

PRODUCT WORKFLOW STUDY

From hidden exceptions to prioritised operational alerts and accountable escalation

An exception-management workflow showing how Corexa can surface blocked jobs, overdue maintenance, stock risks, failed integrations, unread actions and other conditions that require intervention rather than routine processing.

Evidence level stated clearly throughout

Repository-reviewed

Capability basis

Exception handling

Primary workflow

Priority-led

Attention model

Pilot required

Outcome standard



THE CHALLENGE

Why the existing process breaks down

Operational systems work well when every job follows the expected path. Service businesses lose time at the edges: a part request fails, a completed job remains uninvoiced, an alert is duplicated, a customer reply is unread or an integration stops synchronising. These exceptions are individually small but collectively become the hidden workload of the business.

01

Routine and urgent look the same

A high-risk machine alert can sit beside a low-priority informational notification.

02

Blocked work has no owner

The system shows a problem but not the person responsible for resolving it.

03

Failures are discovered late

An integration or API exception may remain unnoticed until a customer or administrator reports the consequence.

04

Duplicate alerts create fatigue

Repeated notifications reduce confidence and make genuine priority harder to identify.

05

Resolution is not recorded

Staff fix the underlying issue but the alert remains open or no reason is preserved.

06

Management cannot see patterns

Recurring exceptions are handled case by case without identifying the failing workflow behind them.

The central risk

The platform can contain all necessary data while the business still misses the small number of exceptions most likely to delay service, lose revenue or damage trust.

THE OPERATING MODEL

A connected workflow around the machine

Corexa can detect operational conditions, apply context and priority, route them to a responsible user, record acknowledgement and preserve the resolution for later analysis.



COREXA CAPABILITY	OPERATIONAL CHANGE	WHY IT MATTERS
Contextual notifications	Creates alerts that can include the operational event and relevant entity.	The user can understand why attention is required.
Health and maintenance alerts	Surfaces condition and due-service issues from the machine lifecycle.	Technical risk can be prioritised before failure.
Inventory alerts and reports	Identifies low stock, stockout and movement concerns.	Parts risk can be addressed before it blocks work.
Unread and assignment indicators	Shows messages, forum activity or job events still requiring review.	Human follow-up is less dependent on memory.
Integration and issue reporting	Provides exception and issue-reporting pathways for failed processes.	Technical failures can become visible operational work.
Notification preferences and deduplication logic	Supports channel control and consolidated notification handling.	Users can reduce noise while retaining meaningful alerts.

CREDIBILITY STANDARD

What is supported - and what is not yet claimed

A credible case study distinguishes repository evidence, real workflow scope and future outcome hypotheses. The evidence level below is deliberately explicit.

✓ Repository capability

The repository contains notification providers, contextual and event-based routes, health services, inventory alerts, issue reports, unread counts and multiple audit or exception documents.

✓ Broad exception scope

Corexa can generate attention signals across jobs, machines, messages, inventory and integrations.

✓ Credible limitation

A large number of alert components can itself create inconsistency; production routing, ownership and deduplication must be tested.

✓ Pilot requirement

A defined exception register should record detection, ownership, acknowledgement, resolution and repeat cause over a controlled period.

Not yet claimed: this case study does not present a measured productivity, revenue, accuracy or customer-satisfaction improvement. A controlled baseline and participant-approved result are required before those claims are published.

Measurement plan for the publishable version

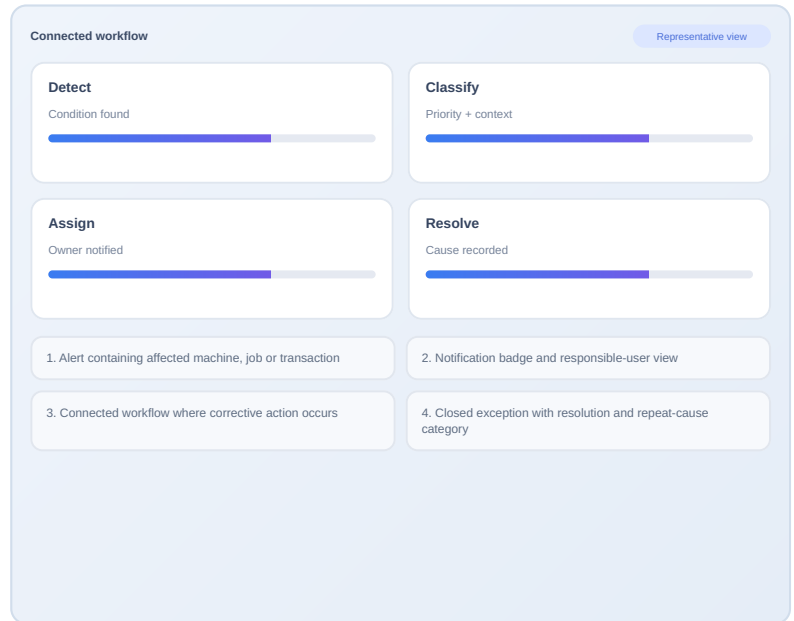
MEASURE	EVIDENCE SOURCE	CREDIBLE REPORTING
Actionable alert rate	Alert sample review	Percentage requiring a real user action rather than informational noise.
Time to acknowledgement	Alert timestamps	Median time from creation to responsible-user acknowledgement.
Time to resolution	Exception register	Median time from detection to verified closure by severity.
Unowned exception rate	Assignment audit	Alerts without a responsible person or role.
Duplicate alert rate	Deduplication review	Repeated alerts representing the same underlying condition.
Repeat cause frequency	Resolution categories	Exceptions recurring from the same workflow or technical cause.

"Operational visibility is not the number of alerts generated; it is the speed with which the right exception reaches an accountable person and is resolved."

MAKE THE STORY TANGIBLE

Show one exception from detection through ownership and verified closure

Use a real blocked or overdue condition and show its source record, severity, recipient, acknowledgement, corrective action and closure. Include the dashboard or report where recurring causes are reviewed.



Recommended evidence-building sequence

Phase 1 - Exception register

Record current blocked work, missed alerts and recurring failure categories.

Phase 2 - Priority rules

Apply ownership, severity, due-time and deduplication standards to selected alert types.

Phase 3 - Verified result

Report actionable rate, acknowledgement, resolution, duplication and repeat-cause measures.

Build the publishable proof layer

Replace representative views with dated interface screenshots, a defined baseline, system-derived measures and an approved participant quotation.

corexa.com.au
sales@corexa.com.au

Publication note: capability descriptions are based on the Corexa repository supplied in July 2026. Confirm the deployed workflow, permissions, integrations and data quality before presenting a capability as a completed customer outcome.