# Enterprise Workflow Intelligence Through Generative AI and Retrieval-Augmented Generation: A Framework for Adaptive Process Automation

**Florence Eribenne**
Bay Cove Human Services, Boston, MA, USA

**Abstract**

Enterprise workflows are usually automated as fixed paths. Every case follows the same steps, and problems are noticed only when a deadline is missed or a customer complains. Real cases differ, however. Some are at risk of delay, some need extra checks, and some would benefit from a different next step. Predictive process monitoring can forecast how a running case will unfold, including its outcome, next activity, and remaining time. Generative artificial intelligence (AI) with retrieval-augmented generation (RAG) can turn those forecasts into grounded, explained recommendations for action. This article proposes a framework for *workflow intelligence*: the ability of an automated workflow to understand each running case, anticipate problems, and adapt its handling. The framework combines event-driven sensing of running cases, predictive models, retrieval of policies and similar past cases, an LLM-based agent that proposes next best actions, guardrails that bound what the agent may do, a configurable workflow engine that executes actions, and scalable services that deliver these capabilities. Using a design-oriented approach grounded in ten studies published between 2017 and 2024, the article maps the evidence base and presents the framework and its case-level intelligence loop. It defines prediction targets, four levels of case autonomy matched to risk, and case-level guardrails. It illustrates the framework with hospital discharge planning. It argues that adaptive automation is safe and effective when predictions are grounded, recommendations are explained, and autonomy is matched to the risk of each action.

**Keywords:** workflow intelligence, adaptive process automation, predictive process monitoring, prescriptive process monitoring, next best action, generative AI, retrieval-augmented generation, LLM agents, guardrails, human-AI interaction, enterprise workflow

## 1. Introduction

Workflow automation has brought consistency and speed to enterprise processes, but most automated workflows are blind to the state of the individual case. They move each case along the same path until something goes wrong. A loan application sits in a queue until it breaches its deadline. An insurance claim is escalated only after the customer complains. A hospital patient's discharge is delayed because transport was not arranged in time. In each situation, the signs were visible in the case's history well before the problem occurred.

Predictive process monitoring addresses this by applying machine learning (ML) to the event logs of running cases. Tax et al. (2017) showed that long short-term memory (LSTM) networks can predict the next activity, its timing, and the remaining time of running cases. Teinemaa et al. (2019) benchmarked methods for predicting case outcomes. Verenich et al. (2019) compared methods for predicting remaining time. Prescriptive process monitoring goes a step further and recommends actions. Weinzierl et al. (2020) proposed recommending the next best action for a running case based on predicted performance.

Predictions and recommendations become useful only when people understand and trust them, and when they fit the organization's policies. Generative AI can explain a prediction in plain language and draft a recommended action. Retrieval grounds that output in policies and similar past cases (Izacard & Grave, 2021). LLM-based agents can plan and carry out multi-step actions using tools (Wang et al., 2023). Acting on these outputs requires guardrails (Sannidhanam, 2024), human-AI interaction design that supports appropriate trust (Amershi et al., 2019), a configurable workflow engine to execute actions (Basireddy, 2022), and scalable services to deliver intelligence to many cases at once (Sannidhanam, 2020).

This article asks: *How can generative AI and RAG be combined with predictive process monitoring to give enterprise workflows the ability to adapt to each running case safely?* It contributes a mapping of the evidence base (Figures 1 and 2), a framework architecture (Figure 3), a case-level intelligence loop (Figure 4), a definition of prediction targets, a four-level autonomy model, case-level guardrails, and an illustrative application.

## 2. Related Work

### 2.1 Predictive process monitoring

Tax et al. (2017) applied LSTM neural networks to event logs to predict the next activity, the time until it occurs, and the remaining time of running cases. They showed improved accuracy over earlier methods across several datasets. Teinemaa et al. (2019) reviewed and benchmarked outcome-oriented predictive monitoring, which predicts whether a case will end in a particular

outcome, such as a missed deadline or a rejected application. They compared ways of encoding case histories and grouping cases, and found that simple encodings combined with gradient-boosted trees often performed well. Verenich et al. (2019) surveyed and benchmarked methods for predicting remaining time and found that no single method was best across all datasets. Together, these studies show that the future of a running case can be predicted usefully from its history.

### 2.2 Prescriptive process monitoring

Weinzierl et al. (2020) proposed prescriptive monitoring that recommends the next best action for a running case. They predict candidate next activities and evaluate each against a performance measure, while respecting constraints on which activities are allowed. This moves monitoring from warning about problems to suggesting how to avoid them.

### 2.3 Retrieval-augmented generation

Izacard and Grave (2021) showed that a generative model reading many retrieved passages at once can answer questions more accurately than extractive approaches, and that accuracy improves as more passages are retrieved. For workflow intelligence, retrieval can supply the policies that govern a case, notes from similar past cases and how they were resolved, and relevant procedures. A generative model can then synthesize these into grounded explanations and recommendations.

### 2.4 LLM-based agents

Wang et al. (2023) survey autonomous agents built on large language models (LLMs). They describe architectures combining profiling, memory, planning, and action through tools, and they discuss applications and challenges, including reliability and evaluation. In workflow intelligence, an agent can gather case information, consult predictions and retrieved knowledge, and propose or take actions through the workflow engine's defined operations.

### 2.5 Guardrails and human-AI interaction

Sannidhanam (2024) examines guardrails and safety mechanisms for LLM-powered enterprise applications, including input filtering, output validation, restrictions on topics and actions, protection of sensitive data, and escalation to humans. Amershi et al. (2019) proposed guidelines for human-AI interaction. They include making clear what the system can do and how well, showing contextually relevant information, supporting efficient correction and dismissal, and explaining why the system did what it did. These guidelines shape how predictions and recommendations should be presented to case workers.

### 2.6 Workflow execution and scalable services

Basireddy (2022) proposes configurable enterprise workflow architecture for digitizing risk management, with steps, approvals, escalations, and remediation defined as configurable workflows with audit trails. The workflow engine defines which actions are possible for a case and records what is done. Sannidhanam (2020) proposes a scalable web services architecture for healthcare data processing, using modular services and horizontal scaling to handle growing data and request volumes. Predictive models, retrieval, and agents can be delivered as scalable services that serve many running cases at once.

### 2.7 Research gap

Predictive and prescriptive process monitoring have developed within process mining research, while generative AI, RAG, and agents have developed in AI research. Little work combines them into an architecture in which workflows adapt to each running case with grounded explanations, bounded autonomy, and governed execution. This article provides such an architecture.

## 3. Research Methodology

The study follows a design-oriented research approach. The problem, that automated workflows treat cases uniformly and react to problems too late, was identified from the literature. Requirements were derived from the reviewed studies: case-level prediction, actionable recommendations, grounding in policies and experience, explanation, bounded autonomy, governed execution, and scalability. The framework was designed from ten studies and demonstrated through an illustrative hospital case.

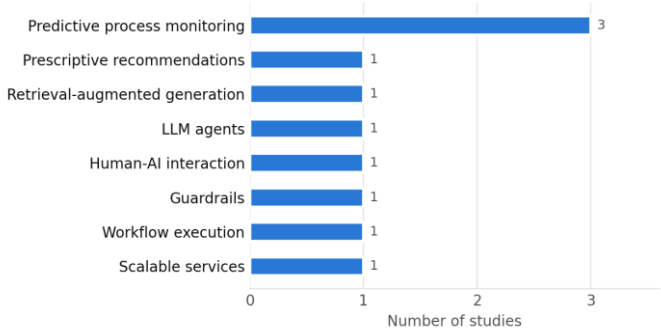


**Figure 1. Number of reviewed studies primarily supporting each theme (n = 10)**

To make the evidence base transparent, each study was assigned to the theme it primarily supports. Figure 1 shows the distribution across themes, and Figure 2 shows the studies by publication year. These figures describe the literature used in this article. They are not

performance measurements. The framework has not been evaluated empirically, which is left to future work.

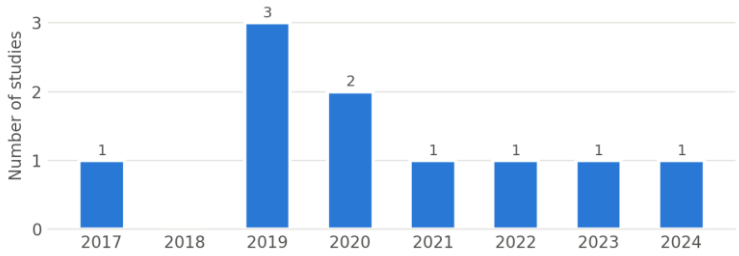


**Figure 2. Reviewed studies by publication year (n = 10)**

Predictive process monitoring is the most developed theme. The generative, agent, and control themes each rest on a single study, which reflects how recently generative AI has been considered for case-level workflow intelligence.

## 4. Framework Architecture

The framework has seven components (Figure 3 and Table 1).

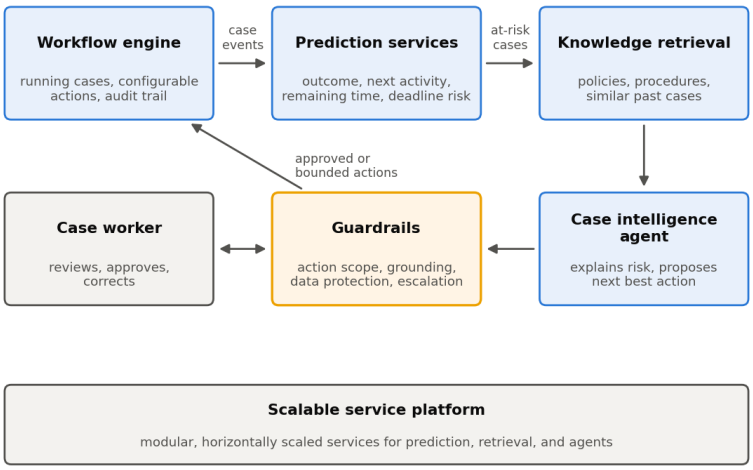


**Figure 3. Framework for enterprise workflow intelligence**

**Case sensing.** Every event in a running case, such as a step started, a document received, a status changed, or a deadline approaching, is streamed from the workflow engine to the intelligence services (Basireddy, 2022).

**Table 1. Components of the workflow intelligence framework**

| Component | Function | Key mechanisms | Supporting studies |
|---|---|---|---|
| Case sensing | Track the state of every running case | Workflow events streamed per case | Basireddy (2022) |
| Prediction services | Forecast case futures | Outcome, next activity, remaining time, deadline risk | Tax et al. (2017); Teinemaa et al. (2019); Verenich et al. (2019) |
| Knowledge retrieval | Supply relevant policies and experience | Retrieval of policies, procedures, similar past cases | Izacard & Grave (2021) |
| Case intelligence agent | Explain predictions and propose next best actions | LLM agent using predictions, retrieved knowledge, and workflow tools | Weinzierl et al. (2020); Wang et al. (2023) |
| Guardrails | Bound what the agent may say and do | Action limits, output validation, data protection, escalation | Sannidhanam (2024) |
| Workflow engine | Execute actions and record them | Configurable actions, approvals, audit | Basireddy (2022) |
| Service platform | Deliver intelligence at scale | Modular, horizontally scalable services | Sannidhanam (2020) |

**Prediction services.** Models trained on historical event logs predict, for each running case, the likely outcome (Teinemaa et al., 2019), the next activity and remaining time (Tax et al., 2017; Verenich et al., 2019), and the risk of missing its deadline.

**Knowledge retrieval.** For cases flagged by predictions, the retrieval service finds the policies that apply, relevant procedures, and similar past cases with notes on how they were resolved (Izacard & Grave, 2021).

**Case intelligence agent.** An LLM-based agent (Wang et al., 2023) combines the predictions and retrieved knowledge. It explains why a case is at risk, then evaluates candidate next actions drawn from those the workflow permits, following the logic of prescriptive monitoring (Weinzierl et al., 2020). It proposes the best action with an explanation and cited sources.

**Guardrails.** Every agent output passes through guardrails (Sannidhanam, 2024). The agent may only propose actions the workflow defines for the case's current state, and outputs are checked against case data and policies. Sensitive information is protected, and actions above set risk levels are escalated to people.

**Workflow engine.** Approved actions are executed through the configurable workflow engine and recorded, together with the prediction, recommendation, and decision, in the case's audit trail (Basireddy, 2022).

**Service platform.** Prediction, retrieval, and agent capabilities are delivered as modular services that scale with the number of running cases (Sannidhanam, 2020).

## 5. The Case-Level Intelligence Loop

Figure 4 shows how the framework handles a single running case.

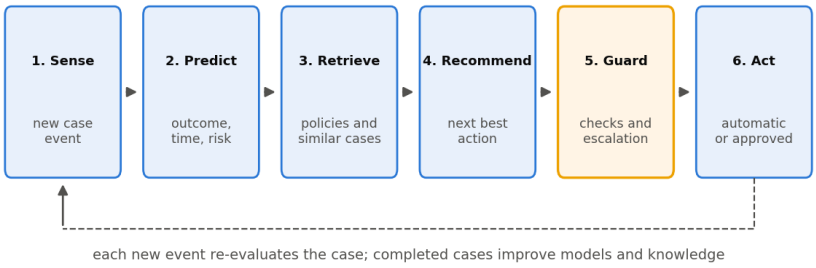


**Figure 4. Case-level intelligence loop**

The loop has six steps. The framework **senses** each new event in the case and predicts the case's future. If the prediction shows risk, it retrieves relevant policies and similar cases. The agent then recommends a next best action with an explanation. Guardrails guard the recommendation, and the workflow engine **acts**, either automatically or after human approval, depending on the autonomy level. Each subsequent event restarts the loop, so the framework continuously re-evaluates the case. Outcomes of completed cases are added to training data and to the retrieval knowledge base.

### 5.1 Prediction targets

Table 2 sets out the prediction targets and how each is used.

**Table 2. Prediction targets for workflow intelligence**

| Target | Question answered | Example use | Supporting studies |
|---|---|---|---|
| Case outcome | Will this case end well? | Flag cases likely to be rejected or escalated | Teinemaa et al. (2019) |
| Next activity | What will happen next? | Prepare resources for the expected next step | Tax et al. (2017) |
| Remaining time | How long until completion? | Update customers and plan capacity | Verenich et al. (2019) |
| Deadline risk | Will this case miss its deadline? | Trigger preventive action early | Verenich et al. (2019); Teinemaa et al. (2019) |
| Next best action | What should be done now? | Recommend the action most likely to improve the outcome | Weinzierl et al. (2020) |

### 5.2 Levels of case autonomy

The consequences of acting on a recommendation vary. Table 3 defines four levels of autonomy, each matched to the risk of the action.

**Table 3. Levels of case autonomy**

| Level | What the system does | Suitable for | Human role |
|---|---|---|---|
| 1. Inform | Shows predictions and explanations | All cases | Case worker decides what to do |
| 2. Recommend | Proposes a next best action with rationale and sources | Moderate-risk decisions | Case worker accepts, edits, or rejects |
| 3. Act with approval | Prepares the action for one-click approval | Routine actions with some consequence | Case worker or supervisor approves |
| 4. Act within bounds | Executes low-risk actions automatically | Reversible, low-risk actions within set limits | Oversight through sampled review and audit |

Following Amershi et al. (2019), every level shows the case worker what the system predicts, how confident it is, and why. Recommendations can be dismissed or corrected with one action, and corrections are recorded to improve future recommendations. Guardrails enforce the autonomy level set for each action type (Sannidhanam, 2024).

### 5.3 Case-level guardrails

Four guardrails are particularly important at the case level. First, action scope: the agent may only propose actions the workflow permits in the case's current state (Basireddy, 2022). Second, grounding: explanations and recommendations must cite retrieved policies or case data, and claims that do not match case records are blocked (Sannidhanam, 2024). Third, data protection: the agent sees only the case data its task requires. Fourth, escalation: low-confidence predictions, conflicting policies, or high-risk actions are routed to people.

## 6. Illustrative Case: Hospital Discharge Planning

Consider a hospital whose discharge workflow coordinates medical sign-off, medication reconciliation, pharmacy dispensing, transport, and follow-up appointments. Delayed discharges occupy beds needed by new patients.

**Sensing and prediction.** Every discharge-related event for each inpatient is streamed to the intelligence services (Basireddy, 2022; Sannidhanam, 2020). Each morning, the prediction services estimate the remaining time for each planned discharge and the risk that it will slip

beyond the planned day (Verenich et al., 2019; Teinemaa et al., 2019). For one patient due to go home today, the model predicts a high risk of delay. The pharmacy dispensing step has not started, and similar cases with this pattern were often delayed by late medication.

**Retrieval and recommendation.** The agent retrieves the hospital's discharge policy, the pharmacy's priority procedure, and notes from similar past cases (Izacard & Grave, 2021). It explains to the discharge coordinator that the patient is at risk of delayed discharge because dispensing has not started and the patient's medication list includes an item that usually takes longer to prepare. It recommends requesting priority dispensing and booking transport for mid-afternoon rather than morning, citing the pharmacy procedure and two similar past cases (Weinzierl et al., 2020; Wang et al., 2023).

**Guardrails and autonomy.** Requesting priority dispensing is a Level 3 action. The agent prepares the request, and the coordinator approves it with one click. Rebooking transport within the same day is a reversible, low-risk Level 4 action, which the system carries out automatically within its limits. The agent's explanation is checked against the patient's record before it is shown. The agent has access only to discharge-related information, not the full clinical record (Sannidhanam, 2024).

**Human oversight.** Later, the agent suggests contacting the patient's family to arrange collection instead of hospital transport. The coordinator knows the family is unavailable and dismisses the suggestion with a note. The correction is recorded for future learning (Amershi et al., 2019).

**Outcome.** The patient is discharged that afternoon. The completed case, including the predictions, actions, and outcome, is added to the training data and the knowledge base.

The case shows workflow intelligence at work. The system anticipated a problem, grounded its advice in policy and experience, acted automatically only where safe, and deferred to human judgment where the person knew more.

## 7. Discussion

### 7.1 Implications

For research, the framework connects predictive and prescriptive process monitoring with generative AI, retrieval, and agents, which have developed separately. It shows how predictions can become explained, grounded, and bounded actions within governed workflows. The evidence mapping suggests that research combining these fields is at an early stage. For practitioners, the framework suggests a clear path. Start by informing case workers with predictions and explanations, add recommendations grounded in policies and past cases, then

introduce approval-based and bounded automatic actions as confidence grows. Guardrails and audit should be in place from the start.

### 7.2 Limitations and future work

The framework was demonstrated through an illustrative case and draws on a focused set of ten studies, and the evidence mapping reflects only these studies. Future work should implement and evaluate the framework, measuring prediction accuracy, the rate at which recommendations are accepted, reductions in delays and escalations, and the frequency and causes of human overrides. Further research is needed on evaluating LLM agents in workflow settings, on learning from dismissed recommendations, on setting autonomy levels from evidence, and on explaining process predictions to non-specialist case workers.

## 8. Conclusion

Automated workflows can become intelligent when they understand each running case, anticipate problems, and adapt their handling. This article proposed a framework that senses case events and predicts outcomes, next steps, and remaining time. It retrieves relevant policies and past experience, uses an LLM-based agent to recommend next best actions with explanations, bounds those actions with guardrails, and executes them through a configurable workflow engine on scalable services. Four levels of case autonomy match automation to risk. The hospital discharge case showed the framework preventing a delay while respecting human judgment. The central conclusion is that adaptive process automation is safe and effective when predictions are grounded, recommendations are explained, and autonomy is matched to the risk of each action.

## References

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., … Horvitz, E. (2019). Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Paper 3, pp. 1–13).

Basireddy, S. R. (2022). Digitizing risk management through configurable enterprise workflow architecture. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 14*(4), 231–239.

Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 874–880).

Sannidhanam, A. H. (2020). Scalable web services architecture for healthcare data processing. *International Journal of Recent Advances in Science and Technology, 7*(01), 1–14. https://ijrast.in/index.php/ijrast/article/view/65

Sannidhanam, A. H. (2024). Guardrails and safety mechanisms for LLM-powered enterprise applications. *International Journal of Engineering Science & Humanities, 14*(3), 273–285. https://www.ijesh.com/j/article/view/933

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. In *Advanced Information Systems Engineering (CAiSE 2017)* (Lecture Notes in Computer Science, Vol. 10253, pp. 477–492). Springer.

Teinemaa, I., Dumas, M., La Rosa, M., & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data, 13*(2), Article 17.

Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Teinemaa, I. (2019). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology, 10*(4), Article 34.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., … Wen, J.-R. (2023). *A survey on large language model based autonomous agents* (arXiv:2308.11432). arXiv.

Weinzierl, S., Dunzer, S., Zilker, S., & Matzner, M. (2020). Prescriptive business process monitoring for recommending next best actions. In *Business Process Management Forum (BPM 2020)* (Lecture Notes in Business Information Processing, Vol. 392, pp. 193–209). Springer.