Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

From Agentic AI to Structured Workflows: Designing Process Architectures for AI-Augmented Administrative Decision Support

M. Thirukkumaran
Government Technology Agency
Elston Foo
elston_foo@enterprisesg.gov.sg
Enterprise Singapore
Ee Chuan Chang
Government Technology Agency
Bei Yin Koh
Enterprise Singapore
Kaezel Cho
Enterprise Singapore
Yu Kai Chong
Government Technology Agency

Research Article

Keywords: Business Process Management, AI-Augmented Workflows, Decision Support, Workflow Architecture, Human-in-the-Loop, Action Design Research

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11541366/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

From Agentic AI to Structured Workflows: Designing Process Architectures for AI-Augmented Administrative Decision Support

M. Thirukkumaran[1], Elston Foo[2*], Ee Chuan Chang[1], Bei Yin Koh[2], Kaezel Cho[2], Yu Kai Chong[1]

[1]Government Technology Agency, Singapore.
[2*]Enterprise Singapore, Singapore.

*Corresponding author(s). E-mail(s): elston_foo@enterprisesg.gov.sg;

Abstract

Public sector agencies increasingly seek to use artificial intelligence in complex administrative workflows, yet guidance on process architecture for AI integration remains limited. We present an action design research study of an AI-augmented grant evaluation process in a Singapore enterprise development agency. Across three architectural iterations, we moved from an agentic AI approach, through a layered extraction-evaluation pipeline, to a workflow-based design that models grant evaluation as a directed acyclic graph of typed process nodes (data ingestion, AI processing, and deterministic computation) with mandatory human review checkpoints. Across the iterations, we observed improvements in consistency, debuggability, and governance compatibility. In a three-month pilot with 11 officers and 123 cases, 80 of the 103 outputs produced were accepted (77.7%), and five officers self-reported a 21% reduction in processing time per application. Seven months later, a census of the nine remaining officers found reported routine use and the lowest ratings for judgement-adjacent outputs; none of the six officers who answered both surveys lowered any of four repeated ratings, and officers reported adapting their practices in different ways (process co-evolution). We derive three design principles for AI-augmented business processes: (1) decompose complex processes into typed, bounded workflow nodes rather than delegating end-to-end reasoning to AI agents; (2) separate AI-suitable tasks from deterministic computation and human judgment within the process model; and (3) design conservative failure modes that route exceptions to human handlers. We also argue that research on structured language-model agents supports, rather than refutes, explicit process structure.

Keywords: Business Process Management, AI-Augmented Workflows, Decision Support, Workflow Architecture, Human-in-the-Loop, Action Design Research

1 Introduction

Administrative business processes, particularly in the public sector, are highly complex, document-intensive, and depend on professional judgment (Dumas et al. 2018). Grant evaluation, a critical function of economic development agencies, exemplifies this: each application requires officers to review eligibility criteria, assess policy alignment, evaluate financial information, and synthesise evidence from heterogeneous documents spanning hundreds of pages (Mkabane and Kinigi 2024).

Advances in artificial intelligence, particularly large language models (LLMs), have drawn significant interest in augmenting these processes. BPM research on AI integration has grown rapidly, covering predictive process monitoring (Teinemaa et al. 2019), automated process discovery (van der Aalst 2016), and more recently, LLMs for BPM tasks (Vidgof et al. 2023). However, much of this work focuses on AI applied *to* process models (e.g., for analysis or prediction) rather than AI integrated *within* process execution as a first-class architectural component.

One particularly active research area is *agentic AI*, where language models are given broad autonomy to reason about and execute tasks with minimal structural constraints. Agentic approaches show promise for open-ended tasks. Whether they suit structured administrative processes, where accountability, reproducibility and policy compliance are paramount, remains an open question for BPM research.

We address this gap through an action design research study (Sein et al. 2011) of an AI-augmented grant evaluation process in a Singapore public sector agency. Facing a near-doubling of expected processing throughput following organisational restructuring, the agency explored AI integration to support officers in evaluating complex grant applications. The design work showed that agentic AI approaches produced inconsistent results for structured administrative evaluation, prompting a shift to a workflow-based architecture that models the evaluation process as a directed acyclic graph (DAG) of typed nodes.

We address four BPM research questions:

RQ1 How should complex administrative processes be modeled to integrate AI effectively?

RQ2 What are the design trade-offs between agentic and workflow-based approaches for structured process automation?

RQ3 How can AI-augmented process designs implement human-in-the-loop principles?

RQ4 How do officers' use and assessment of an AI-augmented process develop during sustained post-adoption use, and how does the process as officers carry it out co-evolve with the process as designed?

What is new in this article. This article extends our BPMDS 2026 conference paper (Thirukkumaran et al. 2026a), which addressed RQ1–RQ3 using the pilot evaluation (T0). It makes two substantive additions: (i) a longitudinal evaluation based on a follow-up survey of all nine officers who remained on the team seven months after the pilot survey, with the four experience items repeated verbatim from T0 and compared within officer for the six who answered both waves (Section 7); and (ii) an analysis of *process co-evolution*, how officers report adapting their work around the system, based on one round of short free-text answers and interpreted through structurational

and trust-in-automation research (Sections 2 and 8.4). It also adds (iii) a discussion that places the original agentic-AI finding in the agent practice of its time and sets out the trade-offs of workflow structure for regulated processes (Section 8.3); and two supporting methodological additions: (iv) an observation that self-reported time savings are not comparable across survey waves when the counterfactual baseline shifts (Section 3); and (v) an expanded methodology and a rewritten limitations section.

Relation to prior publications. The case context, system design and T0 pilot results in Sections 4–6 were first reported in the BPMDS 2026 paper (Thirukkumaran et al. 2026a). The same pilot, including its T0 results, was also reported from a digital-government perspective in a dg.o 2026 paper by an overlapping author team (Thirukkumaran et al. 2026b). The T1 evaluation, the co-evolution analysis and the agentic-harness discussion are new to this article and do not appear in either earlier paper.

2 Background and Related Work

2.1 AI Integration in Business Process Management

Research on AI in BPM covers process mining (van der Aalst 2016), predictive process monitoring (Teinemaa et al. 2019), and prescriptive process monitoring (Weinzierl et al. 2020). Large language models have opened new research directions: Vidgof et al. (2023) examined LLM applications across BPM tasks, Busch et al. (2023) argued for prompt engineering as a way to bring LLM capabilities to BPM research, and Berti et al. (2024) explored LLM applications in process mining. These studies largely position LLMs as analytical tools applied to process artifacts rather than as components of process execution.

In administrative and knowledge-intensive processes, decisions require contextual interpretation and professional judgment, making full automation both technically difficult and normatively problematic (De-Arteaga et al. 2020). Recent work advocates decision support that combines human expertise with computational assistance (Wagner et al. 2025; Grgić-Hlača et al. 2022). However, practical architectural guidance for embedding LLMs in operational business processes remains limited.

More broadly, Dumas et al. (2023) articulate a vision for *AI-Augmented Business Process Management Systems* (ABPMSs) in which AI components operate within explicitly defined *process frames*: boundary-setting structures that constrain AI autonomy while permitting adaptive behaviour within those boundaries. This *framed autonomy*, where AI acts independently within a governed scope, has become a central design challenge for AI-augmented processes. Acitelli et al. (2025) propose automated planning techniques to achieve framed autonomy, encoding procedural and declarative process constraints into formal automata so that recommended AI-driven process continuations minimise the cost of violating the constraints defined by the process frame. In complementary work, Janiesch et al. (2025) introduce *process autonomization*, emphasising processes that adapt with varying degrees of human oversight. Our operational deployment provides empirical evidence of both the risks of unframed AI autonomy and the benefits of architecturally bounded integration.

2.2 Process Modeling and Workflow Architectures

Workflow management systems operationalise process models by coordinating task execution, resource allocation, and information flow according to defined process logic (van der Aalst and ter Hofstede 2005; Dumas et al. 2018). Directed acyclic graphs (DAGs) are widely used as a computational abstraction for workflow orchestration (Deelman et al. 2009), modelling processes as graphs of typed tasks with explicit dependencies that enable parallel execution, failure handling, and auditability. The workflow patterns literature (Russell et al. 2006a,b) provides a vocabulary for process architectures, including patterns for conditional branching, parallel execution, and exception handling that require careful consideration when workflow nodes involve non-deterministic AI components.

2.3 Human-Centric BPM and Decision Support

Human-centric BPM emphasises the role of human judgment, creativity, and interaction in business processes (Rosemann 2014). In knowledge-intensive processes, human decision-making cannot be fully codified; process designs should support and augment human expertise (Reichert and Weber 2012). The concept of human-in-the-loop (HITL) has become prominent in AI systems design, describing architectures where humans maintain oversight and final decision authority (Wagner et al. 2025). HITL principles align with established BPM human task assignment patterns, but AI-augmented processes require new ways to divide labour between automated processing and human judgment (Grgić-Hlača et al. 2022; Power 2002), particularly when AI outputs are probabilistic rather than deterministic.

2.4 Post-Adoption Use and Process Co-evolution

Research on technology in organisations has long held that what a system does in practice is not settled at design time. Barley (1986) showed that identical CT scanners were followed by different forms of organisation in two radiology departments and argued that technology is better understood as an occasion for structuring. Adaptive structuration theory examines why the organisational changes expected from advanced information technologies often do not occur, or occur inconsistently; it distinguishes the structures a technology provides from the structures that emerge as people interact with it (DeSanctis and Poole 1994). Orlikowski (2000) develops a practice lens in which people enact structures through ongoing work that shape their emergent and situated technology use. Leonardi (2011) addresses when people change their routines and when they change the technology itself, suggesting that perceived constraints lead to changes in the technology and perceived affordances to changes in routines.

A second line of work concerns how people rely on automation. Lee and See (2004) argue that automation is often problematic because people fail to rely on it appropriately, and that trust guides reliance particularly when complexity prevents a full understanding of the automation; they frame the design goal as *appropriate reliance*. Parasuraman and Riley (1997) distinguish use, misuse (over-reliance), disuse (neglect or under-use) and abuse of automation. In a simulated flight task, participants working with a highly but not perfectly reliable automated aid missed events the aid did

not prompt them about and followed its recommendations even when other valid indicators contradicted them, which the authors discuss as automation bias (Skitka et al. 1999).

Both matter for AI-augmented BPM because the ABPMS agenda bounds AI autonomy through constraints specified for the process (Dumas et al. 2023; Acitelli et al. 2025; Janiesch et al. 2025), whereas structurational research expects users to shape how any such bound is enacted in practice. We apply both perspectives to the longitudinal data reported in Section 7 and discussed in Section 8.4.

3 Research Methodology

This study uses Action Design Research (ADR) (Sein et al. 2011) as its primary method. ADR is appropriate because the researchers worked within the organisation and iteratively built, intervened with, and evaluated the artifact in actual use. The artifact did not pre-exist; the researchers built and refined it, so a purely observational case study framing does not fit. The authors belonged to the cross-agency project team spanning the Government Technology Agency and Enterprise Singapore, and directly helped with (i) defining system requirements through stakeholder workshops, (ii) designing and implementing the workflow architecture across all three iterations, and (iii) analysing pilot outcomes and refining the system in response. Operational needs and research-driven design decisions jointly shaped the system. The three iterations (Sections 5.1–5.3) correspond to ADR's building, intervention, and evaluation (BIE) cycles (Sein et al. 2011), while deriving generalisable design principles (Section 8) corresponds to ADR's formalisation of learning stage, drawing on design science conventions (Hevner et al. 2004; Peffers et al. 2012) for stating design knowledge.

The research followed four phases: (1) *process analysis* through workshops with officers to map the evaluation process and codify decision criteria; (2) *iterative design* of architectures tested against past cases; (3) *pilot deployment* with 11 officers and 123 live grant cases over three months; and (4) *analysis and reflection*, triangulating system logs, officer issue reports reviewed by a multidisciplinary adjudication committee, and post-pilot surveys.

Follow-up evaluation (T1).

To address RQ4, we conducted a second survey wave (T1) on 25 August 2026, seven months after the T0 survey in late January 2026, when the system was in routine operational use. The T1 instrument had 36 drafted items, with 35 administered in five sections: usage and time per case; output quality, including the usefulness of individual system features; trust and working practices; overall assessment; and three open-ended questions. Most items used five-point scales with labelled anchors. Time per case was requested as free-text hours for standard cases, the case type covered by the T0 time survey, both with the system and, hypothetically, without it. We coded these answers by taking range midpoints, converting minutes to hours and counting eight working hours per day; excluding the one answer given in days lowers the T1 reduction from 43% to 40%. Four experience items were repeated word for word from

the T0 survey. An item asking whether officers would recommend continued system use was dropped before administration because another already asked about intention to continue. That intention item, one of the four repeated T0 items, is the closest repeated measure, although the two items are not equivalent.

Nine of the eleven pilot officers remained on the team at T1; all nine responded to the survey, so T1 is a census of the remaining pilot officers rather than a sample. Because the population is small, we report descriptive statistics, with response distributions for the key items, and do not apply inferential tests; item-level responses are in the published dataset, with a few items withheld and passages redacted (see Declarations), and standard deviations are sample SDs. The instrument recorded respondents' names and work e-mail addresses. For analysis, we replaced names with initials and removed e-mail addresses. In this article and in the published dataset, officers are identified only by participant codes (P1–P10) assigned in random order, to report quotations anonymously; each officer has the same code in both waves. Free-text answers such as "NA" or "none" were treated as no answer.

The full T0 export identifies respondents by initials, the convention used for the T1 file, so officers who answered both waves could be linked before codes were assigned. Seven officers answered the T0 experience items and five the T0 time items. Six of the seven T0 experience respondents also answered at T1; the seventh was no longer on the team, and three T1 respondents had not answered the T0 experience items. For the four repeated items we therefore report both the unpaired group comparison (seven against nine respondents) and a paired comparison for the six matched officers. We report the paired comparison descriptively because the matched group is too small for a meaningful significance test (Section 7.2). We confirmed that matching initials across files denote the same officer. In the T0 time survey, officers could mark a section as not applicable; we excluded those sections from that officer's before and after totals (three entries from two officers). Because time questions differ between waves, we report savings within each wave only. The open-ended descriptions of changed practice were coded into adaptation patterns (Section 8.4).

4 Case Context and Process Analysis

4.1 Organisational Setting

The case organisation is a national enterprise development agency in Singapore that supports domestic businesses through grants, capability development programmes, and advisory services. A flagship grant scheme supports a broad range of enterprise projects, including productivity improvements, innovation efforts, and internationalisation activities.

In April 2025, grant processing operations were centralised into a newly created division to improve standardisation and throughput. After restructuring, each processing officer would need to handle approximately 196 cases per year, nearly doubling prior expectations. Existing source systems were scheduled for replacement only in the medium term, limiting deep system-level automation and requiring an interim solution alongside legacy infrastructure.

Grant Evaluation Process (Simplified BPMN Representation)

(Start)	Application received
Task	Document receipt and categorisation
Task	Eligibility screening
XOR	Eligible? No → Rejection Yes ↓
AND-split	*Parallel evaluation branches:*
	Policy alignment assessment [judgment]
	Financial analysis and ratio computation [computation]
	Cost structure and vendor review [extraction]
AND-join	Synchronise all branch results
Task	Evidence synthesis; Officer review and recommendation
(End)	Decision recorded

Fig. 1 BPMN-inspired representation of the grant evaluation process. XOR denotes exclusive choice; AND denotes parallel split/synchronisation. Task annotations informed the typed node architecture of Section 5.3.

4.2 Grant Evaluation Process

Grant processing is operationally intensive. Each application requires officers to review eligibility, assess alignment with policy intent, evaluate financial information, and synthesise evidence from multiple supporting documents. A process review indicated that an application could require up to 20 person-hours, with the processing officer doing most of the analysis: reviewing heterogeneous documents and seeking clarification from applicants.

Each application typically comprises a primary electronic submission and multiple supporting documents, often exceeding hundreds of pages and spanning heterogeneous formats such as PDFs, spreadsheets, and scanned images.

Fig. 1 presents a simplified BPMN-inspired model of the evaluation process derived from collaborative workshops with officers. The model shows two features critical to system design: an exclusive choice gateway governing eligibility (determining whether the application proceeds or is rejected) and a parallel split gateway dispatching independent evaluation branches that must synchronise before synthesis and final review.

4.3 Process Characteristics and Constraints

Several grant evaluation characteristics complicated automation and shaped the design:

- *Heterogeneous information sources*: Required information was distributed across multiple documents in varied formats, with no standardised structure across applications.
- *Variable document quality*: Financial statements and supporting documents varied widely in format, completeness, and legibility. Some documents contained scanned images of varying quality.

Table 1 Evolution of process architectures across three design iterations.

Dimension	V1: Agentic	V2: Layered	V3: DAG-Based
Process model	Single unbounded task	Three sequential layers	Typed node graph
AI scope	Unconstrained reasoning	Per-layer extraction	Per-node, bounded
Control flow	None (single prompt)	Fixed linear pipeline	Explicit DAG
Conditional logic	Implicit in prompt	Not supported	Native branching
Debuggability	Opaque	Layer-level visibility	Node-level visibility

- *Conditional evaluation logic*: Evaluation steps depended on project scope, sector, and policy intent, with additional checks triggered by earlier findings, creating branching process paths.
- *Professional judgment requirements*: Officers exercised discretion in borderline cases, particularly regarding policy alignment and strategic fit, which could not be fully codified into deterministic rules.

These characteristics map to well-known challenges in knowledge-intensive BPM (Reichert and Weber 2012): process variability, information heterogeneity, and the need to balance structure with flexibility.

To address the infrastructure constraint, officers manually download relevant files and upload them to the AI system via categorised input buckets (e.g., proposal, financial statements, vendor quotations). This added approximately three minutes of manual effort per case but eliminated months of integration development, showing that retaining manual steps sometimes benefits constrained process designs.

5 System Design: Iterative Architecture Evolution

The system, called *Granta* (survey items and officer quotations keep the form's spelling, GRANTA), underwent three major architectural iterations, each informed by the limitations of its predecessor. These iterations, summarised in Table 1, provide empirical evidence of how process architecture affects the reliability of AI-augmented workflows.

5.1 Version 1: Agentic Approach

The initial design treated grant evaluation as an open-ended reasoning task for an AI agent. Application documents and evaluation criteria were provided in a single prompt to o4-mini, an OpenAI reasoning model, without tool access or multi-step control, and the model was asked to produce a comprehensive assessment with minimal structural constraints.

The agentic approach had four fundamental limitations for structured administrative processes:

1. *Non-determinism and error cascading*: Given identical inputs, the agent produced substantively different outputs across runs. Errors in intermediate reasoning cascaded into subsequent assessments, with output variability growing as requirements scaled.
2. *Computational unreliability*: The agent produced incorrect results for numerical calculations (e.g., financial ratios), particularly multi-step computations, confirming that LLMs should not be relied upon for deterministic arithmetic within a business process.
3. *Opaque reasoning*: The agent's evaluation logic could not be inspected or selectively corrected. Explanations were unreliable because the underlying extraction itself could be wrong, making diagnosis impossible.
4. *Cost inefficiency*: The agent spent significant resources reasoning about *which* steps to perform, although policy predetermined those steps.

Quantitative tests on previously approved grant applications confirmed these limitations. Given the same approved application, the agentic approach produced substantively different recommendations across ten independent runs: approximately 30% of runs recommended rejection instead of approval, and several runs generated unsanctioned categories (e.g., "conditionally approved") not defined in the evaluation framework. Output wording varied entirely across every run, and numerical values were frequently misread from source documents or miscalculated, particularly for multi-step financial computations.

In BPM terms, the agentic approach collapsed the entire evaluation process into a single, unbounded task with no internal structure, no intermediate checkpoints, and no separation between AI-suitable and human-suitable activities.

5.2 Version 2: Layered Extraction-Evaluation Architecture

The agentic approach's limitations prompted decomposition into three sequential layers: (1) information extraction from source documents, (2) intermediate processing and evaluation, and (3) final assessment synthesis. Extraction steps were specified declaratively in a configuration file, and the engine executed all extraction calls in parallel, improving both transparency and accuracy relative to V1.

However, process mapping with officers revealed interleaving extraction and evaluation steps with conditional logic (e.g., "if condition X holds, extract Y from document Z") that the linear model could not represent. Process specification was also fragmented across configuration files and application code. Version 2 substantially improved extraction determinism and accuracy over V1, but partial decomposition was insufficient for workflows with branching paths and conditional dependencies.

5.3 Version 3: DAG-Based Workflow Architecture

Version 2 showed that the grant evaluation process is best represented as a directed graph rather than a linear pipeline. This aligned with established BPM principles: complex work should be decomposed into well-defined tasks with explicit control flow, typed inputs and outputs, and clear dependency management (Dumas et al. 2018; Russell et al. 2006b).

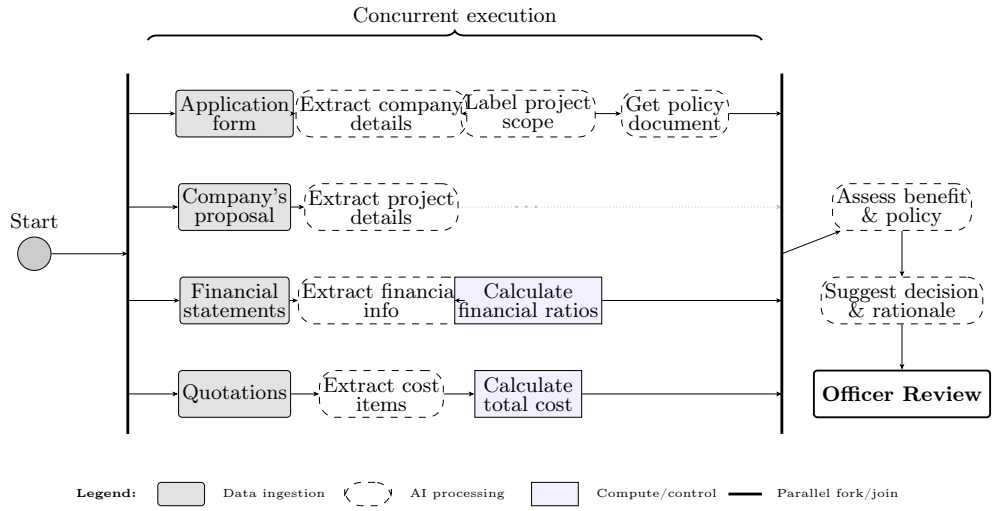


Fig. 2 DAG-based workflow architecture for *Granta*. Vertical bars denote parallel fork/join; four branches execute concurrently per document type. Node shapes distinguish data ingestion, AI processing, and deterministic computation. Only representative nodes shown; the full workflow comprises over 50 nodes.

The final architecture models grant evaluation as a directed acyclic graph (DAG) of typed process nodes (Fig. 2). Each node performs a specific, bounded function, and nodes execute asynchronously as their input dependencies are satisfied. The full evaluation workflow has over 50 nodes, including extracting company background, generating project summaries, analysing cost structures, and computing financial ratios.

The architecture has three primary node types:

Data ingestion nodes standardise heterogeneous input documents into a common internal representation through format conversion, optical character recognition, and structural parsing, isolating document handling from downstream processing logic.

AI processing nodes apply large language models to extract, summarise, or synthesise information, each operating within a bounded scope defined by its prompt specification and input schema. By constraining AI to specific tasks, the architecture preserves LLM strengths (natural language understanding, information synthesis) while mitigating weaknesses (non-determinism, scope drift, opaque reasoning).

Compute and control nodes execute deterministic logic, perform calculations (e.g., financial ratio analysis, eligibility checks), and manage conditional branching. These nodes execute policy-defined computations reproducibly and determine evaluation paths through explicit rules rather than model interpretation.

The DAG architecture instantiates several workflow control-flow patterns (Russell et al. 2006b): parallel split and synchronisation for concurrent branch execution (as shown in Fig. 1), exclusive choice for conditional evaluation paths such as sector-specific checks, and an extended cancellation pattern where any node failure withholds all outputs to address AI-specific failure modes where errors may be plausible rather than explicit. The three node types extend conventional workflow task patterns by

making each step's determinism an explicit, first-class concern, distinguishing between deterministic tasks amenable to conventional verification and non-deterministic AI tasks requiring statistical quality assurance.

This typed architecture has several BPM-relevant properties: independent nodes execute concurrently (reducing end-to-end processing to approximately two minutes per case); AI is used selectively where it helps while deterministic tasks use conventional computation; failures are isolated to specific nodes, enabling targeted diagnosis; and each node produces inspectable intermediate outputs for auditability.

System changes between T0 and T1. The bugs behind the pilot's technical failures (Section 6) had been fixed by the end of the pilot, after which the system ran much more reliably. No new features were added between the T0 and T1 surveys, so apart from these fixes the system evaluated at T1 was largely the one piloted. We have no release records beyond this account.

5.4 Human-in-the-Loop Process Design

The workflow implements human-in-the-loop principles at multiple levels.

At the *process initiation* level, officers select and upload relevant documents, judging which materials matter. At the *output review* level, all system outputs are presented as decision support artifacts; officers verify extracted information against source documents and apply professional judgment before formulating recommendations.

The system uses a *conservative failure mode*: any workflow node failure withholds all final outputs rather than presenting partial results. Stakeholders initially requested that completed node results be shown regardless, but we judged that officers would likely rely on partial outputs without thorough verification, particularly under time pressure. Withholding all outputs upon any failure forces fully manual review, which is intended to prevent automation bias arising from incomplete analyses. There are three distinct situations. A *detected node failure*, such as a parsing or computation error, withholds all outputs. A *missing required upload* is caught earlier: the run stops at the input check with an error and does not start the processing nodes. A *plausible but incorrect output* from a node that completes without error is caught by neither mechanism; only officer verification at the output-review step can detect it.

On *decision authority*, the system makes no autonomous decisions. The system handles document ingestion, information extraction, and deterministic computation; officers select relevant documents, verify extracted information against sources, interpret results with professional judgment, and retain full decision authority and accountability.

5.5 Infrastructure and Deployment

Granta was deployed on government-approved cloud infrastructure. Each workflow runs in an isolated container and stores intermediate node results in a secure relational database for auditability. Execution is asynchronous: officers submit a workflow and return later to review results.

Table 2 Technical failure categories during the pilot (n=20 of 123 cases).

Failure Category	Cases
LLM content filter false positives	13
Document parsing bug (dropped pages during PDF extraction)	3
Computation failure from null inputs	2
File handling bug (Excel upload processing)	1
Potential client-side connection failure	1
Total	**20**

6 Pilot Evaluation and Results (T0)

6.1 Pilot Setup

A structured three-month pilot began in November 2025 with 11 processing officers and 123 unique grant cases drawn from live operations. Officers were briefed on system functionality and instructed to use it strictly as a decision support tool, verifying all outputs against source documents.

Evaluation data were collected through system-level failure logging, structured officer issue reports reviewed by a multidisciplinary adjudication committee (business implementation leads, lead developer, and product manager), and post-pilot surveys measuring self-reported processing time and experience.

6.2 Output Acceptance and Failure Analysis

Of 123 cases processed, 20 (16.3%) failed to produce outputs due to technical issues. Table 2 categorises these failures.

The dominant failure mode, LLM content filter false positives (n=13), arose when document batching to meet API input constraints inadvertently triggered the cloud provider's content safety filters. Refining batching logic resolved this issue, which BPM practitioners should consider when designing workflows with external AI services.

The 103 cases that produced outputs underwent officer review and adjudication. Initially, 74 were reported as satisfactory and 29 as unsatisfactory. After adjudication committee review, 6 were reclassified as non-system issues (UI imperfections, out-of-scope expectations, and user errors), yielding a final acceptance rate of 80 of 103 outputs (**77.7%**). Initial officer acceptance, before adjudication, was 74 of 103 (71.8%), and the 80 accepted outputs amount to 65.0% of all 123 cases.

The remaining 23 unsatisfactory cases involved suboptimal workflow logic or extraction errors from model misrecognition in complex documents. These issues traced to specific, fixable system components, not to fundamental architectural flaws. This validates the granular failure handling of the workflow-based architecture.

6.3 Time Savings

Five of the 11 officers answered an early post-pilot survey. Their mean estimated processing time per application was 7.54 hours ($SD = 2.88$) before the system and 5.98

Table 3 Officer-reported experience scores at T0 (n=7; 5-point Likert scale).

Dimension	Mean	SD	Interpretation
Overall experience	3.7	0.49	Between "acceptable" and "good"
Impact on efficiency	3.9	0.38	Approaching "faster"
Quality of evaluations	3.6	0.53	Between "no change" and "better"
Intention to continue use	4.3	0.76	Between "probably yes" and "definitely yes"

hours ($SD = 2.50$) with it, about **21%** less. These SDs are sample SDs, as throughout this article; the conference version reported population SDs (2.58 and 2.24 hours) for the same two means. In the per-section estimates, savings were concentrated in understanding the project scope and objectives, the project objective and evaluation summary, and the company background, whereas no saving was reported for evaluating the supportability of cost items, assessing cost reasonableness against benchmarks, enhanced checks, key deliverables or shareholding assessment. Officers reported that edge cases needing their judgement, where policy gave no clear guidance, still took as long as before. This is consistent with the system's design as process support rather than decision replacement.

6.4 Officer Experience

Experience surveys (7 of 11 officers responding) assessed four dimensions on five-point Likert scales (e.g., 1="very poor" to 5="excellent" for overall experience; 1="definitely no" to 5="definitely yes" for continued use intent), complemented by open-ended questions and detailed per-task time estimates. Table 3 summarises the results.

Strong adoption intent (mean=4.3) and perceived efficiency gains (mean=3.9) coexisted with more modest quality improvement ratings (mean=3.6). This asymmetry aligns with the system's design as process support: officers valued reduced cognitive load while rating the quality of automated outputs more cautiously. These ratings alone do not show whether officers critically verified the outputs.

In open-ended feedback, officers valued automated summarisation for complex cases, while suggesting improvements to workflow flexibility (e.g., allowing partial execution when documents are missing).

6.5 Costs Relative to Reported Benefits

Building Granta from Version 1 to Version 3 in production took about six months from May 2025, mainly one full-time engineer supported by a UX designer and a second engineer for front-end work. Each evaluation run costs approximately USD 0.50–1.50 in language-model calls plus USD 0.10–0.50 in cloud infrastructure, depending on run duration: USD 0.60–2.00 in total. Officers run the workflow at least twice per case, so on a two-run baseline a case costs USD 1.20–4.00. At 300–400 runs a month, this comes to USD 180–800 a month, or USD 2,160–9,600 a year at unchanged volume. These figures exclude our own time for maintenance and operation, runs beyond two per case, officers' rework on the 23 outputs judged unsatisfactory at T0, the roughly three minutes of manual upload per case (Section 4) and the time officers spend

Table 4 Verbatim-repeated experience items, T0 versus T1 (5-point scales; unpaired groups). T0 and T1 values are recomputed from the anonymised responses; the T0 values agree with Table 3.

	T0 (n=7)		T1 (n=9)		
Item	**Mean**	**SD**	**Mean**	**SD**	**T1 distribution**
Overall experience	3.71	0.49	4.22	0.67	1 acceptable, 5 good, 3 excellent
Impact on time spent	3.86	0.38	4.00	0.50	1 same, 7 faster, 1 much faster
Quality/consistency	3.57	0.53	4.11	0.60	1 no change, 6 better, 2 much better
Intention to continue	4.29	0.76	4.89	0.33	1 probably yes, 8 definitely yes

checking outputs. The evidence of benefit is the 21% reduction in time per application self-reported by five officers at T0, the acceptance of 80 of 103 outputs (65.0% of all 123 cases) and the routine use reported at T1 (Section 7.2). Because the time savings are self-reported, we do not convert them into a net monetary benefit or a break-even estimate.

7 Longitudinal Evaluation at Seven Months (T1)

The follow-up survey (T1) tests whether the pilot findings of Section 6 held once the system was in routine use.

7.1 Design and Respondents

All nine officers who remained from the eleven in the pilot responded to the T1 survey (Section 3). Four experience items were repeated word for word from T0: overall experience, impact on time, quality or consistency of evaluations, and intention to keep using the tool. Seven respondents answered them at T0 and nine at T1, and six officers answered both, so we compare the two respondent groups and the six matched officers on these items.

7.2 Sustained Use and Embeddedness

Officers have made the system part of their regular work. Agreement that it "has become part of my regular grant evaluation workflow" averaged 4.56 (SD 0.73), and the same mean was recorded for "if GRANTA were no longer available, my grant evaluation work would take more time"; eight of nine officers agreed with each statement. Five officers reported using the system "always (e.g. almost daily)", three "often" and one "sometimes"; none reported rare use. Four officers used it for 81–100% of their cases, two for 61–80% and three for 41–60%. All nine reported that the productivity benefit had increased since they first started using the system (three "significantly", six "somewhat").

Officers rated the system more favourably after seven months of routine use. All four repeated items rose (Table 4): overall experience by +0.51, quality by +0.54 and intention to continue by +0.60, with eight of nine officers saying they would definitely keep using it. Perceived impact on time rose least (+0.14).

Table 5 Repeated experience items for the six officers who answered both waves (matched across waves; 5-point scales). Changes are T1 minus T0 for the same officer.

Item	T0 mean	T1 mean	Mean change	Higher / same / lower
Overall experience	3.83	4.50	+0.67	3 / 3 / 0
Impact on time spent	3.83	4.00	+0.17	1 / 5 / 0
Quality/consistency	3.67	4.00	+0.33	2 / 4 / 0
Intention to continue	4.50	5.00	+0.50	3 / 3 / 0

Table 6 Self-reported time per case. The T1 rows are internally comparable; the T0 row is shown for reference only and is not comparable with them.

Wave and unit	n	With system, h	Without, h	Reduction
T1, per standard case	9	3.58 (2.33)	6.33 (4.25)	43%
excl. one entry in days	8	3.09 (1.93)	5.13 (2.37)	40%
T0, per application	5	5.98 (2.50)	7.54 (2.88)	21%

Values are mean (SD). Reduction = 1 − (mean with / mean without). T1 "without" times are hypothetical estimates; the T0 baseline is a recalled pre-adoption time.

The improvement holds for the same officers over time. Among the six who answered both surveys, none rated any item lower: of 24 paired ratings, nine rose and fifteen held (Table 5), and all six would definitely continue using the tool. The paired gain was larger than the group gain for overall experience (+0.67 against +0.51), similar for time and intention to continue, and smaller for quality (+0.33 against +0.54). Overall, officers rated Granta more favourably after seven months.

7.3 Time per Case

Officers estimate that Granta cuts their time on a standard case by 43%, from 6.33 to 3.58 hours (Table 6). All nine estimated a shorter time with the system and agreed that it had reduced their evaluation time (mean 4.67, SD 0.50). Overall, Granta saves officers well over a third of their time on each case.

The savings fall on the core tasks of evaluation. Officers most often named identifying key information and drafting the assessment (8 of 9 each), followed by computations and cost tables (5 of 9). Overall, Granta saves time on the work that defines an evaluation.

7.4 Output Quality by Node Type

Officers rated the system highest where AI extracts and summarises, and lowest where its outputs approach judgement (Table 7). Summaries of company background, objectives and impact received the highest and most uniform usefulness ratings (4.89, SD 0.33; no rating below 4), and all nine officers rated completeness of extracted information 4. Project classification averaged 3.33 with the widest spread (SD 1.22, range 1–5), and the appropriateness of generated rationale was the lowest and most variable

Table 7 T1 feature usefulness and output quality (n=9; 5-point scales), ordered within each block by mean.

Item	Main node type(s)	Mean	SD	Range
Feature usefulness (1 = not useful at all, 5 = extremely useful)				
Summarise company background, objectives, impact	AI processing	4.89	0.33	4–5
Cost-Benefit Assessment (CBA) and grant amount calculation	Compute + AI	4.11	0.60	3–5
Tabulation of cost table	AI + compute	3.89	1.17	2–5
Financial capability assessment	Compute	3.89	0.93	2–5
Project classification	AI (judgement-adjacent)	3.33	1.22	1–5
Output quality (1 = very poor, 5 = excellent)				
Clarity and usefulness of summaries	AI processing	4.33	0.50	4–5
Accuracy of extracted information	Ingestion + AI	4.00	0.50	3–5
Completeness of extracted information	Ingestion + AI	4.00	0.00	4–4
Appropriateness of generated rationale	AI (judgement-adjacent)	3.44	1.13	1–5

quality dimension (3.44, SD 1.13, range 1–5). Overall, officers value the system most for the tasks our design assigns to AI.

Officers placed the system's limits at the point of judgement. Asked for its single biggest limitation, one officer wrote that "GRANTA is not good at assessing the supportability of project scope, which often requires human judgment" (P5). Another named the "[d]etermination of the project supportability" (P2), and a third wrote that the "[f]inal recommendation may not be accurate" (P6). Overall, officers see judgement as their own work.

Judgement-adjacent outputs remained the lowest rated, as at T0 (Section 6), even as perceived quality rose. No features were added between the waves, so the rise likely reflects fewer failures after the pilot's bugs were fixed and officers' growing familiarity with the system. Overall, perceived quality improved as the system stabilised.

7.5 Trust and Verification

Officers trust the system's output as an input to their assessment and keep the judgement for themselves. Agreement that "I generally trust GRANTA's outputs as a useful input to my assessment" averaged 3.89 (SD 0.78). Officers rated their ability to identify when output needs further verification or human judgement at 4.00 (SD 0.87) and the system's support for their professional judgement at 3.89 (SD 0.93), although four of nine were neutral on the last item, so officers differ on where the system's role ends. Overall, officers use the output as material for their own assessment.

Officers keep checking and correcting what the system produces. Six said they "sometimes" need to significantly correct the output and three "rarely"; four "sometimes" disregard it entirely, four "rarely" and one "never", and no officer chose "often" or "always" for either question. This pattern of using the output as a starting point and correcting it where needed is consistent with *appropriate reliance* in the sense of

Lee and See (2004). Overall, officers rely on the system while keeping judgement in their own hands.

7.6 Workflow Adaptation

Officers have each fitted the system into their work in their own way. Agreement that "I've changed how I prepare or organise my cases to work better with GRANTA" averaged 3.67 (SD 1.00; five of nine agreeing), and agreement that "I use GRANTA differently now compared to the first month" averaged 3.22 (SD 0.97; four of nine agreeing). Overall, about half the officers report changing how they work.

The free-text answers show how far these ways of working diverge. One officer reads the proposal first and uses the output as a check: "I will read my proposal first and then use Granta to run through the documents so that I can have a sense of whether Granta has summarised correctly" (P3). Another reuses generated text: "I use the templatized remarks for the project assessment" (P8). A third drafts offline from the outputs: "I am able to draft an offline evaluation based on Granta's inputs which helps me to complete evaluation much faster" (P6). Others use the system selectively for summarisation and the Cost-Benefit Assessment (P5), or know which part of the output to refer to when completing each part of the agency's case system (P10), and one has learned "which parts are usually accurate information" (P2). Overall, officers have built different routines around the same system.

8 Discussion

Table 8 summarises each research question's answer, evidence and main limitation.

8.1 Process Architecture for AI Integration (RQ1)

We modelled grant evaluation as a directed acyclic graph (DAG) of bounded, typed nodes with explicit dependencies (Section 5.3). This applies BPM process decomposition (Dumas et al. 2018) to AI integration: each node has defined inputs and outputs, and a failure can be traced to the node that produced it. The explicit architecture is what makes each step of an AI-supported evaluation inspectable.

Typing makes determinism explicit in process design. We call this *determinism-aware task typing*: classifying each task as deterministic (compute and control), probabilistic (AI processing) or format-dependent (data ingestion, whose output quality depends on the format of the submitted documents). Principle 1 (Section 8.5) turns this classification into design guidance.

Process mapping with officers shaped both the graph and the scope of the project. The mapping revealed extraction and evaluation steps interleaved with conditional logic that the sequential layers of Version 2 could not represent, and this led us to the graph (Section 5.2). It also carried the work across the BPM lifecycle (Dumas et al. 2018), from process discovery through redesign and implementation to monitoring, where pilot failures traced to specific components guided our fixes (Section 6.2). The officers' knowledge of the process therefore set both the structure of the workflow and the extent of the project.

Table 8 Research questions, answers, evidence and main limitations.

RQ	Answer in brief	Evidence	Main limitation
RQ1	Model the process as a DAG of typed, bounded nodes (ingestion, AI processing, compute and control), with determinism made explicit (Principle 1)	Three design iterations; V3 architecture; T0 pilot (Sections 5, 6)	Single organisation and scheme
RQ2	We propose that workflow-based designs suit processes with codifiable logic, accountability requirements and low tolerance for output variability, and that agentic designs may suit exploratory tasks	V1 ten-run test; design observations across V1–V3; harness discussion (Section 8.3)	Versions not compared on a common case set; V1 not re-tested with current agent techniques
RQ3	Separate AI, deterministic and human responsibilities in the model; withhold outputs on detected failures; keep officer verification (Principles 2–3)	HITL design (Section 5.4); T0 and T1 ratings and reported correction behaviour	Effect on automation bias not measured; silent errors rely on officer review
RQ4	Reported routine use; no lower repeated rating among six matched officers; lowest ratings for judgement-adjacent outputs; officers report different ways of working around the system	T1 census (n=9); paired comparison (n=6); one round of free-text answers (Section 7)	Self-report; small groups; adaptations reported once, not observed over time

8.2 Process Structure and AI Integration (RQ2)

Making the process structure more explicit from V1 to V3 made the AI more bounded and inspectable. We moved from one unbounded task (V1) to three sequential layers (V2) and then to a typed node graph with conditional logic and explicit dependencies (V3; Table 1). At each step we observed more consistent outputs, easier debugging and a better fit with governance requirements, because intermediate outputs became visible and failures could be isolated to specific tasks.

We propose matching the AI integration architecture to the characteristics of the process. Workflow-based integration suits processes with codifiable evaluation logic, regulatory accountability requirements and low tolerance for output variability. Where a task within such a workflow needs iteration, a bounded agentic loop can supply it while process boundaries and officer control stay in place (Section 8.3). The proposal extends the spectrum from structured to unstructured processes (Reichert and Weber 2012) to the choice of AI integration architecture.

V3 gives operational form to framed autonomy (Dumas et al. 2023). We embedded the frame in the process architecture: node input schemas bound the scope of each AI task, and prompt specifications set its intended actions. Acitelli et al. (2025) derive the frame through formal automated planning; we encode it directly in specified nodes and their dependencies. The degree of process autonomization (Janiesch et al. 2025) thus becomes an explicit design choice, with the process model recording which tasks receive AI autonomy and which retain officer authority.

8.3 Structure in the Age of Agentic Harnesses

When the project began in May 2025, agent frameworks and tool-use practice were still maturing. ReAct (Yao et al. 2023) and Toolformer (Schick et al. 2023) had shown that language models could interleave reasoning with actions and call external tools, and Wang et al. (2024) had surveyed the emerging components of such agents, but production agentic systems had yet to reach today's sophistication in tool use. Version 1 reflects that state of practice: a single prompt to o4-mini, with no tool access or multi-step control (Section 5.1).

For regulated use cases, a workflow provides the most deterministic outcomes and transparent reasoning. Fixing the evaluation sequence in a process model makes explicit which steps may vary. Deterministic compute nodes execute policy-defined calculations exactly, while inspectable intermediate outputs at every node expose the basis for subsequent decisions. Because accountability attaches to a named officer, the degree of process autonomization (Janiesch et al. 2025) becomes a design decision recorded in the process model (Principle 2), giving operational form to the framed autonomy envisaged for ABPMSs (Dumas et al. 2023).

However, the same fixed sequence makes the workflow brittle on edge cases and prevents officers from bringing further human input into an outcome through clarification or interactive re-prompting. One officer named this inability as the biggest limitation and requested the "ability to re-prompt and clarify" (P4). The most judgement-dependent outputs received the lowest ratings: project classification was the least useful feature (3.33), and the appropriateness of generated rationale was the weakest quality dimension (3.44) (Section 7.4).

Newer agentic-loop designs could address these limits through iterative tool use and human clarification. A direction for future study is to retain workflow structure for auditability and introduce agentic loops for edge cases and clarification, with explicit process boundaries and officer control over each loop.

8.4 Process Co-evolution

Research on AI-augmented BPM proposes that designers bound AI autonomy (Dumas et al. 2023; Acitelli et al. 2025; Janiesch et al. 2025). In our deployment, officers decide how to work within those bounds, and six of them described six different ways of working with the same workflow (Table 9); Section 7.6 reports the agreement scores and the officers' own words.

Our officers reorder their own work around the fixed workflow. P3 reads the proposal first, then uses the system to check "whether Granta has summarised correctly", whereas the depicted workflow places officer review after the AI nodes (Fig. 2). The system itself is unchanged. Structuration theory expects exactly this: people shape how a technology is used through their everyday practice (Orlikowski 2000; DeSanctis and Poole 1994), much as identical CT scanners were followed by different forms of organisation in two radiology departments (Barley 1986).

Where the system helps, officers change their habits; where it constrains them, they ask for changes. Leonardi (2011) describes the same split: perceived affordances lead people to change their routines, and perceived constraints lead them to change

Table 9 Adaptation patterns described in the T1 free-text answers (Q26). One officer per pattern.

Code	What the officer described	Pattern (our label)
P2	Has learned "which parts are usually accurate information"	Perceived-reliability mapping
P3	Reads the proposal first, then runs the system to check "whether Granta has summarised correctly"	Reading-first verification
P5	Uses the system to summarise company background and objectives and to prepare the Cost-Benefit Assessment (CBA)	Task-specific use
P6	Drafts "an offline evaluation based on Granta's inputs"	Offline drafting
P8	"I use the templatized remarks for the project assessment"	Template reuse
P10	Knows which part of the output to refer to when completing each part of the agency's case system	Output navigation

the technology. Officers requested alerts "of signed quotations/proposals" (P3) and of "signed/unsigned financial statement, quotation" (P6), help to "review historical data on supported cost items" (P5), and a fix because the system is "not able to run if one section (ie financial statements) is not uploaded" (P8). P8 encounters our deliberate input check, which stops a run when a required document is missing.

Separate typed outputs let officers trust parts of the output selectively. P2 has learned "which parts are usually accurate information", while P5 chooses the system for company background, objectives and the Cost-Benefit Assessment (CBA). Both depend on outputs that can be told apart, and both fit the *appropriate reliance* that Lee and See (2004) describe, in which trust matches what the automation does well. As designers, we should therefore monitor which outputs officers use and how they use them.

8.5 Human-in-the-Loop Design Principles (RQ3)

Our findings yield three design principles for AI-augmented business processes:

Principle 1: Typed Process Decomposition. Decompose processes into typed, bounded workflow nodes. This extends process decomposition (Dumas et al. 2018) with node types that make determinism explicit: compute nodes are deterministic, AI nodes probabilistic and ingestion nodes format-dependent. Each type can then receive its own quality assurance within one process model, much as workflow resource patterns (Russell et al. 2005) make the resource perspective of a process explicit.

Principle 2: Responsibility Separation within the Process Model. Separate AI-suitable tasks (information extraction, summarisation), deterministic computation (policy-defined calculations, eligibility checks) and human judgement (policy interpretation, final recommendations) within the process model. This extends human task allocation to a three-way division of labour between AI, deterministic automation and officers, each with defined competencies, accountability boundaries and escalation paths, so the model itself records where AI may act. Officers' ratings follow this

division: they rated summarisation the most useful feature (4.89) and the judgement-adjacent outputs lowest, namely project classification (3.33) and the appropriateness of generated rationale (3.44) (Section 7.4). Officers also report that they keep checking what remains theirs to judge: they rated their ability to identify when output needs verification (4.00) about level with their general trust in it (3.89), and six of nine said they "sometimes" need to significantly correct the output and three "rarely", with none choosing "often" (Section 7.5).

Principle 3: Conservative Failure Routing. Route detected failures to officers and withhold incomplete outputs. When a node fails, we withhold all outputs and the case is handled manually; when a required document is missing, the input check stops the run before processing starts (Section 5.4). This extends exception handling patterns (Russell et al. 2006a) to AI components and puts process correctness before process completion, so partial analyses never reach officer review.

Together, the three principles fix in the process model who does what. Typing states how each node can fail, separation assigns each task to AI, computation or an officer, and failure routing returns any failed case to an officer. Final judgement and accountability therefore remain with officers.

8.6 Limitations

While Granta's three-month pilot and seven-month follow-up give a longitudinal view of how the system evolved, the evidence rests largely on officers' own reports from a single implementation.

First, the study covers a single organisation and a single grant scheme. Second, the survey results are descriptive and lack statistical rigour: both waves are small (5 to 7 of the 11 pilot officers answered the T0 surveys, and the T1 survey covers the nine who remained, so the two who left are not represented), only six officers can be paired across waves (Section 7.2), and all other cross-wave comparisons, including the time figures, are unpaired. Third, all ratings and time figures are self-reported, and the T1 "without system" times are hypothetical (Section 7.3). Fourth, our account of system changes between T0 and T1 rests on our own records as developers: bugs were fixed by the end of the pilot and no features were added, so T1 differences may reflect a more stable system, officers' growing familiarity with it or the change in respondents.

Future work should measure process times directly and collect interaction logs, follow larger matched groups across waves, and extend the analysis to other process types and organisations. Version 1 was not re-tested with current agent techniques; testing whether the design principles hold for agent architectures that establish structure during execution is a natural next step (Section 8.3).

9 Conclusion

AI can be integrated into a complex administrative process without weakening accountability if the process model fixes who does what. In Granta, a workflow of typed nodes assigns extraction and summarisation to AI, policy calculations to deterministic code and final judgement to officers. Seven months after the pilot, officers use

the system routinely, rate it more favourably than before and still check and correct its output, so the division of labour held in practice.

For BPM, established ideas of process decomposition and explicit control flow give AI integration a sound footing. The process model stays fixed while officers adapt their own work around it, so their adaptation happens within the structure the model sets.

Future work should test the node typology and design principles in other agencies and processes, and with agent architectures that allow clarification and re-prompting within a node.

Acknowledgements. We thank Executive Director Andrew Cheah and Director Yeo Meow Ling of Enterprise Singapore's Enterprise Division, who sponsored this work, and the division for backing it. We also thank the processing officers who took part in the pilot and both survey waves; their feedback guided system development.

Declarations

- **Funding**: No funding was received for this work.
- **Competing interests**: The authors have no competing interests relevant to this article.
- **Ethics approval and consent to participate**: The study was an operational evaluation of an internal system at Enterprise Singapore, approved by the agency's leadership, Executive Director Andrew Cheah and Director Yeo Meow Ling. Formal ethics review was not required for this internal operational evaluation, and no specific ethical norm or standard (such as the Declaration of Helsinki) was applicable. Participation in the surveys was voluntary. No personal data were collected beyond what identified officers' participation in the pilot and the surveys (at T1, names and work e-mail addresses), and responses were anonymised for analysis. Quotations are reported anonymously under participant codes.
- **Consent for publication**: Not applicable. The article contains no identifiable personal data; survey responses are reported in aggregate or attributed to anonymised participant codes.
- **Data availability**: The extended dataset, comprising the T1 survey instrument and the anonymised T0 and T1 officer responses, is available as a new version of the Zenodo record (concept DOI 10.5281/zenodo.19567512), which also holds the workflow topology in anonymised form, listing node types and the dependencies between them. A few survey items and free-text passages are withheld or redacted there for confidentiality. Government data classification rules prevent release of the grant applications, the underlying case data and the scheme-specific evaluation criteria. For any other data, please contact the corresponding author.
- **Materials availability**: Not applicable.
- **Code availability**: Not applicable.
- **Author contribution**: M. Thirukkumaran and Elston Foo conceived the concept. M. Thirukkumaran was the technical lead and wrote the initial draft of the manuscript. Elston Foo, as innovation and product lead coordinating with the business side, contributed to the analysis and co-wrote the manuscript. Ee Chuan Chang and Yu Kai Chong were developers who contributed to the technical development

of the system. Bei Yin Koh and Kaezel Cho ran the pilot and the surveys with the business team.

- **Relation to prior publications**: The case context, system design and T0 pilot results were first reported in Thirukkumaran et al. (2026a), of which this article is an extended version. The same pilot, including the T0 results, was also reported from a digital-government perspective in a dg.o 2026 paper (Thirukkumaran et al. 2026b). The T1 evaluation, the process co-evolution analysis and the agentic-harness discussion are new to this article and appear in neither earlier paper (Section 1).

References

van der Aalst WMP (2016) Process Mining: Data Science in Action, 2nd edn. Springer, Berlin, https://doi.org/10.1007/978-3-662-49851-4

van der Aalst WMP, ter Hofstede AHM (2005) YAWL: Yet another workflow language. Information Systems 30(4):245–275. https://doi.org/10.1016/j.is.2004.02.002

Acitelli G, Alman A, Maggi FM, et al (2025) Achieving framed autonomy in AI-augmented business process management systems through automated planning. Information Systems 133:102573. https://doi.org/10.1016/j.is.2025.102573

Barley SR (1986) Technology as an occasion for structuring: Evidence from observations of CT scanners and the social order of radiology departments. Administrative Science Quarterly 31(1):78–108. https://doi.org/10.2307/2392767

Berti A, Schuster D, van der Aalst WMP (2024) Abstractions, scenarios, and prompt definitions for process mining with LLMs: A case study. In: BPM 2023 Workshops, LNBIP, vol 492. Springer, pp 427–439, https://doi.org/10.1007/978-3-031-50974-2_32

Busch K, Rochlitzer A, Sola D, et al (2023) Just tell me: Prompt engineering in business process management. In: Enterprise, Business-Process and Information Systems Modeling (BPMDS 2023, EMMSAD 2023), LNBIP, vol 479. Springer, pp 3–11, https://doi.org/10.1007/978-3-031-34241-7_1

De-Arteaga M, Fogliato R, Chouldechova A (2020) A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. In: Proceedings of CHI 2020. ACM, pp 1–12, https://doi.org/10.1145/3313831.3376638

Deelman E, Gannon D, Shields M, et al (2009) Workflows and e-science: An overview of workflow system features and capabilities. Future Generation Computer Systems 25(5):528–540. https://doi.org/10.1016/j.future.2008.06.012

DeSanctis G, Poole MS (1994) Capturing the complexity in advanced technology use: Adaptive structuration theory. Organization Science 5(2):121–147. https://doi.org/10.1287/orsc.5.2.121

Dumas M, La Rosa M, Mendling J, et al (2018) Fundamentals of Business Process Management, 2nd edn. Springer, Berlin, https://doi.org/10.1007/978-3-662-56509-4

Dumas M, Fournier F, Limonad L, et al (2023) AI-augmented business process management systems: A research manifesto. ACM Transactions on Management Information Systems 14(1):1–19. https://doi.org/10.1145/3576047

Grgić-Hlača N, Castelluccia C, Gummadi KP (2022) Taking advice from (dis)similar machines: The impact of human-machine similarity on machine-assisted decision-making. In: Proceedings of HCOMP 2022, vol 10. AAAI Press, pp 74–88, https://doi.org/10.1609/hcomp.v10i1.21989

Hevner AR, March ST, Park J, et al (2004) Design science in information systems research. MIS Quarterly 28(1):75–105. https://doi.org/10.2307/25148625

Janiesch C, Kowalkiewicz M, Rosemann M (2025) Process autonomization: Rethinking business process management. In: BPM 2025 Forum, LNBIP, vol 564. Springer, pp 361–377, https://doi.org/10.1007/978-3-032-02929-4_21

Lee JD, See KA (2004) Trust in automation: Designing for appropriate reliance. Human Factors 46(1):50–80. https://doi.org/10.1518/hfes.46.1.50_30392

Leonardi PM (2011) When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. MIS Quarterly 35(1):147–167. https://doi.org/10.2307/23043493

Mkabane E, Kinigi R (2024) Evaluating the role of AI in grants management: Integration and adoption of technology and innovation. International Journal For Multidisciplinary Research 6(6). https://doi.org/10.36948/ijfmr.2024.v06i06.30248

Orlikowski WJ (2000) Using technology and constituting structures: A practice lens for studying technology in organizations. Organization Science 11(4):404–428. https://doi.org/10.1287/orsc.11.4.404.14600

Parasuraman R, Riley V (1997) Humans and automation: Use, misuse, disuse, abuse. Human Factors 39(2):230–253. https://doi.org/10.1518/001872097778543886

Peffers K, Rothenberger M, Tuunanen T, et al (2012) Design science research evaluation. In: DESRIST 2012, LNCS, vol 7286. Springer, pp 398–410, https://doi.org/10.1007/978-3-642-29863-9_29

Power DJ (2002) Decision Support Systems: Concepts and Resources for Managers. Quorum Books, Westport

Reichert M, Weber B (2012) Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, Berlin, https://doi.org/10.

1007/978-3-642-30409-5

Rosemann M (2014) Proposals for future BPM research directions. In: Proceedings of Asia-Pacific BPM 2014, LNBIP, vol 181. Springer, pp 1–15, https://doi.org/10.1007/978-3-319-08222-6_1

Russell N, van der Aalst WMP, ter Hofstede AHM, et al (2005) Workflow resource patterns: Identification, representation and tool support. In: Advanced Information Systems Engineering (CAiSE 2005), LNCS, vol 3520. Springer, pp 216–232, https://doi.org/10.1007/11431855_16

Russell N, van der Aalst WMP, ter Hofstede AHM (2006a) Workflow exception patterns. In: Advanced Information Systems Engineering (CAiSE 2006), LNCS, vol 4001. Springer, pp 288–302, https://doi.org/10.1007/11767138_20

Russell N, ter Hofstede AHM, van der Aalst WMP, et al (2006b) Workflow control-flow patterns: A revised view. Tech. Rep. BPM-06-22, BPM Center

Schick T, Dwivedi-Yu J, Dessi R, et al (2023) Toolformer: Language models can teach themselves to use tools. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), pp 68539–68551, https://doi.org/10.52202/075280-2997

Sein MK, Henfridsson O, Purao S, et al (2011) Action design research. MIS Quarterly 35(1):37–56. https://doi.org/10.2307/23043488

Skitka LJ, Mosier KL, Burdick M (1999) Does automation bias decision-making? International Journal of Human-Computer Studies 51(5):991–1006. https://doi.org/10.1006/ijhc.1999.0252

Teinemaa I, Dumas M, La Rosa M, et al (2019) Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data 13(2):1–57. https://doi.org/10.1145/3301300

Thirukkumaran M, Foo E, Chang Ee Chuan, et al (2026a) From agentic AI to structured workflows: Designing process architectures for AI-augmented administrative decision support. In: Enterprise, Business-Process and Information Systems Modeling (BPMDS 2026, EMMSAD 2026), LNBIP, vol 594. Springer, pp 213–229, https://doi.org/10.1007/978-3-032-28274-3_14

Thirukkumaran M, Foo E, Koh BY, et al (2026b) AI-assisted decision support for grant processing: A public sector case study. In: Proceedings of the 27th Annual International Conference on Digital Government Research (dg.o 2026), Omaha, Nebraska, USA, June 2026, in press

Vidgof M, Bachhofner S, Mendling J (2023) Large language models for business process management: Opportunities and challenges. In: BPM 2023 Forum, LNBIP, vol 490. Springer, pp 107–123, https://doi.org/10.1007/978-3-031-41623-1_7

Wagner B, Kuebler J, Zalnieriute M (2025) Editorial: Humans in the loop: Exploring the challenges of human participation in automated decision-making systems. Frontiers in Political Science 7. https://doi.org/10.3389/fpos.2025.1611563

Wang L, Ma C, Feng X, et al (2024) A survey on large language model based autonomous agents. Frontiers of Computer Science 18(6):186345. https://doi.org/10.1007/s11704-024-40231-1

Weinzierl S, Dunzer S, Zilker S, et al (2020) Prescriptive business process monitoring for recommending next best actions. In: Business Process Management Forum (BPM 2020), LNBIP, vol 392. Springer, pp 193–209, https://doi.org/10.1007/978-3-030-58638-6_12

Yao S, Zhao J, Yu D, et al (2023) ReAct: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR 2023), URL https://openreview.net/forum?id=WE_vluYUL-X