logging as a control rather than a risk, and the state regulators received the same audit trail during the annual examination without any special preparation.

06 The First 90 Days

Weeks one through three were discovery and the spreadsheet inventory. Weeks four through six stood up the platform with its guardrails — component library, SSO integration, audit logging, the release pipeline with security review — and the mixed team was formed and trained. The endorsement intake app was built in weeks five through eight as the pilot, tested by the underwriting support team that had owned the spreadsheet version, and its retirement was declared a success criterion for the app itself: launching a replacement without decommissioning the shadow tool would only have added a fifteenth spreadsheet.

Weeks seven through ten delivered the next wave in parallel streams — producer license tracking, claims triage and the commissions dispute workflow — with the API connectors to the core policy-admin system landing in week nine. The backlog-clearing effect arrived in week eleven: with the platform and templates in place, the standing six-month request queue was triaged, and the majority of queued requests turned out to be simple workflow apps the team could now assemble in days. Weeks eleven and twelve shipped the remaining first-wave apps and ran the first quarterly review, which retired two small tools nobody was using and tightened two templates based on analyst feedback. By day 90, eleven production apps were live, the backlog was cleared, and operations requests were shipping on a days-not-months cadence for the first time in the insurer's recent history.

07 The Results

The six-month request backlog cleared in three weeks, and new operations requests now ship in days. Endorsement turnaround fell 71% to about 36 hours with two touchpoints instead of nine, and all 14 shadow spreadsheets were retired into governed apps. The CFO's \$1.1M waste estimate has been fully recovered, and internal audit now cites the platform's logging as a control strength rather than a risk — the same audit trail reached state regulators during the annual examination without special preparation.

METRIC	BEFORE	AFTER
IT request backlog	6 months	3 weeks
Endorsement turnaround	-	-71%
Shadow spreadsheets retired	-	14

METRIC	BEFORE	AFTER
Annual operating savings	-	\$1.1M

The compounding effects are organizational. Operations analysts who built their first apps inside templates now propose process improvements before problems accumulate, because the cost of acting on an idea fell from a nine-month queue to a working session. The insurer's response speed on small commercial quotes improved as the endorsement and producer workflows tightened, which the producer community registered immediately. Forrester's Total Economic Impact studies of major low-code platforms have reportedly found returns exceeding 150% within three years for enterprise adopters; Aldergate's recovery of the full \$1.1M waste estimate inside the first year puts it ahead of that curve, and the governance model means the gains are auditable rather than anecdotal.

08 What Your Business Should Do

The Aldergate program generalizes to any organization where critical processes live in spreadsheets and every change queues behind a scarce IT team. Inventory the spreadsheet estate first — list every spreadsheet running a real process, who owns it and what breaks when they do, because that inventory becomes your app roadmap and your risk register. Pilot one painful but contained process, choosing something that annoys everyone but cannot damage policyholder or customer data, since early wins with low stakes earn the political capital harder apps will need later. Set guardrails before citizen builders arrive — publish standards for data access, audit logging and app lifecycles before anyone opens the platform, because governance added after twenty apps exist is governance nobody follows. Build a shared component library centralizing lookups, validation rules and UI patterns so every new app starts from proven parts that already pass review; consistency here is what keeps forty apps maintainable.

Integrate through APIs, not exports, connecting the low-code layer to your core systems with full history — nightly spreadsheet exports recreate the shadow-IT problem with nicer screens and worse governance. Log audit trails from day one, requiring every write, view and approval to be recorded against a named user identity, because regulated businesses cannot retrofit accountability after an examination finds the gaps. Keep risk-rated build lanes: let analysts assemble simple apps from approved templates while anything touching billing, ratings or claims decisions stays with IT, since the lanes prevent costly well-meaning disasters. Finally, retire one legacy tool per launch — make decommissioning an existing spreadsheet or Access database a condition of each new app's success criteria, or you will accumulate platforms the way you accumulated spreadsheets. Apply this eight-step sequence to your own backlog with the action checklist in this practice book, and revisit the reflection questions when the quarterly platform review meets: how many business-critical spreadsheets run your operations and who actually understands each one, could your analysts build safe applications if guardrails existed before they started, and if an examiner asked for two years of approval history tomorrow, could your systems produce it?

REAL-WORLD PROOF & BENCHMARKS

Gartner has projected that by 2025 roughly 70% of new applications developed by enterprises will use low-code or no-code technologies, up from under 25% in 2020. - Gartner

Forrester's Total Economic Impact studies of major low-code platforms have reportedly found returns exceeding 150% within three years for enterprise adopters. - Forrester TEI studies

McKinsey research suggests insurers that digitize core processes can cut cost-to-serve by 20-30% while reducing error rates and audit exposure. - McKinsey & Company

THE PRACTICE PAGES

The 8-action checklist

Work through the list with your team. Tick each box as your organization completes the step, and fill in an owner and a target date so every action has a name and a deadline.

☐

1. Inventory the spreadsheet estate first

List every spreadsheet running a real process, who owns it and what breaks when they do. This inventory becomes your app roadmap and your risk register.

OWNER

TARGET DATE

STATUS

--	--	--

☐

2. Pilot one painful, contained process

Pick a process that annoys everyone but cannot damage policyholder data. Early wins with low stakes earn the political capital harder apps will need later.

OWNER

TARGET DATE

STATUS

--	--	--

☐

3. Set guardrails before citizen builders

Publish standards for data access, audit logging and app lifecycles before anyone opens the platform. Governance added after twenty apps exist is governance nobody follows.

OWNER

TARGET DATE

STATUS

--	--	--

☐

4. Build a shared component library

Centralize lookups, validation rules and UI patterns so every new app starts from proven parts that already pass review. Consistency here keeps forty apps maintainable.

OWNER

TARGET DATE

STATUS

--	--	--

☐

5. Integrate through APIs, not exports

Connect the low-code layer to core systems with APIs and full history, never nightly spreadsheet exports. Exports recreate the shadow-IT problem with nicer screens and worse governance.

OWNER

TARGET DATE

STATUS

--	--	--

☐

6. Log audit trails from day one

Require every write, view and approval to be logged against a named user identity. Regulated businesses cannot retrofit accountability after an examination finds the gaps.

OWNER

TARGET DATE

STATUS

--	--	--

☐

7. Keep risk-rated build lanes

Let analysts assemble simple apps from approved templates, but keep anything touching billing, ratings or claims decisions with IT. The lanes prevent costly well-meaning disasters.

OWNER

TARGET DATE

STATUS

--	--	--

☐

8. Retire one legacy tool per launch

Make decommissioning an existing spreadsheet or Access database a condition for each new app's success criteria. Otherwise you accumulate platforms the way you accumulated spreadsheets.

OWNER

TARGET DATE

STATUS

--	--	--

AFTER THE CHECKLIST: OUR TOP 3 PRIORITIES

From the eight actions above, write down the three you will start first - and the single number you expect each one to move. Three priorities with owners beat eight without any.

PRIORITY 1 NUMBER IT SHOULD MOVE	PRIORITY 2 NUMBER IT SHOULD MOVE	PRIORITY 3 NUMBER IT SHOULD MOVE
--	--	--

THE PRACTICE PAGES

Reflections for your team

Use these questions in your next planning session and type short answers into the boxes. The gaps you find are usually the fastest wins.

Q1. How many business-critical spreadsheets run your operations today, and who actually understands each one?

Q2. What does a six-month wait for a simple workflow cost you in workaround labor and errors?

Q3. Could your operations analysts build safe applications if guardrails existed before they started?

Q4. Which core-system limitations are truly technical constraints, and which are just inherited habits?

Q5. If an examiner asked for two years of approval history tomorrow, could your systems produce it?

Q6. Where would you draw the line between citizen-built apps and work that must stay with engineering?

THE PRACTICE PAGES

My 90-day plan

Finish the session by committing to a plan. Three phases, one goal each, one number that tells you it worked. Bring this page to your next leadership review - or to a free consultation with Emerson Technology, and we will pressure-test it with you.

Days 1-30 Focus: audit and alignment

PRIMARY GOAL**KEY ACTIONS****HOW WE MEASURE SUCCESS****OWNER**

Days 31-60 Focus: build and prove

PRIMARY GOAL**KEY ACTIONS****HOW WE MEASURE SUCCESS****OWNER**

Days 61-90 Focus: scale and measure

PRIMARY GOAL

KEY ACTIONS

HOW WE MEASURE SUCCESS

OWNER

Tip: save a copy per team so each group keeps its own answers. Revisit the reflections every quarter - the questions stay the same even as the numbers move.

BOOK THE REVIEW BEFORE YOU CLOSE THE BOOK

Plans without a review date drift. Before you close this practice book, agree on when you will come back to these pages, who will run that session, and when you will check the numbers again. If you want an outside pair of eyes, bring this page to a free consultation with Emerson Technology and we will pressure-test the plan with you.

REVIEW DATE	REVIEWED BY	NEXT CHECK-IN
------------------------------------	------------------------------------	--------------------------------------

Ready for a transformation like Aldergate Mutual's?

Emerson Technology designs and builds the data, AI, cloud and automation systems behind results like these. On a free consultation call we will map this exact playbook to your business - your data, your constraints, your numbers.

Book a free consultation

support@emersontechnology.com

or start online: emersontechnology.com/contact