Journal of Data Science 0 (0), 1–25
July 2026
DOI: 10.6339/26-JDS1216
Data Science in Action

Leveraging Artificial Intelligence and Automation for Enhancing School Improvement Efforts

Graham Chickering[1,*], Christina Jones[1], and Naomi Blaushild[1]
[1]*American Institutes for Research, Technical Solutions, Arlington, VA 22202, United States*

Abstract

Advances in AI and automation are reshaping qualitative research workflows, making processes more efficient, accurate, consistent, and scalable. This paper presents innovations developed for the Illinois Needs Assessment project, a statewide initiative led by the Illinois State Board of Education and the American Institutes for Research to conduct comprehensive needs assessments for schools that need intensive or comprehensive support. To address the scale and tight timeline requirements of the project, the team designed three interconnected pipelines that work together to produce a finalized report. The first, an Audio Pipeline, uses Whisper and generative AI to automate transcription, text-based speaker role attribution, thematic coding, and insight generation from focus groups and interviews. The second, a Report Generation Pipeline, integrates Airtable automations with AWS infrastructure to produce customized school reports that merge AI-generated findings with survey data, school performance metrics, and contextual comparisons. Third, the Needs Assessment Summary Report automates the assembly of all quantitative and qualitative inputs into a polished, customizable deliverable that combines efficiency with expert review. Together, these pipelines replace ad hoc manual workflows with reproducible, consistent systems that enhance data quality, reduce error, and broaden access for non-technical users. The integrated design demonstrates how automation and generative AI can reduce manual burdens, shorten delivery timelines, and support timely, data-informed, and human-centered decision-making in education.

Keywords *automation; data pipelines; educational research; generative AI; qualitative analysis*

1 Introduction

1.1 Use Case Context

Comprehensive needs assessments are foundational tools in educational improvement efforts, particularly for schools identified for intensive or comprehensive support (U.S. Department of Education, 2001). These assessments serve multiple, overlapping purposes: they provide a systematic diagnostic of school conditions, identify root causes of performance gaps, and establish a shared understanding of challenges and opportunities among stakeholders. Unlike standardized performance metrics alone, comprehensive needs assessments are multi-dimensional by design. They integrate existing administrative data with on-the-ground data collection, such as classroom observations, staff surveys, and interviews or focus groups with school leadership, staff,

*Corresponding author. Email: grahamchickering@gmail.com.

Received August 13, 2025; Accepted January 5, 2026

ILNA School Improvement Practice Areas and Indicators			
Practice Area 1: Leadership and Vision	**Practice Area 2: Curriculum, Instruction, and Assessment**	**Practice Area 3: Culture and Climate**	**Practice Area 4: Targeted Instruction and Support**
1A. Shared Vision and Goals	2A. High-Quality, Defined Curriculum	3A. Positive Interpersonal Relationships	4A. Multitiered Systems of Support
1B. Distributed Leadership	2B. Collaborative Planning	3B. Warm and Responsive Environment	4B. Inclusive and Differentiated Instruction
1C. Culture of Continuous Improvement	2C. High Expectations and Rigorous Instruction	3C. Student Voice and Feedback (Secondary Only)	4C. Enrichment (Elementary Only)
1D. Aligned, Consistent Professional Development	2D. Teacher Observation and Feedback	3D. Family Collaboration	4D. College and Career Readiness Opportunities (Secondary Only)
	2E. Data Collection and Collaborative Data Use	3E. Community Resources and Engagement	

Figure 1: ILNA school improvement practice areas and indicators.

and students. This integration generates an authentic and holistic picture of a school's strengths and areas of improvement (Corbett and Redding, 2017). This comprehensive perspective allows school leaders, district staff, and policymakers to tailor support in ways that account for both structural conditions and lived experiences within schools.

To provide district, state, and school staff with actionable insights for school improvement planning, the Illinois State Board of Education partnered with the American Institutes for Research (AIR) to conduct comprehensive needs assessments for schools designated by the state's accountability report card as needing comprehensive or intensive support. Drawing on state and district school improvement frameworks and prior research on effective practices for school delivery (Lane et al., 2014; Pan et al., 2021), AIR developed the Illinois Needs Assessment (ILNA) framework to provide schools with feedback on four school improvement practice areas and 16–17 supporting indicators, as shown in Figure 1.

Comprehensive needs assessments involve gathering and analyzing multiple sources of qualitative data, which require considerable human and financial resources and time. Data collection begins with surveys administered to school leaders and instructional staff followed by a one- or two-day site visit conducted by AIR researchers. Researchers conduct 10–20 classroom observations, interview school leaders, and facilitate staff focus groups. Following the site visit, AIR researchers synthesize qualitative data with school administrative records and staff survey results, integrating multiple evidence sources to create a nuanced and contextually grounded analysis. This process involves reviewing interview and focus group transcripts, analyzing observation scores, and comparing findings to relevant performance metrics and demographic data. The resulting Needs Assessment Summary Report details each school's strengths, effective practices, and areas of growth, providing stakeholders with an evidence-based foundation for strategic planning and resource allocation.

In the project's first year, AIR conducted 281 individualized needs assessments across the state and received positive feedback from school and district leaders about the usefulness and quality of the reports. However, delivery time for reports could take up to three months, preventing schools from using data from the report during their improvement planning for the upcoming year. Thus, expediting this process for future years of the project became essential.

The sections below describe the three innovations developed by our team to ensure that comprehensive needs assessments provide schools with high-quality data in a timely manner. In

particular, we focus on discussion around benefits of this design, the system architecture, their initial performance, and the implications for future applied qualitative research.

1.2 Research Questions and Contributions

Large-scale qualitative analysis in education presents persistent challenges related to processing speed, reproducibility, human–resource constraints, and the consistency of analytic judgments. Although recent advances in automation and generative AI offer new opportunities to streamline key components of qualitative workflows, questions remain about how these technologies can be integrated responsibly into end-to-end research processes in ways that enhance, not undermine, methodological rigor. To address these gaps and more clearly articulate methodological contributions, this work is guided by the following research questions:

1. How can cloud-native automation and generative AI be incorporated into qualitative research workflows in large-scale mixed-methods projects?
2. What methodological and analytic considerations emerge from combining automated pipelines with structured human-in-the-loop review?
3. Where does human judgment remain essential for ensuring validity, reliability, and contextually grounded interpretation?
4. To what extent do the proposed pipelines improve measurable outcomes, such as transcript processing time, report delivery time, and required human effort, relative to what was achieved under the prior manual workflows?

Beyond detailing the architecture and functionality of the three interconnected pipelines, this work advances several methodological and empirical contributions. Given that workflow speed and delivery timelines were central limitations of the prior years work, we include time-based performance metrics to clarify the practical impact of the proposed system improvements:

- A reproducible, end-to-end analytic framework for AI-assisted qualitative research that integrates transcription, speaker-role attribution, protocol-aligned tagging, thematic extraction, and structured evidence generation. The framework provides measurable performance improvements, including reducing processing time from 4–8 hours to approximately 25 minutes.
- A domain-agnostic orchestration architecture that uses containerization, version-controlled execution environments, and serverless orchestration to ensure transparent and reproducible workflows. This architecture reduces report generation time from roughly 60 minutes to under 10 minutes per school, an 84% reduction, and supports scalable, parallel processing.
- An empirical demonstration of end-to-end delivery-time improvements across the full workflow. Integrating AI-enabled audio processing with automated report generation reduced the overall time between data collection and delivery of finalized reports from several months in Year 1 to a few weeks in Year 2, enabling schools to incorporate findings directly into their improvement planning cycle.

Taken together, these contributions position the work as both a practical system implementation and a methodological advance in how large-scale qualitative research can be conducted with transparency, rigor, and reproducibility.

2 System Architecture and Design

To address the inefficiencies identified in the comprehensive needs assessment process, the team prioritized optimizing the reporting workflow to reduce both the technical burden and time-to-insight, without sacrificing quality. The team developed and improved three interdependent au-

tomation pipelines designed to streamline, accelerate, and enhance the overall reporting process: (1) the Audio File AI Pipeline ("Audio Pipeline" moving forward), (2) the Report Generation Pipeline, and (3) Needs Assessment Summary Report.

- The Audio Pipeline ingests raw audio recordings from focus groups, and automates transcription, text-based speaker role attribution, natural language processing based tagging, and summarization using large language models (LLMs) like GPT-4o. This enables structured extraction of stakeholder insights at a scale and speed unmatched by manual analysis.
- The Report Generation Pipeline then integrates these outputs along with survey data and other school-level indicators to produce tailored, high-quality reports using a cloud infrastructure along with a modular templating system to run and generate reports on demand.
- The Needs Assessment Summary Report serves as the final, synthesized output of this automated ecosystem, combining quantitative performance metrics, survey results, and qualitative findings into a single, coherent deliverable. Generated through a fully containerized, reproducible workflow, the report leverages preformatted templates to ensure consistency while allowing for post-generation refinement by researchers. By embedding human-in-the-loop review within an otherwise automated workflow, the system achieves both efficiency and rigor, enabling consistent, transparent, and reproducible reporting without sacrificing the contextual judgment that qualitative analysis requires.

The purpose of these pipelines extends beyond efficiency. They encode a methodological shift: from ad hoc manual workflows to reproducible, consistent systems that enhance data quality, reduce error, and broaden access for non-technical users. The automation process necessitated stronger error-handling, improved input validation, and the development of user interfaces that empower researchers to self-serve report generation. As such, these pipelines represent not just a technical solution, but a reimagining of how qualitative research infrastructures can support large-scale school improvement efforts.

2.1 Audio Pipeline

The comprehensive needs assessment prioritized on-site qualitative data collection which include interviews and focus groups with school leaders, staff, and students, to capture context-specific insights from participants' own accounts of their experiences, challenges, and priorities. Historically, these qualitative data have been analyzed entirely by human researchers through rigorous coding, in which segments of interview transcripts are categorized according to inductive or deductive frameworks developed by the research team. This work is often supported by qualitative analysis software such as NVivo, which aids in coding, analysis, and inter-rater reliability checks (Miles et al., 2014). This manual approach, while essential for ensuring methodological rigor, is highly labor-intensive and time-consuming, often requiring 4–8 hours of work to code a single transcript, depending on length and complexity (Miles et al., 2019). Given the substantial volume of qualitative data collected in this project and the need to deliver timely, actionable feedback to schools, the team developed AI-assisted tools to expedite analysis without sacrificing analytical quality.

The Audio Pipeline, as shown in Figure 2, was developed to expedite the transformation of raw audio recordings from focus groups and interviews into structured qualitative data. The Audio Pipeline categorizes text from the audio files and aligns to relevant indicators, Figure 1, such that AIR researchers can quickly gather necessary information when writing the final report. This process replaces traditionally manual tasks including transcription, speaker identification, and qualitative coding with a cloud-based system that combines LLMs and researcher-

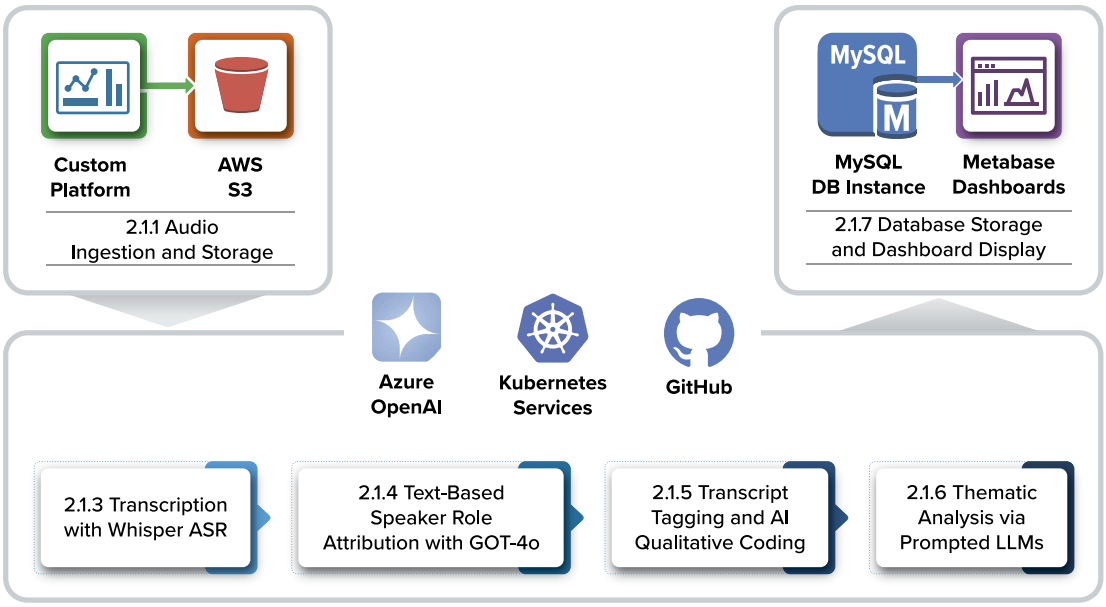

Figure 2: Audio pipeline infrastructure.

validated prompts. This modular pipeline supports human-in-the-loop validation while enabling researchers to process audio files across multiple schools within hours. By automating repetitive stages and coordinating orchestration via cloud services, this architecture delivers significant improvements in delivery time and reproducibility, consistent with documented benefits of managed data pipelines in large-scale analytics contexts (Ogeawuchi et al., 2022).

2.1.1 Audio Ingestion and Storage

The process begins with audio recordings collected during focus groups conducted by the research team. These files are uploaded through a secure, custom-built survey platform and automatically transferred to a centralized Amazon S3 bucket. Once a new file is uploaded, a monitoring script continuously scans the bucket and programmatically initiates processing workflows for unprocessed files. This automated detection mechanism eliminates the need for manual oversight, enabling near real-time ingestion and tracking of interview data.

2.1.2 Processing Infrastructure and Version Control

The core processing steps—transcription, text-based speaker role attribution, tagging, and analysis—are orchestrated through an Azure Kubernetes Service (AKS) cluster. All tasks are containerized for scalability and isolation, with pods configured to automatically pull the latest version-controlled code from a private GitHub repository. This ensures consistent execution across deployments while supporting agile iteration and traceable outputs (Nascimento et al., 2024). Azure Key Vault handles secure credential access, while custom logging mechanisms capture execution metadata and diagnostics to support monitoring, troubleshooting, and traceability (Microsoft, 2025).

2.1.3 Transcription with Whisper ASR

Once a new audio file is detected in the storage system, the transcription process is initiated using OpenAI's Whisper automatic speech recognition (ASR) model, accessed via the Azure OpenAI API. Whisper is a well-established ASR model trained on a large and diverse dataset of multilingual and multitask supervised data. It is capable of producing highly accurate transcriptions even in noisy environments and across a range of speaking styles, making it well-suited for qualitative data collection, such as interviews and focus groups (Radford et al., 2022).

In this pipeline, Whisper converts raw audio recordings into word transcripts that preserve the natural structure and flow of speech. While it does not natively assign speaker labels, the model delivers coherent and temporally aligned transcriptions that serve as a reliable foundation for downstream tasks, such as text-based speaker role attribution, thematic tagging, and qualitative analysis.

By leveraging Whisper's transcription capabilities via the Azure OpenAI API, the pipeline substantially reduces manual labor while delivering high accuracy. Benchmarks indicate that the Whisper medium model achieves a median word error rate of approximately 11.5% on meeting audio, while the large model reduces error rates further in controlled conditions (Voicegain, 2023). In practical Azure deployments, Whisper can process an hour of audio in approximately five minutes, representing a 12× real-time speedup over human transcription (Center for Computation and Visualization, 2025). In comparison, human transcription typically yields a WER of 4–5%, meaning Whisper approaches parity in many real-world contexts while providing consistent reproducibility across large datasets (Glenn et al., 2010).

2.1.4 Text-Based Speaker Role Attribution with GPT-4o

Following transcription, the next step in the pipeline is text-based speaker role attribution. This step is especially critical in interview-based research, where attributing dialogue to the correct speaker (e.g., interviewer versus interviewee) is necessary for accurate downstream analysis. For focus groups in particular, this step also helps researchers understand how widespread a view is among participants.

In our system, text-based speaker role attribution is performed using GPT-4o through the Azure OpenAI API. Custom prompt chains are executed within an AKS cluster to identify speaker roles from the transcribed text (see Supplementary Materials at https://github.com/gchickering21/SDSS2025_materials). These prompts are engineered to infer conversational dynamics based on linguistic and structural cues, such as question–answer sequences, lexical formality, and discourse markers. For instance, if one speaker is consistently initiating turns with interrogatives or follow-up probes, they are likely labeled as the interviewer. Conversely, speakers offering detailed responses, reflections, or domain-specific narratives are marked as interviewees.

The text-based speaker role attribution process incorporates a hybrid approach, combining rule-based heuristics with GPT-4o reasoning to maximize classification accuracy. Transcript segments produced by Whisper are passed to this module, which applies logic to assign meaningful role-based labels such as Interviewer, Interviewee_1, or Interviewee_2. This disambiguation is particularly valuable in multi-participant conversations, where acoustic separation alone is insufficient to resolve speaker identity or intent.By leveraging GPT-4o's nuanced language understanding, our text-based speaker role attribution module provides a scalable and semantically aware solution for speaker labeling (OpenAI, 2023). This module operates reliably even in cases where traditional acoustic diarization methods may struggle due to overlapping speech, microphone variability, or background noise.

2.1.5 Transcript Tagging and AI Qualitative Coding

Once the text-based speaker role attribution step is complete, the transcript is segmented into discrete text chunks based on speaker turns. Each segment corresponds to a single uninterrupted span of speech from one speaker, allowing for a structured representation of the dialogue sequence.

The tagging process begins by identifying all segments spoken by the interviewer. These segments serve as anchor points for downstream analysis, as the interview protocol is designed such that specific interviewer questions correspond to specific school indicators (Figure 1). To operationalize this alignment, we utilize a "interview protocol crosswalk", a structured mapping that links canonical interviewer questions to the set of school improvement indicators.

Using this crosswalk, each interviewer segment is compared to the bank of canonical questions via a semantic similarity algorithm. Specifically, embeddings are computed for each interviewer chunk and for each reference question in the protocol, and cosine similarity scores are used to identify the most semantically aligned question. The matched question, along with its associated indicator, is then applied as a tag to that specific chunk. All interviewee segments that follow are assigned the same indicator as the most recent interviewer chunk, continuing until a new interviewer segment introduces a different topic and a new indicator is assigned. This method dynamically groups transcript content into thematic units that mirror both the conversational flow and the structure of the interview protocol.

By combining linguistic similarity with structured protocol logic, the system reliably tags transcript segments even when interviewer phrasing deviates from the script. This process results in a transcript where each segment is systematically associated with the relevant indicator, preserving the intended structure of the interview for subsequent analysis.

2.1.6 Thematic Analysis via Prompted LLMs

After transcript tagging is complete for each transcript, the pipeline qualitatively analyzes the tagged transcript for common themes in participants' responses using LLMs. The primary objective of this step is to extract findings, identify patterns of agreement or disagreement, and surface illustrative quotes aligned to specific school improvement indicators (Figure 1). For example, after tagging the interview or focus group transcript for responses related to professional development (aligned to Indicator 1D. Aligned, Consistent Professional Development), the pipeline summarizes how participants described the school's professional development opportunities and whether participants agree or disagree with one another about the existence or effectiveness of these offerings and pulls relevant quotations from participants about their perceptions of professional development.

Each tagged transcript is first grouped by indicator (one of the 18 school improvement indicators listed in Figure 1). For each indicator, a structured prompt is constructed containing three main components: (1) a short description of the transcript context, (2) the relevant excerpt(s) from the transcript, and (3) the definition of the indicator and associated examples. The prompt then instructs the LLM to identify up to five key findings, each supported by at least one quote from the transcript. The expected output is a two-column table; one column for the finding description and one for the corresponding supporting quote. Examples of output files can be found at Supplementary Materials at https://github.com/gchickering21/SDSS2025_materials.

By enforcing a consistent structure, requiring direct quotes to support all claims, and incorporating automated quality checks, the approach helps minimize hallucinations and enables systematic comparison across transcripts, indicators, and schools. All prompt responses are stored

alongside the tagged transcript and code metadata, enabling downstream auditing, summarization, and cross-case analysis. The LLMs enable scalable insight generation from large volumes of qualitative data while allowing humans to review and verify accuracy through the use of identified quotes and transparent traceability between model-generated findings and source text.

2.1.7 Database Storage and Dashboard Display

After each transcript has been processed through text-based speaker role attribution (Subsubsection 2.1.4), tagging (Subsubsection 2.1.5), and thematic LLM-based analysis (Subsubsection 2.1.6), the resulting data which consists of speaker-attributed text segments, assigned indicator codes, extracted findings, supporting quotes, and metadata, are uploaded to an AWS Relational Database (Amazon Web Services, 2025b). This database is designed to support efficient querying across multiple dimensions, such as school, indicator, date, or user-defined filters. Versioning is built into the storage layer to ensure traceability and support longitudinal comparisons.

To facilitate interpretation and use of the findings, a frontend dashboard was developed using Metabase, an open source dashboarding tool (Metabase Inc, 2025). This dashboard connects directly with the database and enables users to navigate through each transcript, view associated findings by school improvement indicator (Figure 1), and review AI-generated summaries and quotations. By connecting the outputs of the AI-enabled pipeline to a user-friendly dashboard, this step ensures that researchers can easily access and interact with the results to engage in data-driven decision making.

2.1.8 Researcher Use and Review

After an audio file has been fully processed through all the previous steps, AIR researchers use the data dashboard to review the pipeline's outputs. The data dashboard allows researchers to determine ratings and themes for each topic by reviewing and synthesizing all data relevant to that topic (including responses to questions about that topic across interviews or focus groups) in one place. Researchers review the generated findings, selected quotations, and summaries (Subsubsection 2.1.6) to look for evidence of school structures and practices as described by school leaders, staff, and, in some cases, students. The dashboards also display notes taken by researchers who facilitated or were present for the interview or focus group. These notes provide additional details about the context and makeup of the interview or focus group, in addition to key insights from participants, to complement and bolster the AI generated findings and insights. While the AI findings and interview facilitator notes often provide enough data to begin writing the content of the report, researchers can also access the speaker identified transcript through the data dashboard for additional context and to double check the accuracy of the AI findings.

2.1.9 Initial Evaluation of AI-Generated Findings

The use of AI for transcription and coding allows AIR researchers to spend more time determining school ratings and writing detailed, school-specific Needs Assessment Summary Reports. However, AIR researchers raised concerns about the AI findings' accuracy and usefulness to report writing. While researchers always had access to the original transcripts and audio recordings for additional information and context, revisiting sections of a full interview is time-consuming and undermines the use of the AI generated findings and summaries as way of expediting analysis and report writing. To provide detailed feedback to the data science team between the first

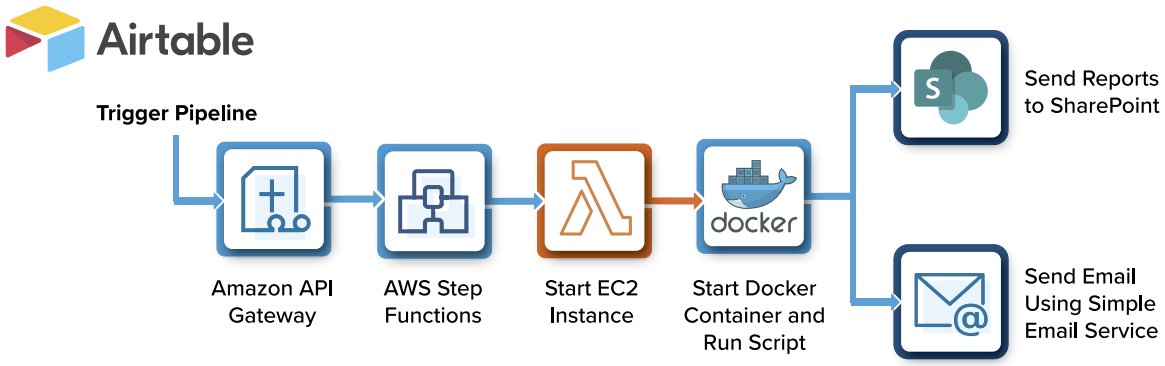

Figure 3: Report generation pipeline infrastructure.

and second years of data collection, five experienced qualitative researchers (led by the third author) conducted a quality control exercise on 10 randomly selected Year 1 transcripts. During this exercise, researchers:

1. Independently applied codes from the ILNA framework to their assigned transcripts, generating up to three findings statements and a summary statement for each indicator based on the coded text.
2. Compared their findings and summary statements for each indicator to those generated by the AI tool.
3. Described whether the AI's findings and summary statements were "accurate" (i.e., substantively similar to the human coder's findings and summary statements and conveyed useful information for report writing) or not.
4. Further described "inaccurate" AI-generated findings and summary statements as containing vague (and effectively unusable) statements, introducing hallucinations, or misrepresenting participants' meanings.

Future quality control processes will involve a more comprehensive, large-sample evaluation of Year 2 outputs to better assess accuracy, reliability, and practical usefulness for report writing. We outline this planned evaluation, including human-coding baselines, reliability metrics, and systematic comparison of AI outputs, in greater detail in the Section 5.

2.2 Report Generation Pipeline

This section outlines the design of the cloud-based architecture supporting the automated generation and distribution of final reports. The Report Generation Pipeline, as shown in Figure 3, was developed in response to growing challenges around scalability, consistency, and delivery time in producing customized reports for a large number of schools. Manually generating reports had become increasingly time-consuming and error-prone, especially as data sources and report templates evolved.

To address these limitations, the team implemented a secure, scalable, and reproducible infrastructure through Airtable and Amazon Web Services. The system integrates multiple cloud-native components including Lambda functions, Step Functions, Elastic Cloud Compute instances (EC2), and containerized execution environments into a cohesive, event-driven pipeline. By leveraging services such as Airtable for initiating the pipeline, EC2 for executing containerized report generation workloads in a controlled, reproducible environment, and SharePoint for

sharing the final reports, the pipeline minimizes manual intervention while maintaining traceability and operational resilience. Its modular design supports iterative updates to code, templates, and data sources without disrupting production workflows, enabling continuous improvement and adaptability over time.

2.2.1 Airtable Interface and Automation Trigger

Airtable, a cloud-based platform that combines the familiarity of a spreadsheet interface with the flexibility of a relational database, serves as the user-facing entry point for the report generation pipeline (Airtable Blog, 2022). Designed for ease of use by non-technical staff, subject-matter experts, and researchers without programming experience, Airtable allows users to interact with complex backend systems through an intuitive, low-code interface. Within the Airtable base, each record corresponds to a specific school or district and tracks the status of its associated report. A custom button field in each row enables users to trigger the automated reporting process directly from Airtable, eliminating the need to navigate external systems (Airtable, 2023).

Upon activation, the Airtable automation executes a webhook that transmits a payload containing relevant project parameters such as a unique school ID, to a designated endpoint on AWS API Gateway. This integration bridges the user-friendly Airtable interface with the backend processing infrastructure, enabling a seamless transition from manual inputs to automated workflows. Airtable's webhook-triggered automations provide an efficient low-code mechanism for connecting spreadsheet-style interfaces to backend processing tools without requiring custom front-end development (Airtable Help, 2025; Shen, 2024).

2.2.2 AWS API Gateway and Step Functions

When a user initiates report generation from the Airtable interface, AWS API Gateway serves as the secure, managed entry point to the backend, receiving webhook requests and routing them to downstream AWS services (Amazon Web Services, 2025a). In this pipeline, API Gateway validates and forwards requests to AWS Step Functions, a serverless orchestration layer that sequences tasks, manages dependencies, and provides built-in error handling, retries, and execution monitoring (Amazon Web Services, 2025d). The workflow includes: (1) starting a preconfigured EC2 instance for report generation, (2) verifying a successful boot, and (3) running containerized R and Python scripts to generate the report. This modular design enhances reliability, enables recovery from partial failures, and supports detailed progress tracking while leveraging AWS-native security, scalability, and maintainability.

2.2.3 EC2 Instance and Dockerized Report Scripts

Following orchestration via AWS Step Functions, the pipeline provisions an EC2 instance, AWS's scalable virtual server environment, configured specifically for running containerized report generation workloads (Amazon Web Services, 2025f). EC2 provides on-demand compute capacity with full control over the operating environment, making it well-suited for executing resource-intensive tasks such as multi-stage statistical processing and document rendering.

Upon startup, the EC2 instance authenticates with Amazon Elastic Container Registry to retrieve the latest version-controlled Docker image and launches the container (Amazon Web Services, 2025g). In this implementation, the image is configured to encapsulate all necessary dependencies, including R, Python, and LaTeX packages and libraries required for producing

high-quality reports. Containerization ensures environmental consistency, reproducibility, and portability across deployments. Within this isolated environment, a Python controller script orchestrates the full report generation process, including: fetching relevant data from Amazon RDS, local CSV files, and the Airtable API; merging structured survey results with qualitative findings from the AI processing pipeline; and executing RMarkdown templates to render reports in both PDF and Word formats using LaTeX styling. The system supports concurrent report generation for multiple schools or districts, enabling scalable processing under tight timelines.

2.2.4 SharePoint Upload and Email Notification

Once reports are rendered (described further in Subsection 2.3), the final outputs are automatically uploaded to designated SharePoint directories associated with each school or district. Microsoft SharePoint is a cloud-based document management and collaboration platform that enables secure storage, organization, and controlled sharing of files within an institution (Microsoft Corporation, 2025). In this pipeline, SharePoint serves as the centralized repository for distributing completed reports to stakeholders. Secure authentication is handled through Azure Key Vault, Microsoft's cloud service for securely managing encryption keys, passwords, and access credentials (Microsoft, 2025). This ensures compliance with institutional data governance protocols and protects sensitive information. The upload process is executed via the Microsoft Graph API, which provides programmatic access to SharePoint document libraries and enables precise placement of reports into their corresponding organizational folders (Microsoft, 2023).

Once reports have been uploaded to SharePoint, Amazon Simple Email Service, a scalable, cost-effective cloud email service, which sends automated notifications to the project team, confirming successful completion and availability of the reports, or alerting users if any errors occur during the process (Amazon Web Services, 2025c). At the same time, the Airtable API programmatically updates the corresponding record in the Airtable base to reflect the latest processing status. This maintains real-time transparency across the project's data pipeline, allowing both technical and non-technical team members to monitor progress and verify completion without manual intervention.

2.3 Needs Assessment Summary Report

The Needs Assessment Summary Report represents the primary analytical product generated by the broader report generation pipeline architecture. While the Report Generation Pipeline (Subsection 2.2) is responsible for infrastructure provisioning, orchestration, and execution, this subsystem, as shown in Figure 4, focuses on the automated assembly of the report itself. Once the necessary compute environment is provisioned via containerized infrastructure, the report generation process is initiated, comprising data extraction, transformation, template population, and document rendering.

2.3.1 Report Generation Script and Starting Docker Container

The report generation process is initiated within a containerized execution environment to ensure reproducibility, version stability, and dependency isolation. A dedicated Docker container, started based off the Docker image described in Subsubsection 2.2.3, serves as the execution unit. This instantiation triggers the primary report generation script, which sequentially invokes data extraction routines, performs transformation logic, and assembles the final report. By standardizing the execution environment, this approach eliminates inconsistencies arising from local

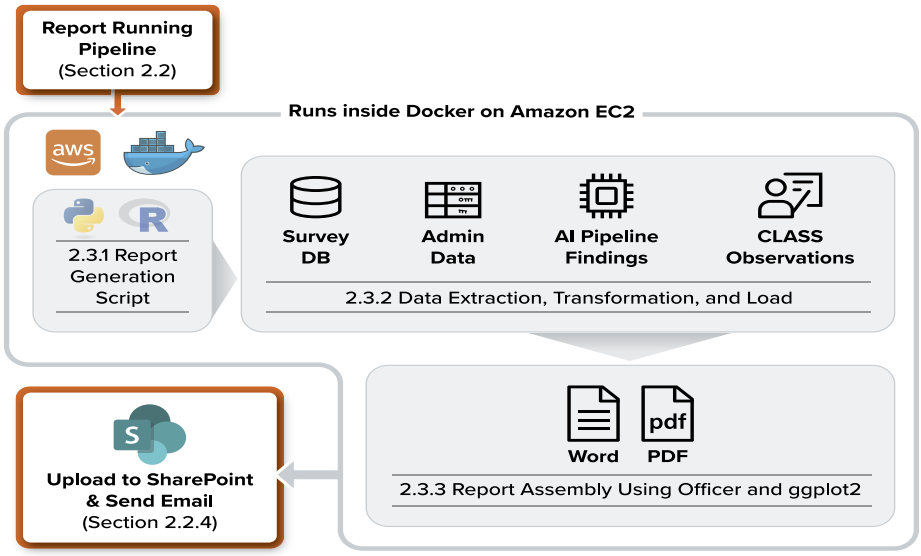

Figure 4: Automated report infrastructure.

development settings, facilitates rollback to prior versions, and aligns with best practices for computational reproducibility in applied research (Boettiger, 2015; Merkel, 2014).

2.3.2 Data Extraction, Transformation, and Loading

Following container initialization, the report generation script initiates an Extract–Transform–Load (ETL) sequence that consolidates heterogeneous data sources into an analytic structure optimized for report generation. The Needs Assessment Summary Report draws on multiple, complementary data streams to produce a comprehensive, evidence-informed portrait of school conditions. Quantitative inputs include structured educator survey responses, capturing perceptions aligned with established school diagnostic frameworks. These are supplemented with school-level administrative indicators such as assessment performance, attendance rates, and demographic profiles. To contextualize these measures, a peer-matching framework is applied, enabling benchmarking against schools with similar structural and demographic characteristics (Bryk et al., 2010).

Qualitative insights are integrated to enrich and contextualize the quantitative findings. These include researcher-reviewed summaries from the Audio Pipeline, which distill themes from structured interviews and focus groups, as well as classroom observation data collected using the Classroom Assessment Scoring System, a validated tool for assessing teacher–student interactions and instructional environments (Pianta et al., 2008).

The ETL workflow retrieves these diverse inputs from cloud-based relational databases, flat files, and external APIs (e.g., Airtable), then applies a series of automated transformation scripts to standardize formats, align variables with the report design schema, and resolve inconsistencies. The resulting harmonized datasets are stored as in-memory R objects, ready for direct integration into the templating functions. Executed entirely within the containerized environment, this process ensures all intermediate products remain ephemeral, thereby enhancing reproducibility.

2.3.3 Word Templates Using Officer and ggplot2

The final assembly of the Needs Assessment Summary Report leverages the officer package for programmatic manipulation of Microsoft Word documents and ggplot2 for high-quality data visualization (Gohel and Skintzos, 2023; Wickham, 2016). Leveraging word templates, preformatted with ILNA's standardized headings, styles, and placeholders, serves as the structural backbone of the report. Officer package work to replace these placeholders with dynamically generated tables, figures, and narrative text drawn from the transformed datasets from Subsubsection 2.3.2.

Quantitative visualizations (e.g., bar plots of survey scores, tables of school metrics) are produced via ggplot2 with consistent color palettes and labeling conventions to ensure interpretive clarity. The Word document also contains key findings and insights authored by researchers to contextualize quantitative results with qualitative evidence. Because the report is generated in Word format, researchers retain full flexibility to make targeted adjustments post-generation, such as refining narrative language or altering spacing and layout, without disrupting the overall structure. This hybrid approach combines the efficiency and reproducibility of automated generation with the adaptability needed for high-stakes reporting, where nuanced interpretation and tailored presentation are essential. After assembly, the finalized document is generated in both Word and PDF formats. The subsequent steps in the workflow, including upload and distribution, are detailed in Section 2.2.4.

2.3.4 Integrating Human Judgment into Automated Reporting

While the pipeline automates data ingestion, formatting, and document assembly, the interpretation of results, and the development of findings and recommendations, remains a human-centered process. All narrative content, summary findings, and evaluative judgments are authored and reviewed by AIR researchers, ensuring that outputs reflect a contextually grounded and evidence-informed understanding of each school's conditions. This human-in-the-loop design preserves the critical role of domain expertise, allowing researchers to interpret nuanced qualitative evidence, reconcile conflicting data points, and tailor recommendations to the unique circumstances of each site. By coupling the efficiencies of automation with the rigor of expert validation, the system delivers reports that are both consistent in structure and sensitive to local context, supporting high-quality, evidence-based decision-making in school improvement efforts.

3 System Performance Metrics

3.1 Audio Pipeline

The Audio Pipeline was developed to address longstanding inefficiencies in processing qualitative interview data at scale. Traditional approaches to transcription, speaker identification, thematic coding, and synthesis required substantial human labor and often delayed the availability of actionable insights. To improve efficiency and scalability, the pipeline automates each stage of processing while maintaining a human-in-the-loop review process to ensure interpretive accuracy. In the second year of implementation, 248 focus group audio files were processed for the 101 participating schools.

Early Performance Indicator 1 – Increased Processing Speed: Under the prior manual workflow, processing and coding a single transcript typically required 5–6 hours of concentrated researcher effort. With the introduction of the Audio File AI Pipeline, this acceleration

Table 1: Performance metrics for the year 2 audio processing pipeline.

Metric	**Value**
Average processing time per transcript	25 minutes
Median processing time per transcript	23 minutes
Range of processing times	15–48 minutes
Typical system throughput	2–3 concurrent audio files
Maximum parallel capacity	Up to 5 audio files
Estimated total processing time for 248 transcripts	75–125 system hours
Estimated manual effort for equivalent workload	1,200–1,500 person-hours

enables qualitative insights to be incorporated into the broader reporting process much earlier in the project timeline.

To contextualize these improvements at scale, Table 1 presents the full set of Year 2 performance metrics, including average and median processing times, system throughput, and estimated reductions in manual labor.

Early Performance Indicator 2 – Error Handling and Robustness: Pipeline logs show that:

- Automated success rate: The pipeline successfully processed approximately 96.7% of audio files without any manual intervention.
- Instances requiring manual intervention were attributable primarily to input-related issues, including corrupted audio, missing segments, or extreme multi-speaker overlap.
- To handle multi-speaker overlap, audio files were reviewed and processed manually for a total of roughly 40 hours of human effort.

This reduction in processing time enabled transcripts to be incorporated into the broader reporting workflow far earlier in the project timeline, supporting faster synthesis and delivery.

3.2 Report Generation Pipeline

The report generation pipeline was designed to address growing challenges in scalability, reliability, and usability in producing individualized school evaluation reports. In Year 1, the team manually generated 281 reports, each requiring nearly an hour of developer effort because the workflow had to be executed, monitored, and debugged.

Early Performance Indicator 1 – Increased Processing Speed: Automation produced a substantial improvement in processing efficiency. As summarized in Table 2, Year 2 runs demonstrated markedly faster report generation times compared to the prior manual workflow, enabling more rapid integration of findings into the overall reporting process.

To contextualize these improvements, Table 2 details key performance metrics, including average and median generation times, throughput capacity, and system-wide reductions in total processing hours.

Early Performance Indicator 2 – Improved Reliability: Automation substantially improved system reliability and robustness. Rigorous input validation and structured error handling ensure that common failure modes such as incomplete data, schema mismatches, or rendering errors are automatically detected, categorized, and communicated via a centralized alerting system. Operational logs from Year 2 show:

- Automated success rate: 94% of reports completed without intervention

Table 2: Performance metrics for the year 2 report generation pipeline.

Metric	**Value**
Average report generation time	12 minutes (vs. 60 mins manually)
Median report generation time	10 minutes
Range of generation times	6–14 minutes per report
Typical parallel throughput	2–5 concurrent reports
Estimated total time for 101 Year 2 reports	10–20 system hours
Estimated manual effort for equivalent workload	101 person-hours (1 hour per report)

- Instances requiring manual intervention were attributable primarily to upstream data issues such as incorrect or missing selection criteria in Airtable or misaligned template metadata.
- To handle scenarios of upstream data issues, these would be manually reviewed and fixed on a case by case basis for a total of roughly 2 hours of human effort.

3.3 Report Delivery Time Evaluation

Report delivery time is defined as the number of days between a school's final site visit and the delivery of its completed Needs Assessment Summary Report. In Year 1, the project team aimed to deliver reports within 6–8 weeks, but this target was rarely met; many reports required 10–12 weeks to finalize. These delays were shaped by several unmeasured but meaningful factors, including competing project timelines, the number of concurrent reports assigned to each researcher, and differences in the completeness and complexity of school-level data. Complete delivery-time records were available for 153 schools, while data for an additional 128 schools were incomplete, and this limitation is acknowledged when interpreting the Year 1 baseline.

Year 2 delivery times reflect both technological and procedural improvements. The automated report generation pipeline substantially reduced the time required to assemble and render reports, but the team also refined several human-driven components of the workflow. These included clearer expectations for editing and approval, more consistent internal review protocols, improved coordination across project roles, and a more stable shared understanding of the reporting process. As a result, Year 2 improvements represent the combined effect of automation and the maturation of project routines rather than a purely technological gain.

To evaluate the system's overall impact, we compared Year 1 and Year 2 report delivery times using standard performance metrics (see Table 3). This evaluation includes descriptive statistics, measures of distributional spread, and appropriate inferential tests to assess whether Year 2 reports were completed more rapidly and with greater consistency. Together, these metrics provide a rigorous empirical foundation for understanding the benefits of the automated workflow in the broader context of improved project processes.

The results indicate a substantial and statistically significant improvement in reporting efficiency from Year 1 to Year 2. Year 2 reports were completed approximately 19 days faster on average, and the median delivery time decreased by 22 days. Because the Year 1 and Year 2 samples represent independent groups rather than matched pairs, we used the Wilcoxon Rank–Sum test (equivalent to the Mann–Whitney U test) to compare distributions across years. The test confirms that the reduction in delivery time is highly unlikely to be attributable to random variation ($p < .0001$). A Welch two-sample t-test provides convergent evidence of a large and

Table 3: Report delivery time comparison between year 1 and year 2.

Measure	**Year 1**	**Year 2**
Mean (days)	69.9	51.2
Median (days)	71	49
SD (days)	13.8	14.0
IQR (days)	16	17
Min–Max (days)	32–102	28–80
Wilcoxon W	12,504 ($p < .0001$)	
Welch Two Sample t-test t-value	10.44 ($p < .0001$)	
95% CI for Difference	15.1 to 22.2 days	

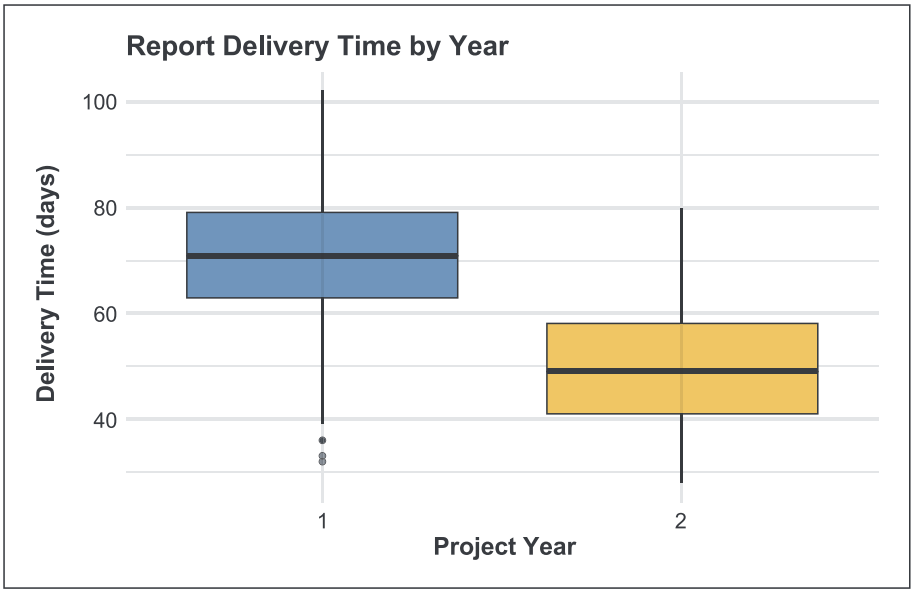

Figure 5: Box plot of report delivery time by year.

meaningful mean difference, with a 95% confidence interval of 15.1–22.2 days. Taken together, these results demonstrate that the Year 2 workflow not only accelerated report production but also increased consistency, as reflected in a narrower delivery range and fewer extreme delays.

Distributional improvements further illustrate the efficiency gains. In Year 1, only 7.3% of reports were delivered within 45 days and 19.9% within 60 days; none were completed within 30 days. In contrast, 4.9% of Year 2 reports were completed within 30 days, 36.6% within 45 days, and 76.2% within 60 days. All Year 2 reports (100%) were finalized within 90 days, compared with 95.4% in Year 1, indicating a meaningful reduction in late outliers. Figure 5 presents a boxplot of delivery times across years, highlighting both the downward shift in central tendency and the reduced variability in Year 2.

These results provide strong evidence that the combination of automated report generation and refined project workflows yielded substantial improvements in delivery time, reliability, and overall consistency of deliverables in Year 2. While the gains cannot be attributed solely to automation, the quantitative patterns suggest that the introduction of cloud pipelines meaningfully

reduced the operational bottlenecks that characterized the earlier manual workflow. Moreover, the reduced variability indicate a more stable and predictable reporting system, which is critical for maintaining timely communication with schools and supporting a coherent improvement process. These findings underscore the value of integrating automation with human-in-the-loop procedures, demonstrating that technological enhancements and refined organizational routines can jointly produce measurable improvements in large-scale mixed-methods research settings.

4 Discussion

Using cloud-native, AI-enabled pipelines for qualitative research tasks, specifically, Audio Pipeline processing and automated report generation, represents a substantial advancement in the methodological toolkit for qualitative researchers. Far from being purely technical upgrades, these systems redefine how large-scale, mixed-methods projects can be conducted, enabling new levels of efficiency, consistency, and analytical depth. By uniting advanced orchestration frameworks, containerized environments, and rigorous data governance practices, the pipelines move beyond streamlining workflows to establish an adaptable, transparent, and reproducible research infrastructure. This discussion examines the broader methodological, operational, and ethical implications of these design choices, with implications for a wide range of applied research domains. The following subsections reflect on the key architectural principles that underpin the system's performance and long-term viability.

4.1 Automation and Orchestration

A central goal across all pipelines was full automation from user input to final output. In the Audio Pipeline, automation is realized through AKS job orchestration and an internal controller script that manages task sequencing and logging. Similarly, the report generation pipeline uses AWS Step Functions and Lambda functions to create an end-to-end serverless automation framework (Maissen et al., 2020). These orchestrators handle conditional logic, error catching, task retries, and status updates, minimizing the need for human oversight. These systems integrate with user-friendly interfaces (e.g., Airtable or custom dashboards) to allow non-technical stakeholders to initiate complex workflows with minimal friction. Such automation strategies align with best practices in scalable, cloud-native system design (Oyeniran et al., 2024).

4.2 Reproducibility and Containerization

Reproducibility is achieved through the consistent use of containerized workloads across these systems. For the Audio Pipeline, each model and processing step is deployed as an AKS pod running a dedicated container image, enabling deterministic behavior across compute environments (Boettiger, 2015). For report generation, pre-configured Docker images built with R, Python, and LaTeX are pulled from Amazon Elastic Container Registry, ensuring that every report is generated in an identical software environment. This approach not only supports reproducibility but also simplifies onboarding, version rollback, and collaborative debugging (Peng, 2011).

4.3 Secure Access and Auditability

Security and auditability were central design priorities due to the sensitivity of school-level data and focus group transcripts. Authentication credentials for accessing APIs and services are

managed using Azure Key Vault and AWS Secrets Manager, which securely store secrets, certificates, and cryptographic keys with role-based access controls and hardware-backed protections (Microsoft, 2025). These tools help minimize exposure of sensitive tokens while maintaining centralized credential governance. Outputs are stored in encrypted Amazon S3 buckets, with access restricted via IAM roles and tightly scoped bucket policies that follow best practices such as least-privilege access and versioning (Amazon Web Services, 2025e). For report uploads to SharePoint, the pipeline uses the Microsoft Graph API authenticated through secrets stored in Azure Key Vault (Microsoft, 2023). These safeguards support institutional data governance policies while enabling traceability and audit readiness across the pipeline.

4.4 Ethical and Privacy Considerations

Deploying AI-driven pipelines in educational settings raises important ethical and privacy considerations, particularly when handling sensitive data such as focus group recordings, survey responses, and school performance metrics. This project prioritized responsible data stewardship, transparency, and safeguards for ethical AI use throughout the pipeline's lifecycle. All data were collected under formal agreements in line with AIR's Institutional Review Board (IRB) guidelines, with cloud access restricted through organizational authentication, role-based permissions, and session limits. Core components used Microsoft Azure OpenAI services (Whisper for transcription, GPT-4o for language understanding), with strict controls to ensure that inputs and outputs were neither shared with OpenAI, used to train foundation models, nor accessible to other customers (Microsoft, 2024). Within Azure environments, data remained the property of the organization, under its sole control, and could be deleted at any time (Microsoft, 2024).

To mitigate bias and accuracy risks, trained researchers systematically evaluated AI-generated transcripts, summaries, and thematic codes for accuracy, relevance, and alignment with source materials. This review was paired with iterative system and prompt refinements, incorporating targeted examples, clarified instructions, and controlled output formats to minimize the likelihood of hallucination, overgeneralization, or unsupported claims. Feedback from each review cycle informed prompt adjustments and system configuration, creating a feedback loop to improve system reliability over time.

Looking ahead, future iterations should incorporate user experience studies to assess stakeholders' understanding of data collection, usage, and interpretation. Expanding transparency efforts such as publishing clear documentation of model behavior, limitations, and overall process, will strengthen trust among participants and institutional partners. Continued alignment with Responsible AI frameworks is essential to ensuring that the pipeline remains ethically sound, socially acceptable, and adaptable as technologies, regulations, and community expectations evolve (Ghimire and Edwards, 2024; Chaudhry et al., 2022).

4.5 Leveraging Human Expertise in Automated Systems

AI-assisted tools substantially accelerate the early stages of qualitative analysis and report production, allowing AIR researchers to devote more time to the relational and interpretive work that cannot be automated. This includes building and maintaining strong relationships with school and district leaders, conducting multi-source qualitative data collection, integrating findings across interviews, surveys, and observations, and developing school-specific feedback grounded in evidence-based improvement frameworks. The ILNA project team brings extensive expertise in educational policy, school improvement, qualitative research methods, and real-

world experience in K–12 settings. This content knowledge ensures that each step of the needs assessment process, from initial engagement with school leaders to the communication of final findings, is responsive, asset-based, and tailored to each school's unique context.

While automated pipelines efficiently perform the initial tasks of transcription, speaker role identification, categorization, and summarization, human judgment remains central to interpreting these outputs and integrating them into a coherent understanding of school conditions. Researchers draw on their methodological training and on-the-ground experience to evaluate the accuracy of AI generated findings, reconcile inconsistencies, and synthesize qualitative data with survey and observational evidence to determine school ratings. They also ensure that written reports authentically reflect each school's voice and context, a level of nuance that requires professional judgment and cannot be replicated by automated tools alone.

Taken together, the use of automation and human-in-the-loop review introduces several methodological considerations. Automated pipelines enhance efficiency, consistency, and scalability, but human involvement remains essential for validating AI outputs, identifying when the system has misinterpreted contextual cues, ensuring that ratings are grounded in multiple forms of evidence, and grounding school experiences within the reports that are credible and actionable for educators. Rather than replacing human expertise, the AI tools serve as accelerators that free researchers to focus on the higher-order interpretive work that underpins rigor, reliability, and contextual relevance in school-based qualitative research.

4.6 Generalizability, Reuse, and Broader Implications

While the current implementation is tailored for school improvement reporting, the modular architecture and cloud-native design principles are inherently domain-agnostic. Components such as serverless orchestration, containerized analytic environments, and Airtable-triggered automation can be adapted for other applied research contexts, and not just educational research. This system provides a foundation for replication and adaptation, extending the benefits of automation and reproducibility across diverse domains. More broadly, these pipelines illustrate how cloud infrastructure and AI models can function as methodological enablers that support scale, transparency, and rigor in research practice. By reducing technical friction and enhancing reproducibility, they lay the groundwork for open and collaborative infrastructures capable of evolving alongside changing analytical needs. As computational methods continue to redefine the social sciences, such architectures will be increasingly important for advancing timely, equity-focused, and evidence-based decision making across sectors.

5 Limitations and Future Work

The proposed system architectures for AI-driven qualitative analysis and automated report generation show promise in streamlining educational research workflows. However, several limitations remain and present opportunities for continued refinement and methodological advancement.

5.1 Audio Pipeline

Despite developing a quality control protocol for the AI model, qualitative researchers and data science team members were unable to regularly implement this process throughout Year 2 due to time and budget constraints. Conducting a formalized evaluation study of the audio-to-insight

pipeline remains a top priority for future work. A comprehensive assessment would draw on a larger, more representative sample of transcripts; include dual independent human coding to establish a reliability baseline; and compare AI-generated codes, findings statements, and summaries against this human-coded "gold standard." Calculating reliability metrics such as percent agreement, Cohen's κ, and Krippendorff's α, alongside documenting common error types (e.g., omissions, hallucinations, or misinterpretations), would provide empirical evidence of accuracy, reliability, and practical usefulness. Such an evaluation would help identify specific failure points, guide targeted refinements to model configuration and workflow design, and strengthen stakeholder confidence in the pipeline's methodological rigor.

The Audio Pipeline, though operationally robust, has sources of variability and modeling limitations that warrant further exploration. One challenge concerns the accuracy of text-based speaker role identification and attribution. Although the current implementation uses Whisper for transcription and GPT-4o and custom prompts for text-based speaker role attribution, nuanced errors persist, especially in overlapping speech or environments with poor acoustic quality. Future work could include incorporating acoustic diarization models such as pyannotate (Bredin and Laurent, 2021), confidence-weighted attribution for downstream tasks (Gal and Ghahramani, 2016), or LLM-based speaker diarization post-processing (Wang et al., 2024).

The transcript tagging process requires further refinement. Although the process was designed to recognize specific questions and topics from interview protocols, qualitative data collection is necessarily flexible and responsive to the research context. Researchers may ask questions out of order to follow the flow of the conversation, rephrase questions, and pose follow-up questions to obtain detailed context-specific information. Future work could explore alignment algorithms and context-aware tagging strategies that can better infer thematic links from nonlinear dialogue sequences (Xiao et al., 2023). This is especially crucial for focus groups given the number of respondents.

Another key limitation stemmed from aligning the pipelines with existing project infrastructure, requiring integration across multiple cloud providers. The Audio Pipeline uses AWS S3 and RDS for storage and retrieval, and AKS for scalable orchestration. This hybrid architecture, driven by pre-existing tools and systems, enabled interoperability with legacy workflows but added complexity in deployment, authentication, and maintenance. Operating across platforms requires duplicated security settings, network permissions, and monitoring, increasing operational overhead and potential failure points (Brans, 2023; Fehling et al., 2014). A unified cloud stack could simplify management and reduce cross-provider latency, and future work should explore migration strategies to streamline efficiency. Such efforts should also consider cost, security, and scalability implications to ensure that consolidation aligns with long-term project needs.

The system is designed to remain adaptable as AI models evolve by leveraging a modular architecture and the Microsoft Azure OpenAI API, which allows underlying language models to be upgraded or replaced without altering the overall pipeline structure. While the current prompts and configurations have been calibrated for the specific models used in Year 2, we anticipate that new model releases or model deprecations will require further evaluations to ensure continued accuracy and stability. To maintain analytic quality, future updates will involve running candidate models on a validation set to compare performance, assess error patterns, and identify any shifts in hallucination behavior or interpretive reliability. Findings from these evaluations will guide iterative refinements to prompts, templates, and component settings, enabling the system to incorporate new capabilities while preserving reproducibility and methodological rigor.

5.2 Report Generation Pipeline and Needs Assessment Summary Report

The report generation pipeline, while operationally stable and functionally complete, currently exhibits processing latency that constrains its responsiveness in iterative or high-volume use cases. In this context, latency refers to the total elapsed time between the initiation of a report generation request and the delivery of a completed output.

The current pipeline architecture assumes that each run produces a complete report from start to finish. In practice, there are frequent scenarios in which users wish to rerun only specific sections of a report, such as when a single component is updated. Adding support for granular reruns would significantly improve user experience, computational efficiency, and agility in project workflows. This functionality would require enhancements to the orchestration logic, as well as finer-grained state tracking and conditional task execution within the pipeline (Fehling et al., 2014).

Although initial feedback from school and district leaders was positive, the evidence available to assess user perceptions of the Year 2 reports is limited. AIR administered a brief feedback survey to state and district partners, but due to substantial policy changes during this period, only eight respondents completed it. While the average rating of 6 out of 10 on the School Report & Debrief component and comments such as "The report was very thorough and captured the majority of the current state of our school" provide encouraging indications of perceived quality, the small sample size limits the generalizability of these findings. Future work should include a more systematic and well-resourced data collection process to evaluate stakeholder satisfaction, usability, and decision-making impact at scale.

6 Conclusion

This paper has presented the design and implementation of three interdependent automation pipelines, one for AI-assisted analysis of focus group audio, one for the automated generation of the report in a cloud environment, and one for the generation of the final report itself, developed to advance qualitative research and technical assistance. These systems were created in response to practical constraints experienced during large-scale school evaluations, where time-consuming manual workflows delayed the delivery of findings critical to school planning and improvement. By addressing these bottlenecks, the pipelines help fulfill the original goals of the initiative: to provide timely, context-aware, and equity-focused insights to schools in need of comprehensive support. Looking ahead, this work offers a model for how educational research infrastructure can evolve. Using automation not as an end, but to increase operational resilience and enable timelier, data-informed decision-making across educational systems.

Together, the Audio Pipeline, Report Generation Pipeline, and Needs Assessment Summary Report, demonstrate how recent advances in automation and generative AI can be harnessed not just for efficiency, but to deepen methodological rigor and promote access. The Audio Pipeline enables structured extraction of stakeholder perspectives at a scale unachievable through manual coding alone, while the Report Generation Pipeline and Needs Assessment Summary Report reduce the timeline for schools to receive actionable feedback.

Critically, these tools were not built to replace human judgment, but to enhance it. Their design reflects a human-centered approach, supporting researcher oversight, embedding validation steps, and maintaining flexibility to accommodate local context. In doing so, they align with the broader goals outlined at the start of this paper: to build systems that are both analytically robust and responsive to the lived realities of schools.

Supplementary Material

The supplementary material includes a GitHub repository with two subfolders—'aiPipeline-SDSS2025' and 'autoreportsPipeline-SDSS2025'—corresponding to the Audio Pipeline and the Report Generation Pipeline + Automated Report described in the manuscript. While the original implementations relied on secure cloud infrastructure, the materials provide insight into system architecture, key processing steps, and expected outputs. Included are mock data, configuration examples, prompts, crosswalks, and selected code, enabling users to review and execute sample scripts to understand each pipeline stage. The supplementary materials also include an additional R Markdown (RMD) file that demonstrates report generation using synthetic school-level data, illustrating how qualitative and quantitative inputs are combined within the automated reporting workflow. Due to reliance on internal systems and proprietary authentication, some components (e.g., secure dataset access, organizational credentials, private APIs) are non-functional outside production. Code exposing security-sensitive logic or deployment details has been removed, but the materials still convey the overall design and practical implementation. Additional documentation in each folder guides navigation of outputs. See https://github.com/gchickering21/SDSS2025_materials for files and documentation.

Acknowledgement

The authors would like to thank the ILNA project team, technical contributors, and research staff who provided critical support throughout the development of the pipelines and this paper.

Funding

This work was funded by the Illinois State Board of Education and supported by AIR.

References

Airtable (2023). Airtable api documentation. https://airtable.com/api. [Online; accessed 23 November 2025].

Airtable Blog (2022). How low and no-code tools increase productivity by breaking silos. https://blog.airtable.com/the-promises-low-code-platforms-should-deliver/. [Online; accessed 23 November 2025].

Airtable Help (2025). When webhook received trigger. https://support.airtable.com/docs/when-webhook-received-trigger. [Online; accessed 23 November 2025].

Amazon Web Services (2025a). Amazon api gateway features. https://aws.amazon.com/api-gateway/features/. [Online; accessed 23 November 2025].

Amazon Web Services (2025b). Amazon rds features. https://aws.amazon.com/rds/features/. [Online; accessed 23 November 2025].

Amazon Web Services (2025c). Amazon simple email service (ses). https://aws.amazon.com/ses/. [Online; accessed 23 November 2025].

Amazon Web Services (2025d). Aws step functions. https://aws.amazon.com/step-functions/. [Online; accessed 23 November 2025].

Amazon Web Services (2025e). Security best practices for amazon s3. https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html. [Online; accessed 23 November 2025].

Amazon Web Services (2025f). What is amazon ec2? https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html. [Online; accessed 23 November 2025].

Amazon Web Services (2025g). What is amazon elastic container registry (ecr)? https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html. [Online; accessed 23 November 2025].

Boettiger C (2015). An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1): 71–79. https://doi.org/10.1145/2723872.2723882

Brans P (2023). Building seamless data pipelines in a hybrid cloud environment. CMSWire. [Online; accessed 23 November 2025].

Bredin H, Laurent A (2021). End-to-end speaker segmentation for overlap-aware resegmentation. [Online; accessed 23 November 2025].

Bryk AS, Gomez LM, Grunow A, LeMahieu PG (2010). *Learning to Improve: How America's Schools Can Get Better at Getting Better*. Harvard Education Press.

Center for Computation and Visualization (2025). Speech-to-text models. https://docs.ccv.brown.edu/ai-tools/services/transcribe/speech-to-text-models. [Online; accessed 23 November 2025].

Chaudhry MA, Cukurova M, Luckin R (2022). A transparency index framework for ai in education. [Online; accessed 23 November 2025].

Corbett J, Redding S (2017). Using needs assessments for school and district improvement: A tactical guide. Council of Chief State School Officers and Center on School Turnaround at WestEd. [Online; accessed 23 November 2025].

Fehling C, Leymann F, Retter R, Schupeck W, Arbitter P (2014). *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer.

Gal Y, Ghahramani Z (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Balcan M.-F, Weinberger K. Q (eds), *Proceedings of the 33rd International Conference on Machine Learning*, 48, 1050–1059. Proceedings of Machine Learning Research.

Ghimire A, Edwards J (2024). From guidelines to governance: A study of ai policies in education. [Online; accessed 23 November 2025].

Glenn ML, Strassel SM, Lee H, Maeda K, Zakhary R, Li X (2010). Transcription methods for consistency, volume and efficiency. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Rosner M, Tapias D (eds), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), Valletta, Malta.

Gohel D, Skintzos P (2023). officer: Manipulation of microsoft word and powerpoint documents (r package version 0.6.2). https://CRAN.R-project.org/package=officer. [Online; accessed 23 November 2025].

Lane B, Unger C, Souvanna P (2014). Turnaround practices in action: A three-year analysis of school and district practices, systems, policies, and use of resources contributing to successful turnaround efforts in massachusetts' level 4 schools. http://www.mass.gov/edu/docs/ese/accountability/turnaround/practices-report-2014.pdf. [Online; accessed 23 November 2025].

Maissen P, Felber P, Kropf P, Schiavoni V (2020). FaaSdom: A benchmark suite for serverless computing. In: Charfi A, Cugola G, Pietzuch P, Jerzak Z (eds), *Proceedings of the 14th ACM*

International Conference on Distributed and Event-Based Systems (DEBS 2020). Association for Computing Machinery (ACM).

Merkel D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014, 239: 2.

Metabase Inc (2025). Metabase: An open-source business intelligence platform. https://www.metabase.com/. [Online; accessed 23 November 2025].

Microsoft (2023). Microsoft graph api overview. https://learn.microsoft.com/en-us/graph/overview. [Online; accessed 23 November 2025].

Microsoft (2024). Data, privacy, and security for azure openai service. https://learn.microsoft.com/en-us/azure/ai-foundry/responsible-ai/openai/data-privacy?tabs=azure-portal. [Online; accessed 23 November 2025].

Microsoft (2025). Azure key vault security features. https://learn.microsoft.com/en-us/azure/key-vault/general/security-features. [Online; accessed 23 November 2025].

Microsoft Corporation (2025). What is sharepoint? https://support.microsoft.com/en-us/sharepoint. [Online; accessed 23 November 2025].

Miles M, Huberman AM, Saldana J (2014). *Qualitative Data Analysis*. Sage Publications Ltd., 3 edition.

Miles MB, Huberman AM, Saldaña J (2019). *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications, Thousand Oaks, CA, 4 edition.

Nascimento RS, Silva AL, Rocha IA, Almeida JJ, Gonçalves G, Santos A, et al. (2024). Availability, scalability, and security in the migration from on-premises systems to azure kubernetes service: A proof of concept. *Computers*, 13(8): 192. https://doi.org/10.3390/computers13080192

Ogeawuchi JC, Uzoka A, Alozie CE, Agboola OA, Owoade S (2022). Next-generation data pipeline automation for enhancing efficiency and scalability in business intelligence systems. *International Journal of Social Science Exceptional Research*, 1(1): 277–282. https://doi.org/10.54660/IJSSER.2022.1.1.277-282

OpenAI (2023). Gpt-4 technical report. arXiv preprint: arXiv:2303.08774.

Oyeniran O, Misra S, Fernández-Sanz L, Damasevicius R (2024). A comprehensive review of leveraging cloud-native technologies for scalability and resilience in software development. *International Journal of Science and Research Archive*, 12(1): 541–549.

Pan J, Walston J, Therriault SB (2021). Relationship between state annual school monitoring indicators and outcomes in massachusetts lowest performing schools, *Technical Report REL 2021-085, Regional Educational Laboratory Northeast & Islands*.

Peng RD (2011). Reproducible research in computational science. *Science*, 334(6060): 1226–1227. https://doi.org/10.1126/science.1213847

Pianta RC, La Paro KM, Hamre BK (2008). *Classroom Assessment Scoring System™: Manual K–3*. Paul H. Brookes Publishing Co.

Radford A, Kim JW, Xu T, Brockman G, McLeavey C, Sutskever I (2022). Robust speech recognition via large-scale weak supervision. arXiv preprint: arXiv:2212.04356.

Shen J (2024). Understanding airtable webhooks and their applications. https://shortcuts.sequentialroutine.com/blog/understanding-airtable-webhooks-applications/. [Online; accessed 23 November 2025].

US Department of Education (2001). Comprehensive needs assessment guidebook. https://www.ed.gov/sites/ed/files/admins/lead/account/compneedsassessment.pdf. [Online; accessed 23 November 2025].

Voicegain (2023). Practical considerations for voice developers considering openai's whisper

asr. https://www.voicegain.ai/post/practical-considerations-for-voice-developers-considering-openais-whisper-asr. [Online; accessed 23 November 2025].

Wang Q, Huang Y, Zhao G, Clark E, Xia W, Liao H (2024). DiarizationLM: Speaker diarization post-processing with large language models. In: *Proceedings of Interspeech 2024*. International Speech Communication Association (ISCA).

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.

Xiao Z, Yuan X, Liao QV, Abdelghani R, Oudeyer PY (2023). Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. [Online; accessed 23 November 2025].