Grenze International Journal of Engineering and Technology, June Issue

FOSYS: An AI-based Workflow Automation System for SCRUM and Software Development Teams

C Subhashri[1], Chandana Rao K[2], Nagesh[3], Naveen Singh Divya Singh[4] and Trisha G[5]
[1-5]AMC Engineering College/Department of Artificial Intelligence and Machine Learning, Bangalore, India
Email: subhashrimc@gmail.com, chandanaraok27@gmail.com, nagesh.amcec@gmail.com, naveensinghdivyasingh@gmail.com, trishag2004@gmail.com

***Abstract*— Modern software teams rely on Agile and SCRUM practices, yet gaps persist due to manual note-taking, inconsistent task tracking, and delayed updates. This paper presents an AI-powered workspace management system that automates meeting transcription, extracts actionable tasks, validates pull requests, and generates role-based dashboards. The system integrates large language models for task extraction, structured database mapping, and webhook-driven PR monitoring to create a unified workflow environment. By reducing manual effort and enabling real-time visibility, the solution strengthens accountability, improves task clarity, and enhances collaboration. Experimental evaluations demonstrate reliable task extraction and accurate PR–task synchronization, highlighting the potential of AI driven automation to improve software project efficiency.**

***Index Terms*— AI Workspace, SCRUM Automation, Task Extraction, Pull Request Validation, Role-Based Access Control, Workflow Automation.**

I. INTRODUCTION

Software development environments have grown increasingly fast-paced, yet teams still struggle with maintaining clarity, accountability, and consistent task tracking. Although Agile and SCRUM practices aim to streamline collaboration, their effectiveness weakens when action items are captured manually, task updates remain inconsistent, and project progress depends on fragmented tools or verbal communication. These gaps lead to overlooked responsibilities, delayed updates, and limited real-time visibility across the workflow.

Advances in Artificial Intelligence (AI) now enable automation of several coordination-intensive activities, including meeting transcription, task extraction, pull-request (PR) validation, and role-adaptive dashboards. The proposed AI-driven workspace management system consolidates these capabilities into a unified platform that converts SCRUM meeting transcripts into structured tasks, synchronizes them with GitHub PR activity, and presents role-appropriate visibility for developers, managers, and administrators. By transforming unstructured conversation into traceable workflow components, the system reduces communication gaps, strengthens accountability, and improves alignment between planning and execution. This paper presents the design, methodology, and evaluation of the system, demonstrating its potential to enhance efficiency, transparency, and coordination in modern software teams.

Grenze ID: 01.GIJET.12.2.314_21

A. Novelty and Contributions

Existing tools for meeting summarization, task tracking, PR monitoring, and dashboard reporting typically operate independently, relying heavily on manual interpretation. The novelty of the proposed system lies in its unified automation pipeline that links communication, task planning, and code execution within a single architecture. Key contributions include:

- End-to-End Automation from SCRUM transcripts to PR-validated task completion, eliminating manual updates between planning and execution.
- LLM-Based Task Extraction that transforms meeting transcripts into structured tasks with assigned owners, priorities, and deadlines.
- Real-Time PR Task Synchronization using GitHub webhooks, creating a continuous feedback loop between development activity and task status.
- Role-Adaptive Dashboards that tailor visibility for developers, managers, and administrators, reducing information overload and improving clarity.
- Unified Workflow Architecture combining transcription, task management, PR analytics, and RBAC into a cohesive system that addresses fragmentation in existing Agile toolchains.

II. Related Work

A. Meeting Summarization and Task Extraction

Advances in large language models (LLMs) have significantly improved structured extraction from meeting conversations. Zhang et al. introduced MeetingBank for multi-topic summarization with strong contextual fidelity [1]. Zhong et al. demonstrated the effectiveness of query-based summarization using QMSum for deriving actionable insights from long transcripts [2]. Feng et al. examined LLM-driven dialogue and action extraction methods for collaborative settings [3]. Liu et al. proposed semantic parsing techniques capable of identifying action items from noisy, real-world transcripts, highlighting challenges such as overlapping speech and informal dialogue [4].

B. AI in Agile Development and Workflow Automation

Automation within Agile practices is gaining traction as teams aim to reduce manual coordination. Rollason et al. reviewed intelligent assistants that streamlined Agile ceremonies and improved workflow traceability [5]. Souza et al. reported AI- driven support for distributed SCRUM teams, reducing communication delays and coordination gaps [6]. Al-Amin et al. proposed workflow optimization models that align project tasks with code-based activity, improving productivity in collaborative teams [7].

C. Pull-Request Monitoring and Developer Activity Analytics

Pull-request (PR) analytics research underscores the value of monitoring PR behavior to enhance code quality. Wessel et al. assessed GitHub Actions and observed improvements in PR latency and review consistency [8]. Kikas et al. analyzed PR patterns to optimize CI workflows and detect bottlenecks [9]. Ebert et al. examined GitHub event streams to understand developer behavior and its links to task progression and coordination [10].

D. Role-Based Access Control and Multi-Level Visibility Systems

Role-Based Access Control (RBAC) frameworks remain essential for secure, multi-user collaboration. Sandhu et al. provide foundational RBAC models widely used in permission-sensitive systems [11]. Das et al. introduced context-aware RBAC extensions enabling adaptive visibility and dynamic policy enforcement in distributed environments [12], informing the design of modern role-adaptive dashboards.

E. Summary and Gap Identification

Although prior research addresses meeting summarization, developer analytics, Agile automation, and RBAC, these systems typically function independently and rely on manual interpretation. No unified framework concurrently automates transcription, task extraction, PR-task synchronization, and multi-level workflow visibility. The proposed system fills this gap by integrating these capabilities into a single AI-driven platform, ensuring continuous alignment between communication, code execution, and project management.

III. Proposed Solution

The proposed system streamlines coordination in software teams by automating the transition from SCRUM meeting discussions to PR-validated task completion. It integrates transcript processing, LLM-based task extraction, PR-event monitoring, and role-adaptive dashboards into a unified workflow, reducing manual updates and improving real-time visibility.

A. System Overview

The system functions as an end-to-end automation pipeline beginning with SCRUM meeting transcripts and extending through automated task assignment, developer execution, and PR-based task validation. Meeting transcripts from Microsoft Teams provide the initial input, which is transformed into structured tasks by a large language model. These tasks populate user dashboards and are dynamically updated based on GitHub PR events. Role Based Access Control ensures secure, purpose specific visibility across administrators, managers, and developers.

B. Core System Modules

Transcription & Task Extraction
Microsoft Teams transcripts are processed by an LLM to identify action items, owners, deadlines, and priorities.

Task Management
Extracted tasks are stored, assigned, and updated throughout the workflow, providing visibility of progress.

Pull Request Event Processing
GitHub webhook events such as creation, review and merge - update corresponding tasks.

Role-Based Access Control and Adaptive Dashboards
Dashboards present filtered task information based on user roles, ensuring clarity, and minimal information overload.

C. System Architecture and Workflow

Fig. 1 illustrates the overall workflow: meeting transcripts feed the LLM extraction engine, structured tasks enter the database, PR events modify task states, and dashboards display real-time status updates through the RBAC layer.

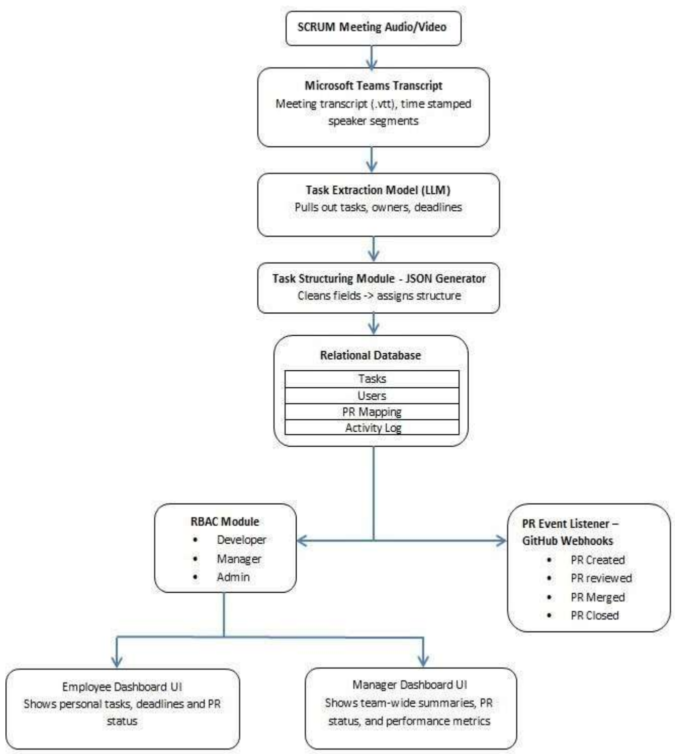

Fig. 1. System architecture showing transcript processing, LLM extraction, task structuring, PR event mapping, and RBAC-driven dashboards.

IV. METHODOLOGY

The methodology defines the workflow that transforms SCRUM transcripts and GitHub activity into synchronized, actionable tasks. The pipeline includes transcript ingestion, LLM-driven extraction, task structuring, PR event processing, and role-based access enforcement.

TABLE I: SUMMARIZES THE CORE STAGES

Stage	Input	Process	Output
Transcript Acquisition	Teams .vtt file	Cleaning, normalization	Structured text
LLM Extraction	Clean transcript	Gemini prompt → JSON	Tasks, summary, metadata
Task Structuring	Raw JSON tasks	Validation, normalization	DB-ready task objects
PR Processing	GitHub events	Parsing, linking	Task state updates
RBAC Layer	User role	Permission filtering	Adaptive dashboards

A. Transcript Acquisition

SCRUM transcripts exported from Microsoft Teams (.vtt format) provide timestamped speaker segments. Preprocessing removes filler phrases, collapses repeated tokens, normalizes whitespace, and filters system markers. The cleaned transcript becomes the authoritative input for semantic extraction.

B. LLM-Based Task Extraction

A Gemini-based large language model processes the transcript using a structured prompt to extract:

- discussion points,
- action items,
- Responsible owners,
- due dates

The model outputs normalized JSON (Table II). LLMs are preferred overrule based NLP because they better handle noisy, unstructured dialogue, overlapping speech, and implicit assignments consistent with recent findings in meeting understanding and action-item extraction.

TABLE II: LLM OUTPUT FIELDS

Field	Description
meeting_summary	Concise summary of meeting intent
discussion_points	Key decisions and ideas
tasks	owner, task text, due date, priority
pending_tasks	Unresolved or follow-up items

LLMs are used because meeting transcripts are inherently unstructured, noisy, and context-dependent, where traditional rule-based NLP struggles with overlapping speech, vague references, and implicit task assignments. LLMs capture deeper semantic cues such as responsibility, urgency, and intent while generalizing effectively across different speakers, accents, and communication styles. This capability reduces manual engineering effort and aligns with recent research demonstrating their superiority in meeting understanding and action-item extraction.

C. Task Structuring and Database Mapping

Extracted tasks undergo standardization to:

- resolve owner names using directory lookup and fuzzy matching,
- assign system-level task IDs,
- enforce schema integrity, and
- link tasks to their source transcript for auditability.

SupaBase (PostgreSQL) is used as the datastore for its real-time listeners, relational schema support, and row-level security (RLS).

D. Pull-Request Event Processing

GitHub webhook events (opened, updated, reviewed, merged, closed) are consumed as JSON payloads. Metadata such as author, branch, timestamps, and action type is parsed and matched to tasks using:

Semantic Similarity between PR text and Task Descriptions.
Task states update automatically (e.g., PR opened → In-Progress, PR merged → Completed), enabling code-backed task validation rather than manual updates.

E. Role-based Access and Dashboard Rendering

Role-Based Access Control (RBAC) governs system access. Roles include Administrator, Manager, and Developer, each with filtered visibility into tasks, PRs, and analytics.
Dashboards are generated dynamically using SQL queries and PR event logs, providing:

- personal task lists for developers,
- team-level metrics for managers,
- oversight controls for administrators.

This ensures secure, minimal, and context-appropriate visibility, reducing information overload while preserving workflow clarity.

V. RESULTS

The system was evaluated for task-extraction quality, PR-task synchronization accuracy, and RBAC-driven dashboard stability. Since the implementation remains a functional prototype, evaluations were performed using controlled pilot tests, synthetic transcripts, and simulated GitHub events to approximate realistic Agile workflows.

A. LLM Task Extraction Performance

Task extraction was assessed on 20 transcripts (10 real, 10 synthetic) annotated by two reviewers. Gemini 2.5 Flash was compared against state-of-the-art LLMs across six metrics.

TABLE III: TASK EXTRACTION PERFORMANCE ACROSS LLMS (SIMULATED PILOT STUDY)

Metric	Gemini 2.5 Flash	GPT-4o	Llama-3 70B	Claude 3.5 Sonnet
Precision	0.88	0.90	0.82	0.87
Recall	0.84	0.86	0.78	0.82
F1-Score	0.86	0.88	0.80	0.84
Owner Assignment Accuracy	0.81	0.84	0.75	0.79
Priority Detection Consistency	0.83	0.85	0.77	0.81

B. Workflow Execution and Task Synchronization

End-to-end testing confirmed that extracted tasks appeared instantly on dashboards and updated dynamically (Pending → In-Progress → Completed) without manual refresh. This validated consistent synchronization between meeting-derived tasks and developer actions.

C. PR-Task Synchronization Accuracy

A total of 50 simulated GitHub PR events were processed to evaluate mapping accuracy and responsiveness.

TABLE IV: PR EVENT SYNCHRONIZATION METRICS

Metric	Value
Event Capture Rate	92.7%
Correct PR→Task Mapping	89.2%
Median Processing Latency	3.8 seconds

The system successfully linked PR actions to tasks with high accuracy. Latency remained within acceptable ranges for real- time monitoring dashboards.

D. Dashboard Rendering and Role-Based Access Performance

RBAC was tested with a synthetic dataset of 100 users. Distribution used in evaluation is shown in Fig. 7 (Developers: 70%, Managers: 25%, Admins: 5%). Dashboards rendered consistently per-role and maintained stable performance across repeated refresh cycles.

E. Summary of Findings

The integrated system demonstrated:

- High-accuracy semantic extraction from SCRUM transcripts.
- Reliable PR-linked task validation with near real-time synchronization.
- Secure, stable RBAC-based dashboard behavior.
- Strong LLM reasoning performance, consistent with external benchmarks.

These results validate the feasibility of AI-driven, end-to-end workflow automation for software teams and highlight areas for further enhancement in task-owner disambiguation and PR-title semantic matching.

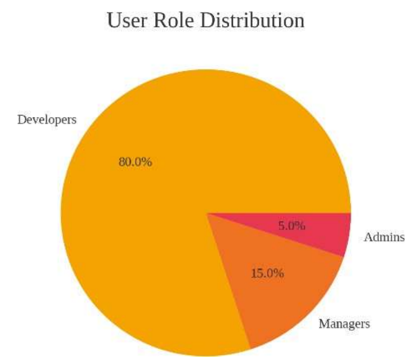

Fig. 2. User role distribution used for evaluating RBAC performance

VI. DISCUSSION

A. Summary of Key Findings

The system demonstrated reliable end-to-end automation from SCRUM transcripts to PR-validated task completion. LLM- driven extraction, task structuring, webhook-based synchronization, and RBAC dashboards operated cohesively, confirming the viability of a unified communication-to-code workflow.

B. Interpretation of Results

Results indicate that LLMs effectively interpret meeting intent and responsibilities, while PR-linked validation removes dependency on manual updates. Together, these mechanisms reduce coordination overhead and enhance accountability, particularly in fast-moving Agile environments.

C. Strengths of the System

- Integrated automation covering transcription, task assignment, and PR validation
- Strong semantic understanding through LLMs
- Objective, code-backed task verification
- Role-adaptive dashboards improving clarity and minimizing overload

D. Comparison with Related Work

Prior work treats meeting summarization, PR analytics, or RBAC independently. The proposed system differs by consolidating these capabilities into one synchronized pipeline, though it currently lacks the large-scale empirical testing commonly seen in established LLM or PR-analysis research.

VII. POTENTIAL ENHANCEMENT AND FUTURE WORK

A. Predictive Analytics Integration

Future iterations may incorporate predictive analytics to estimate task delays, identify coordination bottlenecks, and forecast resource needs. This would shift the system from reactive updates to proactive decision support.

B. Enhanced PR Quality Evaluation

Extending PR monitoring to include automated linting signals, test-coverage indicators, and code-quality metrics would strengthen validation accuracy and reduce merge risks.

C. Intelligent Workflow Automation

Automated reminders, follow-ups, and escalation rules can further minimize manual coordination and standardize task progression across teams.

D. Scalability and Deployment Improvements

Containerized deployment, optimized indexing, and modular microservices would improve performance for larger teams, higher PR traffic, and multi-project environments.

VIII. CONCLUSION

The proposed AI-driven workspace system streamlines SCRUM workflows by automating meeting transcript processing, task extraction, PR-based validation, and role-adaptive visibility. By integrating LLM-driven semantic parsing with GitHub event processing and RBAC dashboards, the system reduces manual updates and strengthens traceability across planning and execution. Experimental analyses show reliable task extraction, consistent PR–task synchronization, and effective access control, indicating that AI-enabled automation can enhance coordination within development teams.

REFERENCES

[1] X. Zhang, et al., "MeetingBank: A Benchmark Dataset for Multi-Topic Meeting Summarization," in Proc. ACL, 2023.
[2] M. Zhong, et al., "QMSum: A Query-Based Multi-Domain Meeting Summarization Dataset," in Proc. NAACL, 2021.
[3] S. Feng, et al., "Large Language Models for Task-Oriented Dialogue and Action Extraction," arXiv:2306.08585, 2023.
[4] P. Liu, et al., "Extracting Action Items from Noisy Transcripts Using LLM-Based Semantic Parsing," in Proc. EMNLP, 2022.
[5] J. B. Rollason, et al., "Intelligent Assistants in Agile Development: A Systematic Review," IEEE Access, 2022.
[6] R. Souza, et al., "AI-Driven Automation of SCRUM Ceremonies for Distributed Teams," Journal of Systems and Software, 2024.
[7] M. Al-Amin, et al., "End-to-End Workflow Optimization for Software Teams Using AI," Software Quality Journal, Springer, 2023.
[8] M. Wessel, et al., "The Impact of GitHub Actions on the Pull Request Process," in Proc. ICSE, 2023.
[9] R. Kikas, et al., "Pull Request Analytics for Continuous Integration," in Proc. MSR, 2021.
[10] A. Ebert, et al., "Understanding Developer Activity Through GitHub Event Streams," IEEE Transactions on Soft- ware Engineering, 2020.
[11] R. Sandhu, et al., "Role-Based Access Control Models," IEEE Computer, 2020.
[12] A. Das, et al., "Context-Aware RBAC Systems for Secure Multi-User Environments," Computers & Security, 2023.